## ANEXO C. CRIPTOSISTEMA RSA

La siguiente es la descripción del funcionamiento del algoritmo RSA, la cual implica un alto conocimiento de teorías matemáticas de grupos, el algoritmo de Euclides y función módulo. Para profundizar más en el tema se recomienda referirse a libros especializados en este tema o a docentes del área de matemáticas avanzadas [3] [4].

1. Cada usuario ***U*** elige dos números primos (actualmente se recomienda que tales números primos tengan más de 200 dígitos) *p* y *q* y calculan *n = p . q.* El grupo a utilizar por el usuario ***U*** es, entonces, $Z_n^*$. El orden de este grupo es *n* y el del grupo $Z_n^*$ es $\phi(n)=\phi(p.q)=(p-1)(q-1)$. Para U es fácil calcular este orden, pues conoce *p* y *q*.

2. Después, U selecciona un entero positivo e, $1\le e\le\phi(n)$, de modo que sea primo con el orden de l grupo $Z_n^*$, es decir, tal que $mcd(e,\phi(n))=1.$

3. Mediante el algoritmo de Euclides extendido calcula el inverso de e en $Z_{\phi(n)}^*$, d; se tiene entonces $e.d\equiv 1(\mathrm{mod}\,\phi(n))$, con $1\le d\le\phi(n)$.

La clave pública del usuario U es la pareja $(n,e)$, mientras que su clave privada es el número *d*. Por supuesto, también deben permanecer secretos los números *p*,*q* y $\phi(n)$.

Si un usuario A desea enviar un mensaje *m* de $Z_n$ a otro usuario B, utiliza la clave pública de B, ($n_b$,$e_b$), para calcular el valor de $m^{e_b}(\mathrm{mod}\,n_b)=c$, que envía a B.

Para recuperar el mensaje original, B calcula $c^{d_b}=\left(m^{e_b}\right)^{d_b}=m^{e_b d_b}\equiv m(\mathrm{mod}\,n_b)$.