

Trabajo Práctico Anual

de Matemática Discreta

Sección 2: Relaciones

 Ejercicio N° 1 : Programar un encriptador de palabras mediante el método RSA, o dos métodos de  
encriptación simples (en caso de no querer usar el RSA); donde dado un programa, se  
muestre al mismo encriptado, con la opción de recuperar el mensaje original.

**Introducción al sistema criptográfico RSA(Rivest, Shamir y Adlemar)**

**RSA** *(*[*Rivest*](https://es.wikipedia.org/wiki/Ronald_Rivest)*,* [*Shamir*](https://es.wikipedia.org/wiki/Adi_Shamir) *y* [*Adleman*](https://es.wikipedia.org/wiki/Leonard_Adleman)*)* es un [sistema criptográfico de clave pública](https://es.wikipedia.org/wiki/Criptografía_asimétrica) desarrollado en [1977](https://es.wikipedia.org/wiki/1977). Es el primer y más utilizado [algoritmo](https://es.wikipedia.org/wiki/Algoritmo) de este tipo y es válido tanto para cifrar como para [firmar digitalmente](https://es.wikipedia.org/wiki/Firma_digital).

La seguridad de este algoritmo radica en el problema de la [factorización](https://es.wikipedia.org/wiki/Factorización) de [números enteros](https://es.wikipedia.org/wiki/Número_entero). Los mensajes enviados se representan mediante números, y el funcionamiento se basa en el producto, conocido, de dos [números primos](https://es.wikipedia.org/wiki/Número_primo) grandes elegidos al azar y mantenidos en secreto.

Como en todo sistema de clave pública, cada usuario posee dos claves de cifrado: una pública y otra privada. Cuando se quiere enviar un mensaje, el emisor busca la clave pública del receptor, cifra su mensaje con esa clave, y una vez que el mensaje cifrado llega al receptor, este se ocupa de descifrarlo usando su clave privada.

Se cree que RSA será seguro mientras no se conozcan formas rápidas de descomponer un número grande en producto de primos.

**Idea del algoritmo**

Supongamos que Daniel quiero enviar un mensaje a Marina un mensaje secreto que solo ella pueda leer.

Marina envía a Daniel una caja con una cerradura abierta, de la que solo Marina tiene la llave. Daniel recibe la caja, escribe el mensaje, lo pone en la caja y la cierra con su cerradura(ahora Daniel no puede leer el mensaje) .Daniel envía la caja a Marina y ella la abre con su llave. En este ejemplo , la caja con la cerradura con la cerradura es la << clave publica >> , y la llave de la cerradura es su << clave privada >>.

**Generación de claves**

1. Cada usuario elige dos números primos **p** y **q**. Deben ser elegidos de forma aleatoria por razones de seguridad.
2. Se calcula **n = pq** . donde **n** se usa como modulo para ambas claves, publica y privada.
3. Se calcula **φ(n) = (p - 1)(q – 1)**, donde **φ** es la **función** **φ** **de Euler**.
4. Se escoge un entero positivo **e** menor que **φ(n)**, que sea coprimo con **φ(n)**. **e** es el exponente de la clave publica
5. Se determina **d** (mediante aritmética modelar) que satisfaga la congruencia

**e . d ≡ 1(mod φ(n)**) , es decir, que **d** sea elmultiplicador modular inverso de **e mod** **φ(n).**

Expresado de otra forma **ed -1** es dividido exactamente por **φ(n) = (p – 1 )(q - 1).**

Esto suele calcularse mediante el algoritmo de Euclides extendido.

**d** se guarda como el exponente de la clave privada.

La clave publica es **(n,e)** ,esto es, el modulo y el exponente de cifrado. La clave privada es **(n,d)**, esto es, el modulo y el exponente de descifrado, que debe mantenerse en secreto.

**Cifrado**

Ahora se convierte cada carácter del mensaje en un numero entero m menor a n mediante un protocolo reversible acordado de antemano. Luego calcula el texto cifrado **c** mediante la operación **c ≡ me (mod n)**

**Descifrado**

Se puede recuperar el mensaje (osea cada uno de los caracteres que lo componen) apartir de c usando su exponente d de la clave privada mediante el siguiente calculo:

**m ≡ cd (mod n)**

Ahora que tenemos **m** , se puede recuperar el mensaje original. El procedimiento anterior funciona porque **cd = (me)d ≡ med (mod n).**

Esto es asi porque, como hemos elegido **d** y **e** de forma que **ed** **= 1 + kφ(n)**, se cumple

**med = m1 + k φ(n) = m (m φ(n))k ≡ m (mod n)**

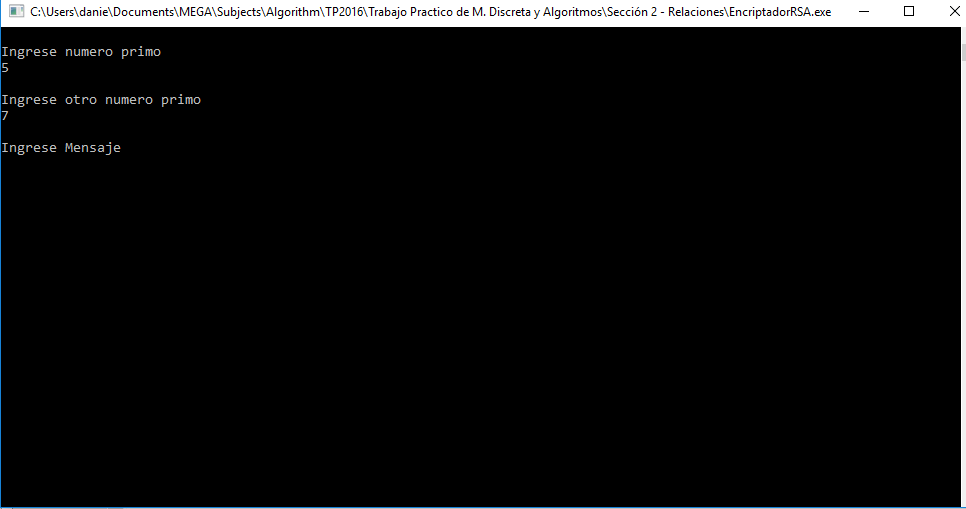
La ultima congruencia se sigue directamente del teorema de Euler cuando **m** es coprimo con **n**. Puede demostrarse que las ecuaciones se cumplen para todo **m** usando congruencias.

Esto muestra que se obtiene el mensaje original:

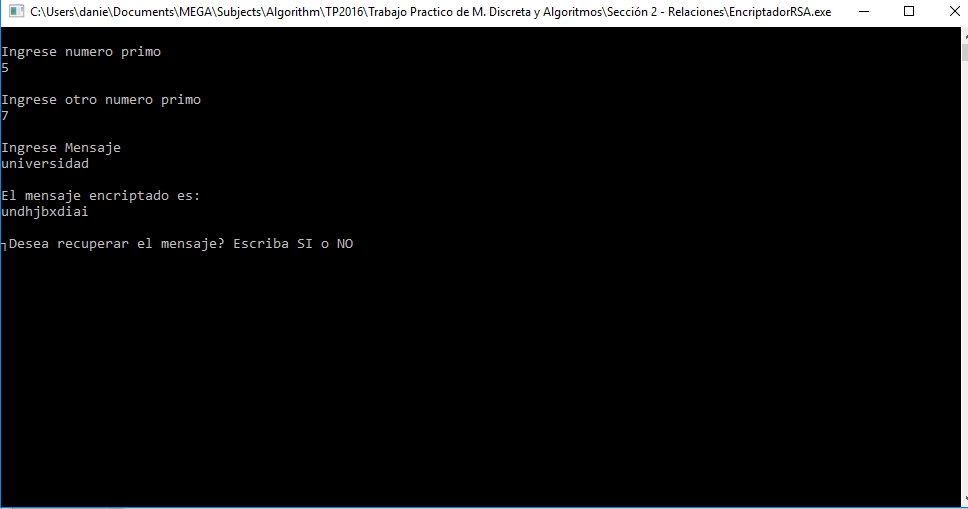
**m ≡ cd (mod n)**

**Manual de Uso**

1. Ingresamos dos números primos



1. Ingresamos el texto que queremos encriptar y mostramos el encriptado



1. Se nos sugiere recuperar el texto encriptado, si ingresamos ‘SI’ se lo recupera inmediatamente y se muestra en pantalla

