Capítulo 12

Algoritmo *RSA*

Nota: Los puntos marcados con ^{MD} se verán en la asignatura Matemética Discreta. Los marcados con ^{MC} en Matemáticas para la Computación. Los marcados con ambos se verán especialmente en la primera asignatura que aparezca, y se recordarán brevemente en la otra.

12.1 Introducción

El método de encriptado de datos conocido como algoritmo RSA, por los nombres de sus inventores (Rivest, Shamir y Adleman) es uno de los más usados hoy día para la transmisión segura de datos a través de canales inseguros. Este documento es una introducción a las bases matemáticas de dicho algoritmo de encriptado escrita para llegar desde unos conocimientos mínimos (el concepto de anillo de los enteros y la existencia y unicidad de la descomposición en factores primos de un entero) hasta la comprensión del algoritmo en sí, tratando de no omitir casi ninguna demostración (aunque algunas se realizarán en la clase, o se dejan como ejercicio, y una de ellas excede el alcance del curso). Está casi íntegramente basado en los libros del prof. Manuel Lucena (Univ. de Jaén) y de los profesores J.M. Basart, J. Rifá y M. Villanueva (Universitat Autònoma de Barcelona). Véase la bibliografía al final.

12.2 Anillo de los enteros^{MD}

Un concepto del que se parte es la existencia de un conjunto llamado de los números enteros, en el que está definida una relación de orden total (ser mayor que) y unas operaciones aritméticas (suma y producto, con las definiciones usuales) que le confieren estructura de anillo, es decir:

$\mathsf{Z} = \{\ldots, -4, -3, -2, -1, 0, 1, 2, 3, \ldots\}$

- La operación $+$ en Z es conmutativa, asociativa, tiene elemento neutro (el 0) y **todo** elemento tiene su simétrico, llamado **opuesto**. Por tanto, $(\mathsf{Z}, +)$ es grupo conmutativo.

- La operación $\cdot$ en Z es conmutativa, asociativa y tiene elemento neutro (el 1). Sin embargo, no todo elemento tiene simétrico, que aquí se llamaría **inverso** (de hecho, solamente el 1 y el -1 lo tienen, y son ellos mismos).

- La operación $\cdot$ es distributiva respecto a $+$.

Por tanto, $(\mathsf{Z}, +, \cdot)$ es un **anillo conmutativo con elemento unidad**, llamado el **anillo de los enteros**.

12.3 Conceptos básicos de aritmética en Z

12.3.1 División entera^{MD}

Dados $a, b \in \mathsf{Z}$, diremos que la división de a entre b tiene cociente q y resto r si es cierto que $a = bq + r$, siendo q un número entero sin restricciones, y r un número entero comprendido entre 0 y $b - 1$.

Ejemplos:
137 dividido entre 21 da como cociente 6 y como resto 11, dado que se puede escribir 137 como $137 = 21 \cdot 6 + 11$.
-137 dividido entre 21 da como cociente -7 y como resto 10, dado que se puede escribir -137 como $-137 = 21 \cdot (-7) + 10$. Nótese que escribir -137 como $-137 = 21 \cdot (-6) - 11$ no cumple los requisitos de la definición de división entera, dado que el supesto resto, -11, no está entre 0 y 21.

12.3.2 Múltiplos y divisores^{MD}

Dados dos números enteros, $a, b \in \mathsf{Z}$, con $a \leq b$, se dice que a es múltiplo de b, o equivalentemente, que b es divisor de a, si existe algún entero $q \in \mathsf{Z}$ tal que $a = q \cdot b$, o lo que es lo mismo, si al realizar la división entera de a entre b, según se ha definido en el punto anterior, el cociente es q y el resto es 0. Es obvio que 1 es divisor de cualquier número entero, dado que $\forall a \in \mathsf{Z}, a = a \cdot 1$ con lo que el cociente es el propio a y el resto 0.

12.4 Definiciones básicas

12.4.1 Máximo común divisor^{MD}

Dados n números enteros, $\{a_1, \ldots, a_n\} \in \mathsf{Z}$ se llama **máximo común divisor**, abreviado m.c.d., al mayor número entero positivo m que es divisor de todos ellos, y se escribe $m = mcd(a_1, \ldots, a_n)$. Como 1 es divisor de cualquier número, en ausencia de otro divisor común mayor, 1 sería el m.c.d. de cualquier conjunto de enteros.

12.4.2 Mínimo común múltiplo^{MD}

Dados n números enteros, $\{a_1, \ldots, a_n\} \in \mathsf{Z}$ se llama **mínimo común múltiplo**, abreviado m.c.m., al menor número entero positivo m que es múltiplo de todos ellos, y se escribe $m = mcm(a_1, \ldots, a_n)$. Como el producto de un a_i cualquiera por cualquier entero (p. ej., por el producto de todos los demás a_j) es múltiplo de a_i, entonces, en ausencia de otro múltiplo menor, al producto de todos los a_i sería el m.c.m. de cualquier conjunto de enteros.

12.4.3 Números primos entre sí^{MD,MC}

Dados dos números enteros a y b, se dice que son primos entre sí si no tienen ningún divisor común, excepto el 1, es decir, si no existe ninguna terna de enteros p, q, r con $p \neq 1$ tales que $a = p \cdot q$ y $b = p \cdot r$.

12.4.4 Números primos^{MD}

Dado un número entero a, se dice que es primo si sus únicos divisores son él mismo y la unidad. De las definiciones anteriores se deduce que no necesariamente un número primo es primo entre sí con cualquier otro número, pero sí lo es con cualquier otro número primo.

12.5 Descomposición en factores primos^{MD}

Dado un número entero a, siempre se puede escribir de modo único como

$$a = p_1^{k_1} \cdot p_2^{k_2} \cdot \ldots \cdot p_n^{k_n} = \prod_{i=1}^{n} p_i^{k_i}$$

siendo todos los p_i números primos estrictamente menores que a y k_i exponentes naturales. A cada uno de los números primos p_i se le llama factor primo de a.

12.5.1 Cálculo del mcd y del mcm por descomposición^{MD}

Dados dos enteros, a y b:
- Su mcd se puede calcular multiplicando todos los factores primos que aparecen en ambas descomposiciones (los factores primos comunes), elevado cada uno de ellos al menor de los dos exponentes con que aparece.
- Su mcm se puede calcular multiplicando todos los factores primos que aparecen en cualquiera de las descomposiciones (los factores primos comunes, y los no comunes), elevado cada uno de ellos al mayor de los dos exponentes con que aparece. Los factores que sean comunes se multiplicarán una sola vez.

Ejemplos:

Hallar mcd y mcm de 643 y 412. Descomponiendo,
$643 = 643 \cdot 1$ (este número es primo)
$412 = 103 \cdot 2^2 \cdot 1$ Luego $mcd(643, 412) = 1$ y $mcm(643, 412) = 643 \cdot 103 \cdot 2^2 \cdot 1 = 264916$

Hallar mcd y mcm de 22253 y 4675.
$22253 = 17^2 \cdot 11 \cdot 7 \cdot 1$
$4675 = 17 \cdot 11 \cdot 5^2 \cdot 1$
Luego $mcd(22253, 4675) = 17 \cdot 11 \cdot 1 = 187$ y $mcm(22253, 4675) = 17^2 \cdot 11 \cdot 7 \cdot 5^2 \cdot 1 = 556325$

En una de las demostraciones siguientes (concretamente, el cálculo de la función multiplicativa de Euler) necesitaremos la siguiente proposición, que pasamos a enunciar:

Prop.:^{MC} Si $mcd(p,q) = 1$, entonces $mcm(p,q) = pq$
Dem.: Por ser el mcd 1, el 1 es el único factor que aparece en ambas descomposiciones (el único factor común). Esto significa que todos los demás son no comunes, por lo que todos ellos deberán multiplicarse para calcular el mcm. Además, deberán hacerlo con el exponente con el que aparecen, que es el mayor, dado que es el único. Por tanto, las dos descomposiciones aparecen íntegras y multiplicadas en el cálculo del mcm, por tanto el mcm es la multiplicación de ambos números.

12.5.2 Algoritmo de Euclides^{MC}

Este es un método para calcular el máximo común divisor de dos números cualesquiera, sin necesidad de descomponerlos en factores primos. Esto es muy importante desde el punto de vista de la computación, porque descomponer un número en factores primos es (que se sepa) un algoritmo de coste exponencial en el tamaño del número, o sea, el tiempo que cuesta ejecutarlo crece exponencialmente con dicho tamaño. Sin embargo, el algoritmo de Euclides cuesta de ejecutarse un tiempo lineal con el tamaño. El algoritmo usa la siguiente proposición:
Prop.: Dados dos enteros positivos a y b, si se divide el mayor entre el menor, lo cual da un cociente q y un resto r, es decir, $a = bq + r$, se cumple que $mcd(a,b) = mcd(b,r)$.
Dem.:
Llamemos $d = mcd(a,b)$ y $d' = mcd(b,r)$. Entonces d es por definición divisor de b; por otra parte, $r = a - bq$, y como d divide a a y también a bq (porque divide a b) entonces d es también divisor de r.
Por otra parte, por definición de d' sabemos que d' es divisor de b y de r. Tenemos pues dos divisores de b y de r, siendo d' por definición el mayor de todos. Así pues, deberá ser $d \leq d'$.
Por otra parte, d' es por definición divisor de b, y como $a = bq + r$ y d' es divisor de b y de r, lo es de a. Tenemos pues que d' es divisor de b y de a, y también que d es divisor de b y de a, y es el máximo de todos ellos. Por tanto, deberá ser $d' \leq d$. El único modo de que simultáneamente $d \leq d'$ y $d' \leq d$ es que $d' = d$, como queríamos probar.

El algoritmo de Euclides se basa en la anterior proposición: dados a y b, se calcula la división entera entre ellos, obteniendo un cociente q y un resto r. Entonces se calcula de nuevo la división entera entre b y r, obteniendo un nuevo cociente y un nuevo resto. El proceso se itera hasta que el resto sea 0, en cuyo caso el último divisor b es el máximo común divisor. Es fácil ver que la sucesión de los restos es estrictamente decreciente, y como se trata de una sucesión de números positivos, llegará a valer 0, con lo que el algoritmo efectivamente acaba.

Ejemplo: sean $a = 643$ y $b = 412$. Entonces, escribiendo en cada línea $a = q \cdot b + r$,
L1) $643 = 1 \cdot 412 + 231$ (dividiendo 643 entre 412)
L2) $412 = 1 \cdot 231 + 181$ (dividiendo 412 entre 231)
L3) $231 = 1 \cdot 181 + 50$ (etc..)
L4) $181 = 3 \cdot 50 + 31$
L5) $50 = 1 \cdot 31 + 19$
L6) $31 = 1 \cdot 19 + 12$
L7) $19 = 1 \cdot 12 + 7$
L8) $12 = 1 \cdot 7 + 5$

L9) $7 = 1 \cdot 5 + 2$
L10) $5 = 2 \cdot 2 + 1$
L11) $2 = 2 \cdot 1 + 0$

Por tanto, el mcd de 643 y 412 es el último divisor b, o sea, 1. En este caso particular hemos encontrado que 643 y 412 son primos entre sí. En general, obsérverse que en cada paso el algoritmo sustituye cada dividendo por el último resto obtenido, y cada divisor por el penúltimo resto, calculando entonces de nuevo la división para generar los actuales cociente y resto. Esto no vale para los dos primeros pasos, en los que aún no se tienen dos restos anteriores. Si llamamos `D,d,c` y `r` a arrays conteniendo los dividendos, divisores, cocientes y restos de cada paso respectivamente, representamos con `/` la división entera y con `mod` el resto de dicha división, el algoritmo queda:

Inicializaciones:
$D(1) \longleftarrow a$
$d(1) \longleftarrow b$
$c(1) \longleftarrow D(1)/d(1)$
$r(1) \longleftarrow D(1)$ `mod` $d(1)$
$D(2) \longleftarrow d(1)$
$d(2) \longleftarrow r(1)$
$c(2) \longleftarrow D(2)/d(2)$
$r(2) \longleftarrow D(2)$ `mod` $d(2)$
(sigue...)

(...continúa)
$i \longleftarrow 3$
`Mientras` $r(i) \neq 0$ `hacer`
  $D(i) \longleftarrow r(i-2)$
  $d(i) \longleftarrow r(i-1)$
  $c(i) \longleftarrow D(i)/d(i)$
  $r(i) \longleftarrow D(i)$ `mod` $d(i)$
  $i \longleftarrow i+1$
`finmientras`

Nótese que esta es una versión no óptima del algoritmo, especialmente en lo que respecta al espacio usado (no es realmente necesario usar arrays), pero se da por ser la más clara.

12.5.3 Algoritmo extendido de Euclides^{MC}

Una variación interesante del algoritmo de Euclides que será útil posteriormente es el algoritmo extendido. Es posible probar que el máximo común divisor de dos enteros $d = mcd(a, b)$ (y en general cualquier otro divisor) se puede escribir siempre como combinación lineal de ambos, o sea, existen dos enteros λ y μ tales que $d = \lambda \cdot a + \mu \cdot b$. A esta igualdad se la llama "Identidad de Bezout". Para encontrar dichos enteros basta recorrer los pasos del algoritmo de Euclides visto antes, al revés, realizando en cada paso sólo las sustituciones hacia atrás que sean pertinentes, y exactamente ellas. En el paso i-ésimo se debe sustituir en el despeje del resto correspondiente a dicho paso; debe hacerse en uno de sus dos sumandos cada vez, y usando para ello la igualdad del paso $i-2$, excepto en el primero, en que se usa la igualdad del paso inmediatamente anterior. En el ejemplo visto antes, comenzamos en la línea que contiene el mcd (línea 10) y lo despejamos (en este caso vale 1):

$1 = 5 - 2 \cdot 2$

Ahora, sustituímos en el segundo sumando usando la línea 9
$1 = 5 - 2 \cdot (7 - 1 \cdot 5) = 5 - 2 \cdot 7 + 2 \cdot 5 = 3 \cdot 5 - 2 \cdot 7$

Sustituímos en el primer sumando, usando la línea 8
$1 = 3 \cdot (12 - 1 \cdot 7) - 2 \cdot 7 = 3 \cdot 12 - 3 \cdot 7 - 2 \cdot 7 = 3 \cdot 12 - 5 \cdot 7$

Sustituimos en el segundo sumando usando la linea 7

$1 = 3 \cdot 12 - 5 \cdot (19 - 1 \cdot 12) = 3 \cdot 12 - 5 \cdot 19 + 5 \cdot 12 = 8 \cdot 12 - 5 \cdot 19$

Sustituimos en el primer sumando usando la linea 6
$1 = 8 \cdot (31 - 1 \cdot 19) - 5 \cdot 19 = 8 \cdot 31 - 8 \cdot 19 - 5 \cdot 19 = 8 \cdot 31 - 13 \cdot 19$

Sustituimos en el segundo sumando usando la linea 5
$1 = 8 \cdot 31 - 13 \cdot (50 - 1 \cdot 31) = 8 \cdot 31 - 13 \cdot 50 + 13 \cdot 31 = 21 \cdot 31 - 13 \cdot 50$

Sustituimos en el primer sumando usando la linea 4
$1 = 21 \cdot (181 - 3 \cdot 50) - 13 \cdot 50 = 21 \cdot 181 - 63 \cdot 50 - 13 \cdot 50 = 21 \cdot 181 - 76 \cdot 50$

Sustituimos en el segundo sumando usando la linea 3
$1 = 21 \cdot 181 - 76 \cdot (231 - 1 \cdot 181) = 21 \cdot 181 - 76 \cdot 231 + 76 \cdot 181 = 97 \cdot 181 - 76 \cdot 231$

Sustituimos en el primer sumando usando la linea 2
$1 = 97 \cdot (412 - 1 \cdot 231) - 76 \cdot 231 = 97 \cdot 412 - 97 \cdot 231 - 76 \cdot 231 = 97 \cdot 412 - 173 \cdot 231$

Sustituimos en el segundo sumando usando la linea 1
$1 = 97 \cdot 412 - 173 \cdot (643 - 412) = 97 \cdot 412 - 173 \cdot 643 + 173 \cdot 412 = 270 \cdot 412 - 173 \cdot 643$

de donde hemos hallado que $1 = \lambda \cdot 412 + \mu \cdot 643$ con $\lambda = 270$ y $\mu = -173$.

Como ejercicio, posiblemente para realizar en prácticas, escríbase la versión general del algoritmo extendido de Euclides usando la misma notación propuesta en el algoritmo simple. Como veremos después, el algoritmo extendido es muy interesante porque nos permitirá calcular inversas módulo n, una operación necesaria para el cálculo de claves en el RSA.

12.6 Aritmética modular

12.6.1 Relación de congruencia^{MD,MC}

Def.: Dado un entero positivo, n, se define la **relación de congruencia módulo n** entre dos enteros, a y b, y se escribe

$$(a)_n \equiv (b)_n$$

(que se lee: 'a es congruente con b en módulo n') si a y b dan el mismo resto al dividirlos entre n

Ej.: en módulo 5, los números -10,0,15,875,...son todos congruentes (cualquiera lo es con cualquiera de los otros), dado que al dividirlos por 5 todos dan como resto 0. Del mismo modo, -13,2,17,182 son igualmente congruentes, puesto que al dividir entre 5 todos dan como resto 2 (obsérvese el caso de $-13 = (-3) \cdot 5 + 2$).

Nótese que para generar todos los números que son congruentes con uno cualquiera, a, en módulo n no tenemos más que sumar (restar) n cada vez, es decir, serán $a + n, a + 2n, a + 3n, \ldots, a - n, a - 2n, a - 3n, \ldots$. De este modo, la diferencia entre cualesquiera de ellos, $a + pn$ y $a + qn$ será $(p - q)n$, o sea, múltiplo de n. Esto nos permite la siguiente definición alternativa de congruencia:

Def.: Se dice que $(a)_n \equiv (b)_n$ si existe un entero k tal que $(a - b) = k \cdot n$

La prueba de la equivalencia de ambas definiciones es trivial, y se deja como ejercicio.

12.6.2 Propiedades de la relación de congruencia^{MD}

A partir de la segunda definición de congruencia es trivial probar que esta relación tiene las propiedades reflexiva, simétrica y transitiva (se probará en clase), por lo cual es una relación de equivalencia, con lo que el conjunto de los enteros Z queda dividido en clases de equivalencia, disjuntas entre sí. Para la congruencia módulo n habrá exactamente n de estas clases (los enteros que al dividirlos entre n dan como resto 0, los que dan como resto 1,...,los que dan $n-1$) y las denotaremos por $[0]_n \dots [n-1]_n$. De este modo, cada clase de equivalencia tendrá un representante canónico, que tomaremos como el menor entero positivo que pertenezca a ella, pero teniendo claro que cualquier otro miembro de la clase la representa igualmente; por ello, para todo el desarrollo posterior, será equivalente escribir, p. ej., $(3)_5$ que $(8)_5$, dado que ambos pertenecen a la clase $[3]_5$.

Def.: Definimos el conjunto de los enteros módulo n, denotado Z_n, como el conjunto finito (de exactamente n elementos) formado por las n clases de equivalencia $[0]_n, \dots, [n-1]_n$ en que resulta dividido el conjunto de los enteros Z por la relación de congruencia.

Definamos ahora en Z_n las operaciones suma y producto:

$$[a]_n + [b]_n = (a' + b')_n \text{ y}$$
$$[a]_n \cdot [b]_n = (a' \cdot b')_n$$

siendo a' cualquier elemento de la clase $[a]_n$ y b' cualquier elemento de la clase $[b]_n$.

Ejemplos:
$[2]_4 + [3]_4 = (5)_4 = [1]_4$
$[5]_7 \cdot [4]_7 = (20)_7 = [6]_7$

Estas operaciones tienen las siguiente propiedades (se probará en clase):

- La suma es asociativa, conmutativa, tiene elemento neutro (el $[0]_n$) y **todo** elemento tiene su simétrico, llamado **opuesto**. (el opuesto de la clase $[a]_n$ es la clase $[n-a]_n$, dado que su suma sería $(n)_n = [0]_n$, o sea, sumados dan el elemento neutro. Por ello, $(\mathsf{Z}_n, +)$ es grupo conmutativo.
- El producto es asociativo, conmutativo y tiene elemento neutro (el $[1]_n$)
- El producto es distributivo respecto a la suma.

Por todo ello, podemos afirmar que las operaciones suma y producto entre clases recién definidas dotan a Z_n de estructura de **anillo conmutativo con elemento unidad**. Hasta aquí, la única diferencia con Z es que Z_n es un conjunto finito.

12.6.3 Inversas en Z_n^{MD,MC}

Respecto al elemento simétrico del producto, que llamaremos **inverso**, nótese que **no** todos los elementos de Z_n distintos del elemento $[0]_n$ lo tienen (es decir, no todos son invertibles), pero sí algunos. Como ejemplos:

Ej. 1: En Z_8, $[3]_8$ tiene inverso, que es él mismo, $[3]_8$, dado que $[3]_8 \cdot [3]_8 = (9)_8 = [1]_8$, o sea, su producto da el elemento neutro. Pero sin embargo, $[4]_8$ no lo tiene, puesto que no existe ningún elemento de Z_n tal que multiplicándolo por $[4]_8$ nos de $[1]_8$.
Ej. 2: En Z_7 todo elemento distinto del $[0]_n$ tiene inverso. (Ejercicio: ver cuáles son los inversos de cada uno de los siete elementos).

Nótese que en Z_n, si $[a]_n$ es el inverso de $[b]_n$, entonces $[b]_n$ lo es de $[a]_n$; y véase igualmente que una diferencia sustancial entre Z_n y Z es que en Z los únicos elementos invertibles eran 1 y -1.

Def.: Definimos el conjunto Z_n^* como el subconjunto de Z_n formado por aquellos elementos que son invertibles.

Estos elementos vienen caracterizados por la siguiente proposición:
Prop.: Un elemento $[a]_n$ de Z_n es invertible si, y solo si, a y n son primos relativos, es decir, si $mcd(a, n) = 1$.
Dem.:
($\Rightarrow$): Si a es invertible, $\exists\, [b]_n \,/\, [a]_n \cdot [b]_n = [1]_n$, o sea, $(ab)_n = (1)_n$ de donde $ab = kn + 1$. Imaginemos ahora que 1 no fuera el mcd de a y n, sino que estos tuvieran otro, digamos $d > 1$. Entonces, $a = pd$ y $n = qd$ con p, q enteros. Así pues, sería $pdb = kqd + 1$, de donde $d(pb - kq) = 1$. Ahora bien, si el producto (usual, no en módulo) de dos enteros es 1, o bien ambos son 1, contra hipótesis de que d era mayor que 1, o bien ambos son -1, también contra la misma hipótesis. Hemos llegado pues a contradicción, de donde el mcd de a y n debe ser 1.
($\Leftarrow$): Según vimos al explicar el algoritmo extendido de Euclides, $\forall c_1, c_2 \in \mathsf{Z}/mcd(c_1, c_2) = 1, \exists \lambda, \mu \in \mathsf{Z}/\lambda c_1 + \mu c_2 = 1$. Esto en nuestro caso significa que podremos escribir $1 = \lambda a + \mu n$, o sea, $\lambda a = -\mu n + 1$, lo que significa que el producto de λ y a es un múltiplo de n, más 1, o sea que pertenece a la clase $[1]_n$. Por definición de inverso, λ es pues inverso de a. Si λ fuese negativo, nótese que podríamos sumarle n las veces que hiciera falta hasta alcanzar un número positivo que seguiria cumpliendo la última igualdad y sería el representante canónico de la clase inversa de aquella a la que a pertenece. Análogamente, si λ fuera mayor que n podríamos restarle n las veces necesarias hasta que estuviese comprendido entre 0 y n.
Corolario: Si n fuese primo, todos los enteros entre 1 y n son primos relativos con él (o sea, $\forall p < n, mcd(p, n) = 1$). Esto siginifica que todos los elementos de Z_n serían invertibles (o sea, que $\mathsf{Z}_n^* = \mathsf{Z}_n$), y por tanto, por tener cada elemento su inverso, que $(\mathsf{Z}_n, +, .)$ sería un cuerpo. Dichos cuerpos son un caso particular de las estructuras algebráicas conocidas como Campos de Galois.

Para alguna de las demostraciones posteriores será necesario simplificar congruencias. Para poder hacerlo probaremos la siguiente propiedad:
Prop.:^{MC} (propiedad de simplificación de congruencias) Si a y b son congruentes módulo n, o sea $(a)_n = (b)_n$, entonces para cualquier entero k se cumple que a y b multiplicados por k serán congruentes en módulo kn, o sea, $(ka)_{kn} = (kb)_{kn}$.
Dem.:
Usando la definición alternativa de congruencia que dimos antes, $(a)_n = (b)_n$ significa que existe un entero p tal que $(a - b) = pn$. Multiplicando por k ambas partes de esta igualdad,

$k(a-b) = kpn$, o sea, $(ka - kb) = p(kn)$, de modo que existe un entero p tal que la diferencia de los números ka y kb es dicho entero por kn. De nuevo por la definición, esto significa que $(ka)_{kn} = (kb)_{kn}$. La demostración es invertible.

12.7 Orden y sus propiedades

12.7.1 Orden de un elemento de Z_n ^{MD,MC}

Otro de los conceptos que necesitaremos después es el de **orden** de un elemento del anillo Z_n. La pregunta que nos hacemos es cuáles son la potencias sucesivas de algún elemento a, y en concreto, si alguna de esas potencias sucesivas nos va a dar el elemento neutro, $[1]_n$. Con este objeto se define:

Def.: Definimos el **orden** e de un elemento a de Z_n como el mínimo número natural e (o sea, no nulo) tal que $a^e = (1)_n$, si dicho número existe (es posible que para algunos elementos a de Z_n ningún natural cumpla esta igualdad).

Ej. 1: En Z_{11} el orden de 2 es 10, puesto que $2^1 = 2 = (2)_{11}$, $2^2 = 4 = (4)_{11}$, $2^3 = 8 = (8)_{11}$, $2^4 = 16 = (5)_{11}$, $2^5 = 32 = (10)_{11}$, $2^6 = 64 = (9)_{11}$, $2^7 = 128 = (7)_{11}$, $2^8 = 256 = (3)_{11}$, $2^9 = 512 = (6)_{11}$, $2^{10} = 1024 = (1)_{11}$.

Ej. 2: En Z_4, el número 2 no es invertible, ya que $2^1 = 2 = (2)_4$, $2^2 = 4 = (0)_4$, $2^3 = 8 = (0)_4$, y así sucesivamente, no dando ninguna de estas potencias un múltiplo de 4, más 1 (o sea, $(1)_n$).

El problema ahora es pués cómo saber qué elementos de Z_n tienen orden y cuáles no. Esto nos lo da la siguiente caracterización:

Prop.: En Z_n, un elemento a tiene orden si, y sólo si, es invertible.
Dem.:
($\Rightarrow$) Si a tiene orden significa que existe un natural e tal que $a^e = (1)_n$; pero a^e puede escribirse como $a \cdot a^{e-1}$, con lo que $a \cdot a^{e-1} = (1)_n$. Esto significa precisamente que a^{e-1} es la inversa de a (su producto da el elemento neutro), luego a es invertible.

($\Leftarrow$) Podemos ir calculando las potencias sucesivas de a en módulo n, las cuales son un conjunto de infinitos números. Pero como todos ellos pertenecen a Z_n, que es finito, debe haber necesariamente números repetidos, es decir, dos potencias, la r-ésima y la s-ésima con (digamos) $s > r$ serán tales que $(a^s)_n = (a^r)_n$. Ahora bien, a es invertible; multiplicando ambas partes de esta igualdad por el inverso de a nos quedaría $\left(a^{s-1}\right)_n = \left(a^{r-1}\right)_n$, y si seguimos haciendo esto $s - r$ veces, terminaremos obteniendo que $(a^{s-r})_n = (1)_n$. Luego existe un número $e = s - r$ tal que a elevado a él nos da el elemento neutro. Si este número es el menor de los que cumplen dicha propiedad, él sería el orden. En otro caso, habría otro número más pequeño que lo sería, pero de cualquier modo tal número existe.

12.7.2 Función multiplicativa de Euler^{MD,MC}

El siguiente concepto importante en nuestro desarrollo es la definición y propiedades de la llamada **función multiplicativa de Euler**. Informalmente, si tenemos un entero p, éste será divisible por algunos, aunque no todos, los enteros anteriores a él (en el caso extremo de que fuese primo, por ninguno). O sea, algunos de estos enteros serán primos relativos con p y otros no. La función de Euler nos dirá, para cada p, cuántos lo son.

Def.: Se define la función multiplicativa de Euler, $\phi(r)$, como el número de números enteros mayores o iguales que 1 y menores que r que son primos relativos con él, es decir:
$\phi(r) = \#\{s \ / \quad s >= 1 \text{ y } s < r \text{ y } mcd(s,r) = 1\}$
donde # denota el cardinal del conjunto.

Veamos ahora cómo, dado un número r, podemos calcular el valor de $\phi(r)$.

1. Si r es primo, $\phi(r) = r - 1$
2. Si r es el producto de dos números primos entre sí, o sea, $r = pq$ con $mcd(p,q) = 1$, entonces $\phi(r) = \phi(p)\phi(q)$
3. Si $r = p^k$ con p primo, entonces $\phi(r) = p^k - p^{k-1} = p^k(p-1)$

Dem.:^{MD} La demostración de estas tres reglas para el cálculo de $\phi(r)$ involucra lemas y teoremas anteriores, cuya demostración excede el alcance de este curso. Los alumnos interesados pueden consultarla en el primero de los libros citados en la bibliografía.

12.7.3 Orden de un elemento en Z_n^* y su relación con $\phi(n)^{MC}$

Estableceremos ahora una proposición y un teorema que nos permitirán conocer la relación que existe entre el órden de un elemento de Z_n y el valor de la función $\phi(n)$.

Prop.: Sea a un elemento de Z_n^*, es decir, un elemento de Z_n invertible. Ya fue probado que a tiene orden, es decir, existe un entero positivo e tal que $(a^e)_n = 1$. Pues bien: dicho número e es divisor de $\phi(n)$, es decir, existe un entero positivo k tal que $\phi(n) = ke$.
Dem.:
Construyamos las sucesivas potencias de a en modulo n, $\left(a^0\right)_n, \left(a^1\right)_n, \ldots, \left(a^{e-1}\right)_n, (a^e)_n = (1)_n$. No habrá entre los e primeros elementos de esta serie ninguno repetido, puesto que de ser así estaríamos en un ciclo que no pasaría nunca por el 1, y por tanto a no tendría orden, como estamos suponiendo.

Nótese que el conjunto así generado tiene (exceptuando al último elemento, que vuelve a ser 1) exactamente e elementos; ahora bien, este conjunto puede ser todo Z_n^*, y por tanto el cardinal de Z_n^*, que es $\phi(n)$, sería el de este conjunto, o sea, e y la proposición se cumple con $\phi(n) = 1 \cdot e$.

Pero también puede ser que queden elementos es Z_n^* que no sean potencias de a. Sea b uno de estos elementos, y construyamos los productos $\left(b \cdot a^0\right)_n, \left(b \cdot a^1\right)_n, \ldots, \left(b \cdot a^{e-1}\right)_n$. Dichos números (y hay e de ellos) son todos diferentes, por el mismo argumento que antes (no puede

haber ciclos, puesto que b es invertible); además, son diferentes de cualquiera de los a^j generados antes, ya que si alguno fuera igual, $\left(b \cdot a^i\right)_n = \left(a^j\right)_n$, multiplicando esta igualdad por el inverso de a^i, que es a^{e-i}, se tendría que $(b)_n = \left(a^{j+e-i}\right)_n$, o sea, b sería potencia de a, contra hipótesis. Entonces, el cardinal del conjunto $\left\{\left(a^0\right)_n, \left(a^1\right)_n, \ldots, \left(a^{e-1}\right)_n, \left(b \cdot a^0\right)_n, \left(b \cdot a^1\right)_n, \ldots, \left(b \cdot a^{e-1}\right)_n\right\}$ es $2e$.

Nuevamente, puede que este conjunto sea todo Z_n^*, con lo cual $\phi(n) = 2e$, o que no. Si no, se escoge un nuevo elemento c de Z_n^* distinto de a y de b, y se repite el proceso, lo que añadirá e elementos más. Como Z_n^* es finito, al cabo de un número finito de pasos, digamos λ, todo Z_n^* habrá sido generado, y se tendrá que $\phi(n) = \lambda e$, siendo por tanto e un divisor exacto de $\phi(n)$, como queríamos probar.

Teorema de Euler: Si a es un elemento de Z_n^*, entonces $\left(a^{\phi(n)}\right)_n = (1)_n$.
Dem.:
En efecto, por la proposición anterior, $\phi(n) = \lambda e$, siendo λ un entero, y e el orden de a. Entonces,
$\left(a^{\phi(n)}\right)_n = \left(a^{\lambda e}\right)_n = \left((a^e)^\lambda\right)_n = \left(1^\lambda\right)_n = (1)_n$
donde se ha usado que, por ser e el orden, a^e es (en módulo n) igual a 1.

12.8 El criptosistema RSA^{MC}

12.8.1 Proposición previa^{MC}

Queda sólo una proposición que es necesario enunciar y probar para conocer toda la base teórica que la implementación más sencilla del RSA necesita. Dicha proposición es la siguiente:

Prop.: Sea n un número natural que es producto de dos números primos, p y q. Sea ahora t un número natural perteneciente a la clase $[1]_{\phi(n)}$, o sea, $(t)_{\phi(n)} = (1)_{\phi(n)}$. Entonces, para cualquier número x, $\left(x^t\right)_n = (x)_n$, o sea, en módulo n, x^t y x pertenecen a la misma clase.
Dem.:
En primer lugar escribamos la descomposición de x en factores primos, que será de la forma $x = p_1^{\alpha_1} \cdot \ldots \cdot p_r^{\alpha_r} \cdot p^\beta \cdot q^\gamma$. Es posible que los factores p y q no aparezcan en la descomposición de x, con lo que β, γ o ambos podrían ser 0, pero eso no altera el resto de la demostración.

Por otra parte, asumimos que $(t)_{\phi(n)} = (1)_{\phi(n)}$, por lo que t se podrá escribir como $t = \lambda\phi(n) + 1$ siendo λ un entero. Entonces, $\left(x^t\right)_n$ se puede escribir como:

$$\left(x^t\right)_n = \left(x^{1+\lambda\phi(n)}\right)_n = (x)_n \cdot \left(\left(x^{\phi(n)}\right)^\lambda\right)_n \quad (*)$$

Ahora bien,
$\left(x^{\phi(n)}\right)_n = \left(p_1^{\alpha_1\phi(n)} \cdot \ldots \cdot p_r^{\alpha_r\phi(n)} \cdot p^{\beta\phi(n)} \cdot q^{\gamma\phi(n)}\right)_n$

No obstante, por ser factores distintos de la descomposición en factores primos de un mismo número x, cualquier factor o producto de ellos es primo relativo con cualquier otro, en particular, $mcd(p_i, pq) = mcd(p_i, n) = 1$. lo que quiere decir que p_i es invertible en módulo

$n = pq$, y por tanto cumple la condición del Teoreme de Euler. Aplicando dicho teorema, $\left(p_i^{\phi(n)}\right)_n = (1)_n$ para cualquier i entre 1 y r.
Así pues, la igualdad anterior queda

$\left(x^{\phi(n)}\right)_n = \left((p^{\phi(n)})^\beta \cdot (q^{\phi(n)})^\gamma\right)_n$

Sustituyendo en la igualdad marcada con (*) obtenemos:

$\left(x^t\right)_n = \left(p_1^{\alpha_1} \cdot \ldots \cdot p_r^{\alpha_r} \cdot p^\beta \cdot q^\gamma \cdot \left((p^{\phi(n)})^\beta \cdot (q^{\phi(n)})^\gamma\right)^\lambda\right)_n = \left(p_1^{\alpha_1} \cdot \ldots \cdot p_r^{\alpha_r} \cdot (p^{\lambda\phi(n)+1})^\beta \cdot (q^{\lambda\phi(n)+1})^\gamma\right)_n =$
$\left(p_1^{\alpha_1} \cdot \ldots \cdot p_r^{\alpha_r} \cdot (p^t)^\beta \cdot (q^t)^\gamma\right)_n$ (**)

Ahora probaremos que $\left(p^t\right)_n = (p)_n$. Para ello, veamos que, como el $mcd(p, q) = 1$, se cumplirá,, usando el teorema de Euler visto arriba con $a = p$ y $n = q$, que

$\left(p^{\phi(q)}\right)_q = (1)_q$. Elevando ambas partes de la igualdad a $\lambda\phi(p)$ queda

$\left(p^{\lambda\phi(p)\phi(q)}\right)_q = \left(1^{\lambda\phi(p)}\right)_q = (1)_q$

Usando ahora la propiedad de simplificación de congruencias al revés (es decir, multiplicando ambos miembros y la base de congruencia por p) queda:

$\left(p \cdot p^{\lambda\phi(p)\phi(q)}\right)_{qp} = (p \cdot 1)_{qp}$, y como $pq = n$,

$\left(p^{\lambda\phi(p)\phi(q)+1}\right)_n = \left(p^t\right)_n = (p)_n$

De modo análogo, y partiendo de $\left(q^{\phi(p)}\right)_p = (1)_p$ se puede llegar a $\left(q^t\right)_n = (q)_n$. Usando estas dos expresiones, la igualdad marcada con (**) queda:

$\left(x^t\right)_n = \left(p_1^{\alpha_1} \cdot \ldots \cdot p_r^{\alpha_r} \cdot p^\beta \cdot q^\gamma\right)_n = (x)_n$, como queríamos probar.

12.8.2 El algoritmo RSA^{MC}

Veamos cómo se pueden usar los resultados obtenidos anteriormente para la construcción de un sistema que permita la comunicación segura entre un emisor, que llamaremos E, y un receptor, R. Se supone que ambos pueden ejecutar en sus respectivas máquinas las operaciones que deseen, y guardar la información que precisen, sin que ésta sea conocida por nadie. No obstante, cuando el emisor mande un mensaje encriptado al receptor no puede estar seguro de que el canal no sea espiado, y por ello su objetivo es que, aunque este mensaje encriptado sea interceptado por un espía, éste no podrá desencriptarlo, ni por tanto entenderlo.

Los pasos son:

1. En privado, el receptor R escoge dos números primos p y q muy grandes (de unas 100 cifras cada uno), y los multiplica, obteniendo $n = pq$.

2. También en privado, el receptor obtiene el valor de la función multiplicativa de Euler, $\phi(n)$, que como sabemos por el caso segundo de su procedimiento de cálculo, será en este caso igual a $\phi(n) = \phi(pq) = \phi(p)\phi(q) = (p-1)(q-1)$, dado que p y q son primos entre sí, y cada uno de ellos es primo.

3. En privado, el receptor R escoge un número e tal que $1 < e < \phi(n)$ de manera que sea primo relativo con $\phi(n)$, y le calcula su inverso módulo $\phi(n)$, que llamaremos $d = \left(e^{-1}\right)_{\phi(n)}$.

4. R se guarda en secreto el par de números (d, n), lo cual es la llamada **clave privada**, y hace público el par de números (e, n), a los que llamaremos su **clave pública**.

5. E, que desea enviarle el mensaje confidencial x a R, lo encripta del siguiente modo:
$Enc(x) = (x^e)_n$
cosa que puede hacer, pues conoce los números e y n que R hizo públicos. Ahora, envía el número $Enc(x)$.

6. R recibe un número $y = Enc(x)$ y ejecuta con él la siguente operación:
$Des(y) = \left(y^d\right)_n$
cosa que puede hacer, pues él mismo sí conoce el valor de su propia clave privada, d. Lo que consigue es:
$Des(y) = \left((x^e)^d\right)_n = \left(x^{(ed)}\right)_n = (x)_n$, puesto que d y e eran inversos en módulo $\phi(n)$ (cuidado, no n), y por tanto ed es un cierto número t que pertenece a la clase $[1]_{\phi(n)}$, de lo que, según la última proposición demostrada, $\left(x^t\right)_n = (x)_n$. En resumen, R puede conocer el mensaje x que E le envió.

12.8.3 Aclaraciones^{MC}

Nos quedan todavía algunos puntos por aclarar que son necesarios para la comprensión total del algoritmo y de su uso. Concretamente:

1. Si el espía intercepta el número y, no puede ejecutar la operación de desencriptado porque no conoce d. La única forma en que pudiera conocerla es calculándola como la inversa de e (que sí es pública) en modulo $\phi(n)$, y eso no es posible porque no conoce $\phi(n)$. Aun así, el lector pensará que no es difícil conocer $\phi(n)$ dado n (que es público): se descompone n en factores primos, lo que daría $n = pq$, y se calcula $\phi(n)$ como $(p-1)(q-1)$. El problema está en que descomponer un número en factores primos es un algoritmo que se supone NP-completo (o sea, su complejidad es exponencial con el tamaño del número), y hoy día la descomposición de un número de 200 cifras llevaría del orden de un millón de años de cálculo incluso al más potente ordenador en uso. La clave de todo este algoritmo está precisamente aquí: calcular un número dada su descomposición en factores primos es trivial, pues basta multiplicar los factores. Sin embargo, hallar los factores dado el número es costosísimo.

2. p y q deben ser primos. ¿ Cómo sabemos que lo son?. Una posibilidad es efectivamente descomponerlos. Aun cuando hayamos visto que esto es en la práctica imposible para n, puede aún estar a nuestro alcance para p y q, que son mucho más pequeños. En cualquier caso, y aun si no fuera así, existen tests probabilísticos que deciden, en un

tiempo lineal con el tamaño del número, si éste es o no primo con ínfima probabilidad de equivocarse. La forma exacta de tales tests queda fuera del alcance de este curso.

3. En el tercer paso del algoritmo que ejecuta el receptor para generar su clave pública hemos dicho que dado $\phi(n)$ se escoge un número e primo relativo con él. Esto se puede hacer tomando e al azar, y calculando mediante el algoritmo de Euclides el máximo común divisor de e y $\phi(n)$. Si es 1, los números son primos entre sí; si no, deberemos escoger otro e. Como dijimos, el algoritmo extendido de Euclides es de coste lineal con el tamaño de e, y esto es por tanto factible.

4. En el mismo paso se calcula la inversa de e en módulo $\phi(n)$. Una posiblidad de calcular inversas es usar el algoritmo extendido de Euclides, el cual, dados e y $\phi(n)$, nos devuelve su **mcd**, que será 1, puesto que e y $\phi(n)$ fueron escogidos como primos entre sí, y también los dos números d y h tales que $d \cdot e + h \cdot \phi(n) = 1$. d será entonces inverso de e en módulo $\phi(n)$, dado que su producto con e es mñultiplo de $\phi(n) + 1$. Si la d devuelta por el algoritmo fuese negativa, podemos sumarle $\phi(n)$ para obtener el representante positivo más pequeño (canónico) de su clase, el cual por supuesto también da 1 al multiplicarlo por e. En principio, una posibilidad alternativa de calcular la inversa sería usar el teorema de Euler, con lo que $d = \left(e^{\phi(\phi(n))-1}\right)_{\phi(n)}$. No obstante, esto requeriría el cálculo de $\phi(\phi(n))$, y por tanto, la descomposición en factores primos de $\phi(n)$, lo que es impracticable, pues $\phi(n)$ es un número casi del mismo órden que n; según hemos argumentado en la primera aclaración, esto sería costosísimo.

5. Las operaciones de encriptado y desencriptado requieren la elevación de un número a una potencia, en módulo n, lo cual puede hacerse usando el algoritmo llamado de "multiplicar y elevar", el cual es como sigue:
Dados tres enteros positivos, a, b y n, se trata de calcular $\left(a^b\right)_n$. Para ello escribamos el exponente en el sistema binario, y llamemos k al número de bits que se deben usar para representarlo: $b = b_k \cdot 2^k + b_{k-1} \cdot 2^{k-1} + \ldots + b_1 \cdot 2 + b_0$. Entonces,

$$\left(a^b\right)_n = \left(a^{b_k \cdot 2^k + b_{k-1} \cdot 2^{k-1} + \ldots + b_1 \cdot 2 + b_0}\right)_n = \left(\left(a^{b_k}\right)^{2^k}\right)_n \cdot \left(\left(a^{b_{k-1}}\right)^{2^{k-1}}\right)_n \cdot \ldots \cdot \left(\left(a^{b_2}\right)^4\right)_n \cdot \left(\left(a^{b_1}\right)^2\right)_n \cdot \left(a^{b_0}\right.$$

Llamemos r al resultado de la operación, y rescribamos la igualdad anterior, donde por simplicidad, omitiremos el módulo n, sabiendo que todas las operaciones deben hacerse en dicho módulo (cosa posible puesto que como sabemos $(\alpha \cdot \beta)_n = (\alpha)_n \cdot (\beta)_n$):

$$r = \left(\left(a^{b_k}\right)^{2^{k-1}} \cdot \ldots \cdot \left(a^{b_2}\right)^2 \cdot a^{b_1}\right)^2 \cdot a^{b_0}$$

y a su vez esto se puede escribir leyéndolo de derecha a izquierda como:

$$r = r_0 \cdot \left(r_1 \cdot \left(r_2 \cdot \ldots \cdot (r_k)^2\right) \ldots\right)^2\Big)^2$$

siendo $r_k = a^{b_k}$ (el paréntesis más interno), $r_{k-1} = a^{b_{k-1}} \cdot r_k^2$ (el siguiente paréntesis, leyendo de dentro a fuera), $r_{k-2} = a^{b_{k-2}} \cdot r_{k-1}^2$, etc. con lo que el resultado es el producto de k factores, cada uno de los cuales se obtiene multiplicando el anterior por a^{b_i}, y elevando el resultado al cuadrado; de ahí el nombre de este algoritmo. Ahora bien,

notemos que a^{b_i} vale, o bien a, si el bit b_i es 1, o bien 1, si el bit b_i es cero, y en este último caso podemos omitir la multiplicación por 1. Además de eso, nótese que el único factor que no se eleva al cuadrado es el último que multiplicamos (el que corresponde al bit 0). Por ello, el algoritmo queda:

$r \longleftarrow 1$
`Desde i` $\longleftarrow$ `k hasta 0 hacer`
  `si` $(b_i = 1)$
    $r \longleftarrow (r \cdot a)_n$
  `finsi`
  `si` $(i \neq 0)$
    $r \longleftarrow (r^2)_n$
  `finsi`
`findesde`

Al ir realizando las operaciones en módulo n los resultados no son nunca mayores que n, y por tanto resultarán manejables en la representación numérica que estemos usando.

6. En la práctica, la operación de desencriptado se puede realizar de modo más eficiente guardando en la clave privada no sólo d, sino también p y q, lo que transforma esta operación en la resolución de un sistema de dos ecuaciones en congruencias con dos incógnitas; para ello se usa el resultado conocido como Teorema Chino de los Restos, cuya demostración queda fuera del alcance de este curso. En cualquier caso, nótese que se trata de un problema de eficiencia.

12.9 Bibliografía^{MC}

"Fonaments de matemàtica discreta", *J.M. Basart, J. Rifá, M. Villanueva*, primera edición, Universitat Autònoma de Barcelona, Servei de Publicacions, ISBN 84-490-08555-7.

"Criptografía y Seguridad en Computadores", tercera edición, versión 1.14, *Manuel Lucena, Universidad de Jaén*. Libro electrónico gratuito disponible en `http://www.di.ujaen.es/~mlucena/lcripto.`