



MISTI
MAESTRÍA EN INGENIERÍA EN
SEGURIDAD Y TECNOLOGÍAS
DE LA INFORMACIÓN



INSTITUTO POLITÉCNICO NACIONAL

**ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD CULHUACÁN**

MAESTRÍA EN INGENIERÍA EN SEGURIDAD Y TECNOLOGÍAS DE LA INFORMACIÓN

IMPLEMENTACIÓN DEL ALGORITMO DE CIFRADO RSA

Presenta:

Jonathan Eduardo Garcá García

jgarcia1404@alumno.ipn.mx

Profesor:

Dr. Eduardo Vázquez Fernández

Fecha:

06/06/2022

<https://jon2g.github.io/WebSharpRSA/>

NOTA:

El programa esta publicado en la siguiente dirección:

<https://jon2g.github.io/WebSharpRSA/>

El código fuente puede consultarse en:

<https://github.com/Jon2G/WebSharpRSA>

Capturas de pantalla del programa en funcionamiento:



Algoritmo de cifrado RSA

Llaves de cifrado

Llave publica

Llave pública

Llave privada

Llave privada

Tamaño de la llave
512

GENERAR PAR DE LLAVES

Texto:
Texto:
Hola mundo

☒ Cifrar

CIFRAR

☐ LIMPIAR

1) Procedemos a generar el par de llaves:

256

512

1024

2048

GENERAR PAR DE LLAVES

Podemos escoger entre distintos tamaños para n



El par de llaves se muestra en formato Base64 :

Llave pública

Llave pública

```
-----BEGIN RSA PUBLIC KEY-----
A8HRuhYXFDS1hA0JeNZietg5huly/azdTeH5etEaWNd
3rOc2B3ILS/nf4o78zyJGE5+xxHID+IBZ
fewTMRnbEEF2Os1GSM8D0szy+HB165avhrdxDf3bN0
tMcIpt+NOAcfqk4D7YlrRY0stZIEZIfDmn
KVcYhjITU2jecmsZMd0t63/AG1rToj7zsr/1dNW/+rfIIB
gVSGtc2x8YiFmRYe9clUjtTSaR0SXJ
wcokX0rzRuxad2dIFSkMRDnw7D7FPDvscNzg0syH21
AJ8p0pj8jZb6IEhp1Ux/7LF8J0hk3BU7Rt
sfnFqud8caXOCIBZXSvctuZXbtXoeRawx5jVmwA=
```

Llave privada

Llave privada

[Redacted private key content]

Por defecto la llave privada esta censurada

Llave pública

Llave pública

```
-----BEGIN RSA PUBLIC KEY-----
A8HRuhYXFDS1hA0JeNZietg5huly/azdTeH5etEaWNd
3rOc2B3ILS/nf4o78zyJGE5+xxHID+IBZ
fewTMRnbEEF2Os1GSM8D0szy+HB165avhrdxDf3bN0
tMcIpt+NOAcfqk4D7YlrRY0stZIEZIfDmn
KVcYhjITU2jecmsZMd0t63/AG1rToj7zsr/1dNW/+rfIIB
gVSGtc2x8YiFmRYe9clUjtTSaR0SXJ
wcokX0rzRuxad2dIFSkMRDnw7D7FPDvscNzg0syH21
AJ8p0pj8jZb6IEhp1Ux/7LF8J0hk3BU7Rt
sfnFqud8caXOCIBZXSvctuZXbtXoeRawx5jVmwA=
```

Llave privada

Llave privada

```
sfnFqud8caXOCIBZXSvctuZXbtXoeRawx5jVmwA=
6e26EsazKlfpqe5U5OA3VEg9DUmb1hUje9/ANo
BE1iUtlglzFidC/jxim+dmBznBAEP0y1Be1yLe
9lxiYozz/JVZLIwhSkEEcdZkC4xPU2qKg+1yClc2
hC9sBJMO6VWI+wZSkMtTw0Nr7PUJI39xl3h
Hr5bCjaRzZiLoVvmCZGfrKPiC4vh9ksccxCdBLF
9wNdPVJLs1iK+06i+aHZC3JpIRc2jsRSi/teX
UF90Gnh8Jme6UNHD5P7K2rCb5MOKYhH/fcY
m10Str4GK7tDBKrbYn4mCiJl As7s02YHD9+8a7
```

Tamaño de la llave
2048

GENERAR PAR DE LLAVES

Cifrado

2) Ingresamos el texto que se desea cifrar en el recuadro

Texto:

Texto:

Hola mundo

1



Cifrar



CIFRAR



LIMPIAR

Y presionamos cifrar:

Texto:

Texto:

8rLdGdxK19yWYX1QXkP6WDChjxOr1FQSYMgtwdkXJEQe7jtOd0yn/tkjc3X3fYPj79DUTQcqzrgluOT1OwnYvVqSj7dzqy
kDlfl+WlIp1cbimN++HHxY2K7InGI20p8xUEQnEK2qRL3QGdt4izGrNnQXYOFwrf3WKfLS1U9P10qdufxsAgBUP4v4QUp
7GzllloAzkk0kv91HMGatcrhYEusIVtdSBcPSLeIw4rKZ7uresGEEz+A8gUgZdkcfi5guJlikFrnuswexfgKNowsh5kO1vpGx4
VlcJKR1eT2NpucFpXuyONsT2rvFI14OK6gKPCShYa7SVq5y5buKBfarDg==

2



Cifrar



CIFRAR



LIMPIAR

El texto cifrado se imprime en formato Base64

Descifrado

3) Alternamos el estado del switch bajo el texto para cambiar a modo de descifrado



Decifrar



DECIFRAR



LIMPIAR

4) Presionamos descifrar (La llave privada debe ser correcta para este paso)

Texto:

Texto:

Hola mundo

 Decifrar

 DECIFRAR

 LIMPIAR

Si todo es correcto se nos mostrará el texto original

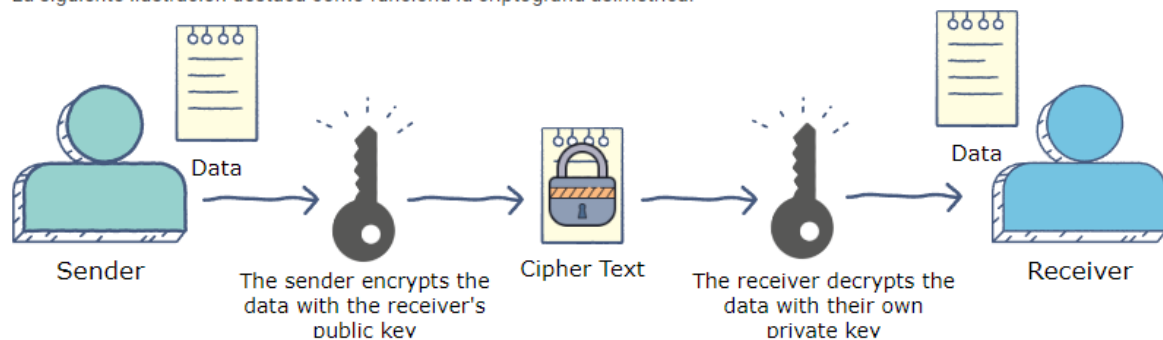
Teoría

El algoritmo de cifrado RSA

El algoritmo RSA es un algoritmo criptográfico asimétrico; esto significa que utiliza una clave pública y una clave privada (es decir, dos claves diferentes vinculadas matemáticamente). Como sugieren sus nombres, una clave pública se comparte públicamente, mientras que una clave privada es secreta y no debe compartirse con nadie.

El algoritmo RSA lleva el nombre de quienes lo inventaron en 1978: Ron Rivest, Adi Shamir y Leonard Adleman.

La siguiente ilustración destaca cómo funciona la criptografía asimétrica:



Teorema de Euler

El teorema de Euler es la base del algoritmo RSA, y es importante entenderlo antes de explicar el algoritmo.

La idea es que si tenemos dos números a y n tal que son coprimos, entonces: $a\phi(n) \equiv 1 \pmod{n}$

Donde ϕn es conocida como la función ϕ de Euler, y para números primos, es: $\phi p = p - 1$

Para un producto de dos números primos pq entonces $\phi(pq) = \phi(p)\phi(q)$

Generando las claves

1. Seleccione dos números primos grandes x, y .
Los números primos deben ser grandes para que a alguien le resulte difícil descifrar.
2. Calcular $n = x * y$
3. Calcular la función Phi $\phi(n) = (x - 1)(y - 1)$
4. Seleccionar un número entero e , tal que e sea coprimo con $\phi(n)$ y $1 < e < \phi(n)$
El par de números (n, e) conforman la llave pública.

Nota:

Dos enteros son coprimos si solo un el único entero positivo los divide es 1

5. Calcular d de tal forma que $e * d = 1 \bmod(\phi(n))$

El par de números n, d conforman la llave privada.

d puede ser encontrada utilizando el algoritmo extendido de Euclides

Cifrado

Cada caracter del texto plano debe convertirse en su representación decimal.

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

De modo que el texto completo es representado con el número: 1234567891011121314151617181920212223242526

$$C = P^e \bmod(n)$$

$$C = 1234567891011121314151617181920212223242526^e \bmod(n)$$

Donde C representa al texto cifrado.

Descifrado

Teniendo el número C que representa al texto cifrado hacemos:

$$P = C^d \bmod(n)$$

El resultado de P será el número que representa al texto original

Algoritmo de cifrado:

Implementación propuesta:

```
double x = 3;
double y = 7;

double n = x*y;

double e = 2;
double phi = (x-1)*(y-1);
while (e < phi)
{
    if (gcd(e, phi)==1)
        break;
    else
        e++;
}

BigInteger k
BigInteger limit = p * q;
for (BigInteger i = 1; i < limit; i += 2)
{
    if (i != p && i != q){
        if (i>1 && i < phi && phi % i != 0){
            k=i;
            break;
        }
    }
}

BigInteger d = (1 + (k*phi))/e;

BigInteger msg = 20;

// Cifrado
BigInteger c = BigInteger.Pow(msg, e);
c = BigInteger.Mod(c, n);

// Descifrado
BigInteger m =BigInteger.Pow(c, d);
m = BigInteger.Mod(m, n);

//Algoritmo extendido de euclides
int gcd(int a, int h)
{
    int temp;
    while (1)
    {
        temp = a%h;
        if (temp == 0)
            return h;
        a = h;
        h = temp;
    }
};
```