EXPLORACIÓN MATEMÁTICA

Criptografía: Algoritmo RSA

Convocatoria: mayo 2019

INIDCE

1.- Introducción

Desde pequeño me han gustado las nuevas tecnologías, es por esto que decidí aprender a cerca del funcionamiento de los ordenadores e Internet. Según iba aprendiendo me daba cuenta que la base de todas las tecnologías son las matemáticas. Un día la aplicación de móvil de mensajería Whattsapp, empezó a mandar mensajes a sus usuarios que debían seguir unos pasos para hacer más seguras sus conversaciones, fue aquí donde me pregunté ¿Cómo se hace para que yo pueda mandar y recibir mensajes sin que nadie más pueda verlos?

Investigando me encontré con la criptografía que por definición es el arte de escribir con clave secreta o de un modo enigmático. El origen de la criptografía se remonta a los egipcios con los jeroglíficos, sin embargo, los primeros en utilizar realmente este arte fueron los romanos, para poder enviar información militar sin que pueda ser interceptada. En estos casos la criptografía es muy simple, ya sea desordenando las letras del mensaje o poniendo una serie de obstáculos para tapar unas y dejar ver otras letras y poder ver el mensaje al completo.

Actualmente, la criptografía se ha vuelto mucho más compleja. Con la aparición de radio, pero sobre todo de ordenadores, se ha avanzado mucho en este campo, sobre todo con fines militares o secretos de estado, pero, esta tecnología también se utiliza en cualquier vía de comunicación a través de Internet. Esta criptografía se basa completamente en las matemáticas, concretamente en la aritmética modular, que voy a explicar a continuación para ser capaces de entender cómo funciona el algoritmo RSA

2.- Aritmética modular. Congruencias. Teorema de la congruencia lineal

2.1.- Definición

Dados a, b $\in \mathbb{Z}$ y n $\in \mathbb{Z}^+$, se dice que a es congruente con b módulo n y se denota:
$a \equiv b \pmod{n}$ si $a = b + kn$ donde k $\in \mathbb{Z}$

O bien, $a \equiv b \pmod{n}$ si el resto de dividir a entre n es igual al resto de dividir b entre n

También se puede decir:

$a \equiv b \pmod{n}$ si (a-b) = k · n siendo k $\in \mathbb{Z}$, esto es, $n/(a-b)$

Para entenderlo mejor pongamos un ejemplo numérico. $23 \equiv 3 \pmod{5}$, ya que

23 = 3 + 4·5, o bien, 23-3 = 4·5, o bien 5 / (23-3)

Se puede deducir que

$$a = b + kn \implies a - b = kn \,/ k \in \mathbb{Z}$$

A la congruencia $a \equiv b \pmod{n}$ se le denomina congruencia lineal

2.2.- Proposición

La congruencia lineal $a \equiv b \pmod{n}$ es una relación de equivalencia

Demostración:

$a \equiv b \pmod{n}$ es una relación de equivalencia si verifica las propiedades siguientes:

i. Propiedad reflexiva: $a \equiv a \pmod{n}$
ii. Propiedad simétrica: Si $a \equiv b \pmod{n} \Longrightarrow b \equiv a \pmod{n}$
iii. Propiedad transitiva: Si $a \equiv b \pmod{n}$ y $b \equiv c \pmod{n} \Longrightarrow a \equiv c \pmod{n}$

Por ser la congruencia lineal una relación de equivalencia, se pueden agrupar en clases de equivalencia.

La clase de congruencia $a \equiv b \pmod{n}$, se denotará:

$[a]_n$ ó bien $[b]_n$ ya que son dos representantes de la misma clase de equivalencia.

Ejemplos:

- $[0]_5 = \{..., -10, -5, 0, 5, 10, ...\} = [5]_5 = [10]_5 = ...$

Se denota $\mathbb{Z}_n$ al conjunto de clases de congruencia módulo n

$\mathbb{Z}_n = \{[0]_n, [1]_n, ..., [n-1]_n\}$

En el ejemplo puesto anteriormente tenemos: $\mathbb{Z}_5 = \{[0]_5, [1]_5, [2]_5, [3]_5, [4]_5\}$, $[5]_5 = [10]_5$, $[6]_5 = [1]_5$

Propiedad 1:

Dados $a, b \in \mathbb{Z}$, $n \in \mathbb{Z}^+$

Si $a \equiv b \pmod{n}$ entonces se verifica que $t \cdot a \equiv t \cdot b \pmod{n}$ tal que $t \in \mathbb{Z}$

2.3.- Teorema de la congruencia lineal

Dados $a, b \in \mathbb{Z}$ y $n \in \mathbb{Z}^+$

La congruencia $ax \equiv b \pmod{n}$, x tiene solución si y solo si $d \equiv 0 \pmod{b}$ siendo d = m.c.d. (a, n), o lo que es lo mismo, d/b.

Corolario I: Si mcd (a, n) = 1, entonces $ax \equiv b \pmod{n}$ tiene una única solución.

Lema I: Dados $a \in \mathbb{Z}$ y $n \in \mathbb{Z}^+$

Si $d = \text{mcd}\,(a, n) = 1$ se verifica que:

$ai \pmod{n} \neq aj \pmod{n}$ $\forall$ i $\neq$ j con I > 0 y j < n

Demostración: por reducción al absurdo:

Supongamos lo contrario, que existen i > 0 y j< n con i ≠j tales que verifican $ai \equiv aj \pmod{n}$

Entonces: n / (ai-aj), lo que es lo mismo, n / a (i – j). Pero sabemos por la hipótsis del lema que d = mcd (a,n) = 1, entonces deducimos que:

n / (i – j), es decir, $i \equiv j \pmod{n}$

Como 0 < i, y j > n; entonces necesariamente i = j, llegando a una contradicción ya que i ≠ j

■ (c.q.d)

Teorema I:

Dados a, b ∈ $\mathbb{Z}$ y n ∈ $\mathbb{Z}^+$

Si mcd (a, n) = 1 entonces se verifica que $ax \equiv 1 \pmod{n}$.

Demostración:

Sabemos que $n \cdot n^{-1} = 1 \; \forall \; n \in \mathbb{R}$

Si multiplicamos por a ∈ $\mathbb{Z}$ todos los elementos de $\mathbb{Z}_n = \{[0]_n, [1]_n, \ldots, [n-1]_n\}$ y aplicando la propiedad 1 obtenemos el conjunto:

$\{[0]_n, [a]_n, [2a]_n, \ldots, [(n-1)a]_n\}$

Donde todos sus elementos son distintos aplicando el lema 1. Ahora bien, sigue siendo clases de congruencia módulo n, por ello los elementos de este nuevo conjunto pertenecen $\mathbb{Z}_n$, por lo que tiene que existir un valor igual a $[1]_n$, lo que implica que a ∈ $\mathbb{Z}$ tiene inversa módulo n, que será la solución de la congruencia lineal:

$$ax \equiv 1 \pmod{n}$$

■ (c. q. d)

Para hallar la solución de una congruencia lineal se emplea la identidad de Bézout

Identidad de Bézout: Sean a, b ∈ $\mathbb{Z} \setminus \{0\}$ y $d = \text{mcd}\,(a, b)$; existen x, y ∈ $\mathbb{Z}$ tal que:

$$ax + by = d$$

Siguiendo nuestra congruencia $ax \equiv 1 \pmod{n}$, su identidad de Bézout es:

$$ax + ny = 1$$

Por tanto, al resolver la identidad, la x es el inverso de a módulo n

Para entenderlo mejor, pongamos un ejemplo numérico, calcularemos 16^{-1} en $\mathbb{Z}_{81}$. Por lo que siguiendo la identidad de Bézout nos queda:

$$16x + 85y = \text{mcd}\,(16, 85)$$

Para resolver esto, utilizamos el algoritmo de Euclides:

$$85 = 16 \cdot 5 + 5$$

$$16 = 5 \cdot 3 + 1$$

Con esto demostramos que el mcd (16, 85) = 1. Tras esto sustituimos los restos de abajo arriba, por lo que:

$$1 = 16 - 3 \cdot 5 = 16 - 3 \cdot (85 - 16 \cdot 5) = 16 - 3 \cdot 85 + 16 \cdot 15 = 16 \cdot 16 - 3 \cdot 85$$

Con esto vemos que x = 16 e y = -3

2.4.- Teorema de Euler-Fermat

Llamaremos $\mathbb{Z}^*_n$ al conjunto de elementos $[p_i]$ invertibles de $\mathbb{Z}_n$. Por el Teorema I, p_i es coprimo con n

Volvamos con un ejemplo numérico, si n = 18; $\mathbb{Z}^*_{18} = \{[1], [5], [7], [11], [13], [17]\}$

2.4.1.- Función Φ de Euler

Se define la función Φ de Euler como el cardinal de $\mathbb{Z}^*_n$, es decir, el número de elementos $[p_i]$ en $\mathbb{Z}^*_n$

$$\Phi(n) = |\mathbb{Z}^*_n|$$

Esta función indica cuantos números menores que n hay coprimos con n, siguiendo con nuestro ejemplo; $\Phi(18) = |\mathbb{Z}^*_{18}| = 6$

Lema II: Dados dos números primos p y q se verifica que $\Phi(pq) = (p-1)(q-1)$

2.4.2.- Teorema de Euler-Fermat

Dados $a \in \mathbb{Z}$ y $n \in \mathbb{Z}^+$, si mcd (a, n) = 1, entonces $a^{\Phi(n)} \equiv 1 \pmod{n}$

3.- Algoritmo RSA

El nombre RSA viene dado por sus creadores en 1997 Rivest, Shamir y Adleman. Este algoritmo es el más utilizado debido a su sistema diferente al resto, lo normal era un sistema en el cual el código se encripta con una clave, y la misma clave sirve para desencriptarlo, el problema que esto supone es la transmisión de la calve, ya que cualquiera que pueda interceptar el mensaje, también interceptará la clave. Expliquemos el ejemplo que pone Diego Córdoba de las cajas fuertes en su blog *RSA: ¿Cómo funciona este algoritmo de cifrado?* Este sistema, también llamado simétrico es como si envías un mensaje dentro de una caja fuerte a alguien, y en el mismo paquete va la llave que lo abre, por lo que si alguien lo intercepta, puede abrir la caja sin problemas. El algoritmo RSA es asimétrico o de clave doble. Teniendo una clave que todos conocen y otra que sólo tu posees, sigamos con el símil de las cajas fuertes, este sistema sería una caja fuerte con dos llaves, una que la abre

y otra que la cierra, de tal modo que cuando alguien quiere enviarte un mensaje privado, tú le das la llave que cierra la caja, él mete el mensaje y e da la caja con la llave que la cierra, y solo tú tienes la llave que la abre, por lo que si alguien lo intercepta, tiene la caja cerrada con la llave que la cierra, por lo que no puede abrirla.

Es por esta seguridad que ofrece el algoritmo que se ha hecho el algoritmo más utilizado actualmente. Vamos a explicar matemáticamente como funciona este modo de encriptación

3.1.- Pasos del algoritmo

Como he dicho necesitamos una clave pública que será (n, e) y una privada que será (n, d)

1. Se eligen dos números primos p y q, cuanto mayores sean estos números mayor fortaleza tendrá el encriptado
2. Calculamos n, siendo $n = p \cdot q$
3. Escogemos e (clave pública), que debe ser primo relativo con $(p - 1) \cdot (q - 1)$
4. Calculemos d (clave privada) que será la inversa de e módulo $((p - 1) \cdot (q - 1))$. Siguiendo el Teorema I, sabemos que existirá d tal que:
   $d \cdot e \equiv 1 \pmod{(p - 1) \cdot (q - 1)}$)
5. Siendo M el mensaje a cifrar y C el mensaje cifrado:
   $$C = M^e \pmod{n}$$
6. Luego, para descifrar el mensaje:
   $$M = C^d \pmod{n}$$

Veamos ahora por qué el paso 6 nos devuelve el mensaje sin cifrar, es decir, veamos la demostración del paso 6:

En el paso 4 llegamos a:

$d \cdot e \equiv 1 \pmod{(p - 1) \cdot (q - 1)}$

Por la definición de congruencias sabemos que:

$d \cdot e = 1 + k\,(p - 1) \cdot (q - 1) \;/\; k \in \mathbb{Z}$

Por tanto:

$$(M^e)^d = M^{e \cdot d} = M^{1+k\ (p-1)\cdot(q-1)}$$

Aplicando el lema II: $(\Phi\, q) = (p\text{-}1)(q\text{-}1)$, y como $n = p \cdot q$, obtenemos:

$$M^{1+k\cdot(p-1)(q-1)} = M^{1+\,k\cdot\Phi(n)} = M \cdot (M^{\Phi(n)})^k$$

Por el teorema de Euler, si M y n son primos entre sí: $M^{\Phi(n)} \equiv 1 \pmod{n}$, entonces:

$$M \cdot (M^{M^{\Phi(n)}})^k \equiv M \cdot 1^k \pmod{n} \equiv M \pmod{n} \Rightarrow (M^e)^d \pmod{d}$$

Por lo tanto, como $C = M^e \pmod{n}$, tenemos:

$$M = C^d \pmod{n}$$

■ (c. q. d)

3.2.- Ejemplo de cifrado / descifrado RSA

Como ejemplo cifraré mi número favorito, el 18.

1. Se eligen dos números primos $p = 11$ y $q = 19$, deben ser mucho mayores, pero para que el ejemplo se haga más fácil y entendible he escogido uno muy pequeños
2. Calculamos $n = p \cdot q$; $n = 209$
3. Escogemos $e = 119$, primo relativo con $(p - 1) \cdot (q - 1) = 10 \cdot 18 = 180$
4. Calculamos la inversa de $e = 119$ módulo 180

$$d \cdot 119 \equiv 1 \pmod{180} \Rightarrow d = 59$$

5. Ciframos el mensaje $M = 18$

$$C = 18^{119} \pmod{209} \Rightarrow C = 151$$

6. Ahora para descifrar el mensaje:

$$M = 151^{59} \pmod{209} \Rightarrow M = 18$$

Así demostramos que el mensaje se puede descifrar y cifrar sin alterar el mensaje.

Estos cálculos han sido realizados con ayuda del programa ExpoCrip de la Universidad Politécnica de Madrid, de uso libre, mostrado en la figura 1. Este programa es capaz de hacer los cálculos con números mucho mayores.

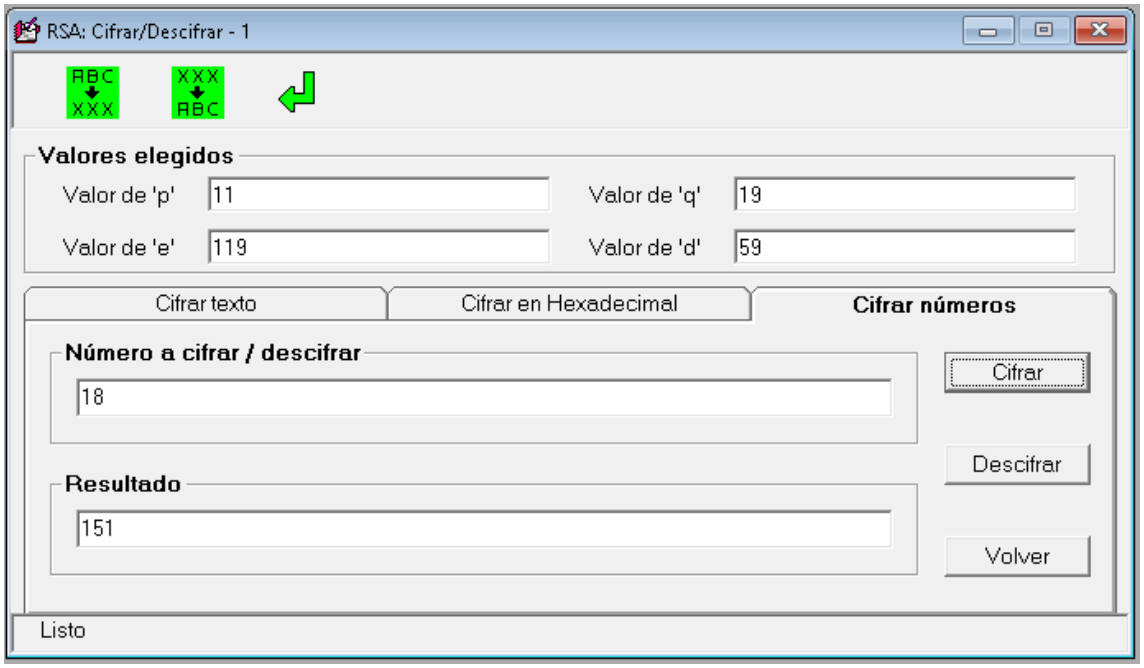

Figura 1. Programa ExpoCrip de la U.P.M

3.3.- Fortaleza del algoritmo RSA

Como he expuesto anteriormente, este algoritmo es muy seguro ya que en ningún momento se comunica la clave que desencripta todo el mensaje. Ya que para que nos envíen un mensaje cifrado, debemos comunicar la calve pública (n, e), así nos pueden cifrar M para obtener C. A nosotros nos llega el mensaje cifrado C y nosotros podemos obtener M a partir de C ya que tenemos la clave privada (n, d) que nadie más conoce, es más, el propio emisor del mensaje no sería capaz de descifrar su propio código. Ahora pensemos, nosotros hemos calculado d, ¿por qué no podrá calcularlo un interceptor?, porque para ello necesitan p y q, es cierto que tienen n, su producto, por lo que para obtener p y q les bastaría con factorizar n y ya podrían calcular d. Si bien factorizar n a mano es muy costoso, hay programas informáticos que son capaces de hacerlo en menor tiempo, aún haciéndolo en menos tiempo a través de un ordenador, este tiempo que tarda es demasiado grande, es inviable. Existe otro método para hallar claves denominado fuerza bruta que consiste en probar todas las claves posibles hasta lograr la adecuada, bien, este procedimiento es prácticamente inútil para este algoritmo ya que, lo recomendables es elegir p y q de unas 150 cifras, por lo que n tendrá 300, entonces, existen demasiadas combinaciones de números para encontrar el correcto con 300 cifras. Si bien este algoritmo actualmente es seguro y el más utilizado, existe una amenaza, los ordenadores cuánticos. El propio ExpoCrip tiene una opción de realizar ataques teniendo n, en nuestro caso, lo realiza muy velozmente ya que el número n es muy pequeño.

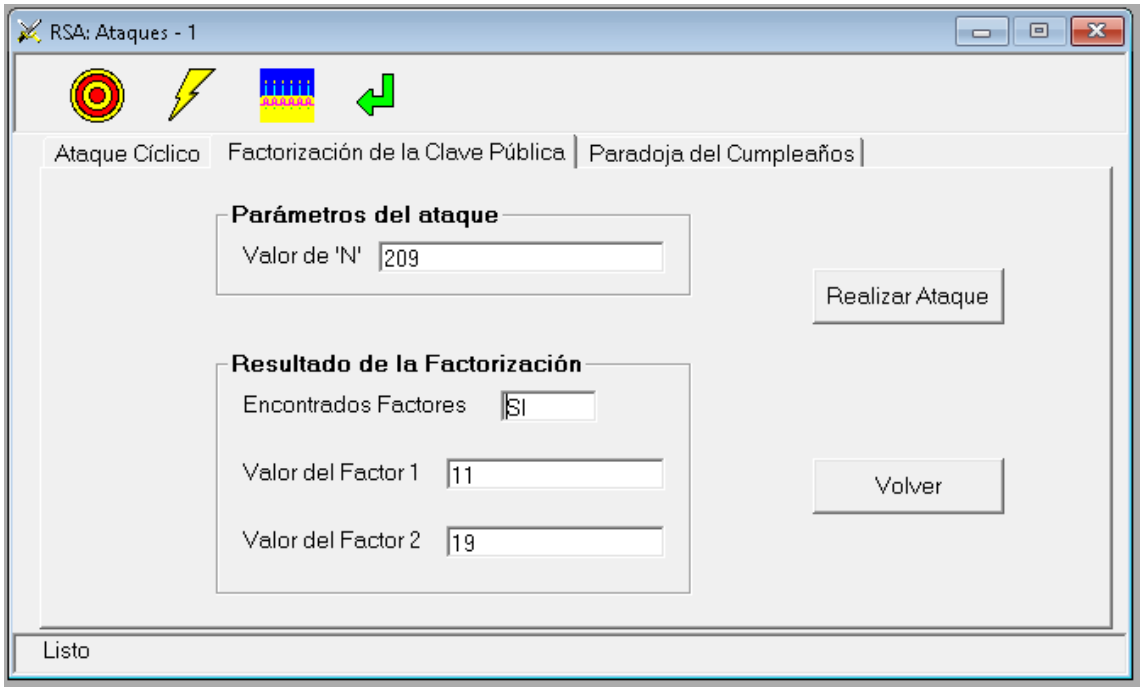

Figura 2. Programa ExpoCrip de la U.P.M

En 1977 uno de los creados del RSA, Rivest, calculó que con la tecnología de la época para factorizar un número de 129 cifras se necesitarían alrededor de $4 \cdot 10^{16}$ años para conseguirlo. En 1994 se intentó factorizar un número de 129 cifras y tardó 8 meses en hallar sus factores con gran despliegue de medios. Esto desvela que es posible su factorización, sin embargo, actualmente, el número con más cifras factorizado fueron 307 en 2010, gracias a 3 grandes universidades y muchos superordenadores, lo consiguieron tras 11 meses, lo que equivaldría a 100 años con un ordenador.

4.-Conclusión

El tema de la criptografía siempre me ha interesado y ahora en podido aprender cómo funciona el algoritmo más popular y más usado. Si bien es cierto que al principio me costó entenderlo ya que no conocía el campo de la aritmética modular. Por ello, antes de empezar tuve que hacer un estudio sobre este campo, una vez entendí el funcionamiento de la aritmética modular, me resultó sencillo el desarrollo de la criptografía.

La seguridad de este algoritmo, por el momento y con la tecnología actual es indiscutible que es bastante alta, pero, el desarrollo de ordenadores cuánticos presenta una amenaza ya que sería capaz de realizar cálculos muy complejos, con grandes cifras en un tiempo muy reducido, por lo que se hallarían los factores de n de forma muy rápida, haciendo viable el ataque por fuerza bruta.

5.- Bibliografía

CARMONA COLLADO, Luis Miguel, Departamento de Matemática Aplicada de la Facultad de Informática (U.P.M.)

http://www.dma.fi.upm.es/recursos/aplicaciones/matematica_discreta/web/aritmetica_modular/enteros.html [Consulta: 3 Noviembre 2018]

CÓRDOBA, Diego, *RSA: ¿Cómo funciona este algoritmo de cifrado?* Junco TIC

https://juncotic.com/rsa-como-funciona-este-algoritmo/ [Consulta: 18 Agosto 2018]

VERA LOPEZ, Francisco J. *Aritmética Modular*, Universidad de Murcia. Disponible en:

https://webs.um.es/pacovera/miwiki/lib/exe/fetch.php?id=inicio&cache=cache&media=aritmeticamod.pdf [Consulta: 13 Noviembre 2018]