```
#include "RSA.h"

RSA::RSA()
{

    alfabeto = "ABCDEFGHIJKLMNOPQRSTUVWXYZ,.-(
)abcdefghijklmnopqrstuvwxyz<>*1234567890";
    generarclave();

}
void RSA::generarclave()
{
    srand(time(NULL));

    //asignamos a un string los primos
    ifstream leer;
    leer.open("Primeros_1000000_Primos.txt");//hay 2 archivos más, con
los primeros 10000, 100000, 1000000 de primos.(cambiar el numero por esos
de ser necesario)
    string primos;
    getline(leer,primos);
    leer.close();

    //calculando la cantidad de primos en el string
    ZZ siz=conv<ZZ>("0");
    ZZ tam = conv<ZZ>(primos.size());
    for(int i=0;i<tam;i++)
        if(primos[i]=='.')siz++;


    //escojemos de manera aleatoria 2 primos
    ZZ num_rand = fun.mod(conv<ZZ>(rand()),siz);
    ZZ privada1 = aleatorio(num_rand,primos,tam);
    num_rand = fun.mod(conv<ZZ>(rand()),siz);
    ZZ privada2 = aleatorio(num_rand,primos,siz);

    //calculamos las variables n y fi de n
    n=privada1*privada2;
    ZZ fi_n=(privada1-1)*(privada2-1);

    //clave privada
    num_rand = fun.mod(conv<ZZ>(rand()),siz);
    privada=fun.mod(aleatorio(num_rand,primos,siz),n);
    while(fun.mcd(privada,fi_n)!=1)
        privada=fun.mod(aleatorio(num_rand,primos,siz),n);

    //clave publica
    publica=fun.inv_mult(privada,fi_n);
}
ZZ RSA::cifrar(string mensaje,ZZ clave,ZZ n){

    //Posición en alfabeto y convercion de int a ZZ
    ZZ pos = conv<ZZ>(alfabeto.find(mensaje));

    // Cifrado
    return fun.pow(pos,clave,n);

}
```

```
string RSA::descifrar(ZZ mensaje){

    //Descifrado
    ZZ pos = fun.pow(mensaje,privada,n);

    //Converción de ZZ a string
    ostringstream aux;
    aux << alfabeto[conv<int>(pos)];
    string cifrado = aux.str();

    return cifrado;
}

ZZ RSA::aleatorio(ZZ rand,string primos,ZZ tam){

        //escoje el primo en posición rand y lo guarda en base
        string base="";
        for(int i=0;rand!=0;i++){
                if(primos[i]=='.')
                        rand--;
                if(rand==0)
                    for(int j=i-1;primos[j]!='.';j--)
                        base=primos[j]+base;
            }

        //Converción de string a ZZ
        istringstream aux(base);
        ZZ primo;
        aux >> primo;
        return primo;

}
```