Funzionamento dell'algoritmo RSA

Ad ogni lettera dell'alfabeto associamo un numero da 0 a 25; il messaggio da cifrare *'x'* sarà quindi una successione finita di numeri: $x=(x_1, x_2, ..., x_r)$.

Gli spazi su cui operiamo sono $P=K=Z_n$ (Z_n è l'insieme quoziente di Z rispetto alla relazione di congruenza; per maggiori informazioni clicca su aritmetica modulare) dove *n* è un numero naturale tale che $n=p\cdot q$, con *p* e *q* due numeri primi aventi ciascuno almeno un centinaio di cifre.

Nota: l'aver preso dei numeri primi *p* e *q* molto grandi, ci assicura il fatto che essi non saranno numeri compresi tra 0 e 25, per cui varrà sicuramente la condizione: $M.C.D.(x_i, n)=1 \; per \; i=1,2, ..., r$ che mi garantisce la validità del Teorema di Eulero.

Scegliamo un numero intero *a* tale che $M.C.D.(a, \varphi(n))=1$, dove *φ(n)* è la funzione di Eulero calcolata in *n*.

Dato che *a* è primo con *φ(n)*, si può trovare grazie all'algoritmo euclideo un numero intero *b* tale che $a\cdot b\equiv 1 \; mod \; \varphi(n)$.

Il cifrario RSA è quindi così costituito:

$$P=C=Z_n$$
$$K_1 \; chiave \; pubblica: \{n, a\}$$
$$K_2 \; chiave \; privata: \{n, a, p, q, b\}$$
$$cifratura: \; e(x)\equiv x^a \; mod \; n$$
$$decifratura: \; d(y)\equiv y^b \; mod \; n$$

Verifichiamo la bontà del cifrario. Data una lettera x_i del messaggio *x*, se la cifriamo e poi la decifriamo dovremo riottenere x_i , cioè $x_i \rightarrow e(x_i) \rightarrow d(e(x_i))\equiv x_i \; mod \; n$. Studiamo:

$$d(e(x_i))\equiv(e(x_i))^b \; mod \; n, \; ma \; (e(x_i))^b\equiv((x_i)^a)^b \; mod \; n, \; da \; cui, \; per \; l'algoritmo \; eucideo, \; vale$$
$$((x_i)^a)^b\equiv(x_i)^{(1+k\cdot\varphi(n))} \; mod \; n; \; infine, \; per \; il \; Teorema \; di \; Eulero, \; (x_i)^{(1+k\cdot\varphi(n))}\equiv x_i \; mod \; n$$

Il cifrario è quindi ben definito perché risulta: $d(e(x_i))\equiv x_i \; mod \; n$