Note sulla correttezza di RSA e sulla complessità degli attacchi

P. Bonatti

21 novembre 2016

1 Richiami elementari di algebra

Elevamento a potenza di binomi

Ricordiamo la definizione di coefficiente binomiale:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}\,.$$

Ora dati tre interi a, b e c, con $c > 0$,

$$(a+b)^c = \binom{c}{c}a^c + \binom{c}{c-1}a^{c-1}b + \cdots + \binom{c}{c-k}a^{c-k}b^k + \cdots + \binom{c}{0}b^c\,. \quad (1)$$

L'equazione (1) può essere facilmente dimostrata per induzione su c.

Semplici richiami di algebra modulare

Dati tre interi i, j e n, diciamo che i *è congruo a* j *modulo* n, in simboli

$$i \equiv j \bmod n\,,$$

se $(i \bmod n) = (j \bmod n)$, cioè il resto della divisione intera per n è lo stesso per i e per j. In altre parole se

$$i = q_i n + r_i \quad (2)$$
$$j = q_j n + r_j \quad (3)$$

dove $r_i < n$ e $r_j < n$, allora $i \equiv j \bmod n$ significa che $r_i = r_j$. Notare che se $j < n$ allora $i \equiv j \bmod n$ implica che $j = (i \bmod n)$.

La congruenza è ovviamente una relazione di equivalenza, che viene rispettata dalle classiche operazioni aritmetiche. Per esempio possiamo moltiplicare entrambi i membri dell'equivalenza per una stessa quantità:

$$\text{se } i \equiv j \bmod n \text{ allora per ogni } k,\ ki \equiv kj \bmod n\,. \tag{4}$$

La dimostrazione di (4) è pressochè immediata. Rappresentando i come in (2), dove q_i è il quoziente della divisione di i per n e r_i è il resto, abbiamo che $ki = kq_in + kr_i$. Il primo termine non contribuisce al resto della divisione di ki per n, essendo un multiplo di n, quindi il resto della divisione sarà uguale a $kr_i \bmod n$. Analogamente, usando (3), il resto della divisione di kj per n deve essere $kr_j \bmod n$. Siccome per ipotesi i e j sono congruenti, vale $r_i = r_j$ e di conseguenza anche $(kr_i \bmod n) = (kr_j \bmod n)$, e questo dimostra che $ki \equiv kj \bmod n$. (QED)

Approssimativamente parlando, tutti i numeri congruenti si comportano allo stesso modo. Ad esempio tutti gli interi congruenti a 1 sono elementi neutri per la moltiplicazione:

$$\text{se } i \equiv 1 \bmod n \text{ allora per ogni } j,\ j \cdot i \equiv j \bmod n\,. \tag{5}$$

La dimostrazione di (5) è semplicissima: se q_i e q_j sono i quozienti delle divisioni intere di i e j per n, e se r_i e r_j sono i rispettivi resti, allora valgono (2) e (3). Inoltre, se vale $i \equiv 1 \bmod n$, allora $r_i = 1$. Come conseguenza di (2) e (3),

$$j \cdot i = q_jq_in^2 + q_jn \cdot r_i + r_j \cdot q_in + r_jr_i\,.$$

In questa equazione, i primi tre termini sono multipli di n e non contribuiscono al resto. L'ultimo, cioè r_jr_i, per ipotesi è uguale a $r_j \cdot 1 = r_j < n$. Quindi $(j \cdot i \bmod n) = r_j = (j \bmod n)$ da cui, per la definizione di congruenza, otteniamo $j \cdot i \equiv j \bmod n$ (QED).

Come nell'algebra classica, gli interi congruenti a 1 restano tali anche dopo un qualunque elevamento a potenza:

$$\text{se } i \equiv 1 \bmod n \text{ allora per ogni } c,\ i^c \equiv 1 \bmod n\,. \tag{6}$$

Questo si dimostra velocemente formulando i in termini di quoziente e resto: $i = q_in + r_i$. Quindi, applicando (1) a questo binomio, otteniamo

$$i^c = \binom{c}{c}(q_in)^c r_i^0 + \cdots + \binom{c}{1}(q_in)^1 r_i^{c-1} + \binom{c}{0} r_i^c\,. \tag{7}$$

In questa somma solo l'ultimo termine non è esplicitamente un multiplo di n, quindi è l'unico a poter contribuire con un valore non nullo al resto della divisione di i^c per n. Inoltre, essendo $\binom{c}{0} = 1$ (per definizione) e $r_i = 1$ (per l'ipotesi che $i \equiv 1 \bmod n$) abbiamo $\binom{c}{0} r_i^c = 1$. Concludiamo quindi che $(i^c \bmod n) = 1$, ovvero $i^c \equiv 1 \bmod n$. (QED)

Per dimostrare la correttezza di RSA ci servirà anche la seguente proprietà:

$$(i \bmod n)^c \bmod n = i^c \bmod n\,. \tag{8}$$

Proviamo la sua correttezza similmente alla proprietà precedente. Esprimiamo i^c nella forma (7). I primi c termini contengono tutti un fattore n quindi non contribuiscono al resto della divisione di i^c per n. L'ultimo termine è $\binom{c}{0} r_i^c = r_i^c$. Quindi il resto della suddetta divisione è uguale a $r_i^c \bmod n$, cioè vale

$$(i^c \bmod n) = (r_i^c \bmod n)\,.$$

Ora ci basta notare che r_i è proprio $(i \bmod n)$, per definizione. Sostituendo r_i con $(i \bmod n)$ nell'equazione qui sopra otteniamo proprio (8). (QED)

Infine avremo bisogno del *Teorema di Eulero*: per ogni intero i coprimo con n (cioè tale che $\mathrm{MCD}(i, n) = 1$):

$$i^{\,\Phi(n)} \equiv 1 \bmod n\,.$$

Noi utilizzeremo questo teorema in una forma equivalente in cui $i^{\,\Phi(n)}$ viene prima elevato a un qualunque esponente k (resta congruo a 1 grazie a (6)), poi entrambi i lati dell'equivalenza sono moltiplicati per i (la congruenza è rispettata grazie a (4)). Il risultato di questa trasformazione è

$$i^{\,k\cdot\Phi(n)+1} \equiv i \bmod n\,. \tag{9}$$

2 Prova di correttezza di RSA

In RSA la cifratura e la decifratura di un messaggio $m < n$ con le chiavi (e, n) e (d, n) si effettuano così:

$$\begin{aligned} c &= m^e \bmod n\,, && (c \text{ è il messaggio cifrato}) \\ m' &= c^d \bmod n\,. \end{aligned}$$

Ovviamente la procedura è corretta se $m' = m$, ed è questo che vogliamo dimostrare qui.

Facciamo un primo tentativo. Sostituiamo la definizione di c in quella di m'. Otteniamo:

$$\begin{aligned} m' &= (m^e \bmod n)^d \bmod n && \text{che per la (8) è uguale a} \\ &= m^{ed} \bmod n && (10) \\ &= m^{k\Phi(n)+1} \bmod n && \text{(per qualche } k\text{)} \quad (11) \\ &= m\,. && (12) \end{aligned}$$

L'eguaglianza tra (10) e (11) è giustificata dalla definizione delle chiavi: ricordate che sono scelte in modo che $ed \equiv 1 \bmod \Phi(n)$, quindi per qualche intero k, $ed = k \cdot \Phi(n) + 1$.

L'eguaglianza tra (11) e (12) è giustificata dal Teorema di Elulero nella forma (9), applicata ad $i = m$. Bisogna ricordare, però, che il Teorema di Eulero vale solo se m è coprimo con n, quindi le eguaglianze qui sopra sembrano dimostrare la correttezza di RSA *solo per i messaggi m tali che $MCD(m, n) = 1$*. Ovviamente questo non è sufficiente: è necessario poter cifrare *qualunque* $m < n$.

È questa la ragione per cui in RSA n viene scelto come prodotto di *esattamente* due numeri primi p e q: sotto questa ipotesi la (9) vale per *tutti* gli $i < n$ e di conseguenza la dimostrazione qui sopra prova la correttezza di RSA per tutti i possibili input $m < n$. Nel seguito dimostreremo questa generalizzazione di (9) provando che (11) e (12) sono uguali per tutti gli $m < n$.

Come già detto, se $\mathrm{MCD}(m, n) = 1$ allora l'eguaglianza vale per il Teorema di Eulero, quindi a noi resta solo da dimostrarla per i casi in cui $\mathrm{MCD}(m, n) > 1$. Assumiamo dunque che sia così.

Dato che i fattori primi di n sono p e q, questi sono anche i soli divisori possibili in comune tra m e n, quindi, se $\mathrm{MCD}(m, n) > 1$, m deve essere multiplo di p o di q, ma *non di entrambi*, altrimenti $m \geq p \cdot q = n$, mentre per definizione RSA accetta solo messaggi $m < n$. Qui dimostriamo che (9) vale sotto l'ipotesi che m sia un multiplo di p, diciamo

$$m = h \cdot p\,. \tag{13}$$

Non riporteremo la dimostrazione per il caso in cui m è multiplo di q perchè è perfettamente simmetrica e si ricava sostituendo p con q e viceversa.

Notate che q è primo, quindi m potrebbe avere divisori in comune con q solo se fosse un multiplo di q, ma abbiamo già escluso questa eventualità

(che implicherebbe $m \geq n$). Di conseguenza, $\mathrm{MCD}(m, q) = 1$ e il teorema di Eulero si può applicare, garantendo che $m^{\Phi(q)} \equiv 1 \bmod q$. Questo, mediante la (6), implica che per qualunque esponente x,

$$m^{x \cdot \Phi(q)} \equiv 1 \bmod q \,. \tag{14}$$

Scegliendo $x = k \cdot \Phi(p)$, per un qualunque intero k, otteniamo dalla (14): $m^{k \cdot \Phi(p)\Phi(q)} \equiv 1 \bmod q$, da cui, sfruttando il fatto che $\Phi(p)\Phi(q) = (p-1)(q-1) = \Phi(n)$, deriviamo

$$m^{k \cdot \Phi(n)} \equiv 1 \bmod q \,. \tag{15}$$

Questo significa che per qualche intero ℓ,

$$m^{k \cdot \Phi(n)} = \ell q + 1 \,.$$

Moltiplicando entrambi i lati per m, che per (13) è uguale ad hp, otteniamo:

$$m^{k \cdot \Phi(n)+1} = \ell q (hp) + m \,.$$

da cui, riordinando i fattori e ricordando che $n = pq$:

$$\begin{aligned} m^{k \cdot \Phi(n)+1} &= \ell h (pq) + m \\ &= \ell h \cdot n + m \end{aligned}$$

che implica

$$(m^{k \cdot \Phi(n)+1} \bmod n) = m \,,$$

il che è quanto dovevamo dimostrare.

Analisi di complessità di alcuni attacchi a RSA

Si può procedere calcolando, a partire dall'informazione disponibile,

- i fattori primi p e q di n;
- direttamente $\Phi(n)$;
- direttamente d.

Discutiamo brevemente questi tre metodi a partire dall'ultimo.

Ricavare d da una firma $f = h(m)^d \bmod n$ (dove h, m e n sono noti) o da $c^d \bmod n = m$ (dove chi cifra conosce m, c e n) significa calcolare un

logaritmo discreto, problema per cui non sono noti algoritmi polinomiali. Notare che si tratta di attacchi di tipo "known plaintext".

Alternativamente, ricavando direttamente $\Phi(n)$ e sfruttando e, che fa parte della chiave pubblica, si può calcolare efficientemente d con lo stesso algoritmo utilizzato per la generazione delle chiavi; inoltre anche n è noto, essendo parte della chiave pubblica; quindi conoscendo $\Phi(n)$ si può ricavare efficientemente la chiave privata (d, n).

Infine ottenendo p e q (ad esempio fattorizzando n) si può calcolare $\Phi(n) = (p-1)(q-1)$ in tempo polinomiale. Da $\Phi(n)$ e dalla chiave pubblica si ottiene poi la chiave privata come descritto in precedenza.

Quest'ultimo punto evidenzia che il calcolo di $\Phi(n)$ non è più difficile della fattorizzazione di n (a meno di un costo polinomiale). Nel seguito mostriamo che in realtà i due attacchi hanno la stessa complessità mostrando come ricavare p e q da $\Phi(n)$ in tempo polinomiale.

Supponiamo quindi di conoscere $\Phi(n)$. Per definizione, quando n è il prodotto di due primi p e q,

$$\Phi(n) = (p-1)(q-1) = pq - p - q + 1 = n - (p+q) + 1$$

da cui si ottiene

$$p + q = n + 1 - \Phi(n)\,. \tag{16}$$

Si osservi che tutti i termini nella parte destra di questa equazione sono noti, quindi anche il valore di $(p+q)$ è noto. Ora si osservi che

$$\begin{aligned} (p+q)^2 &= p^2 + q^2 + 2pq \\ (p-q)^2 &= p^2 + q^2 - 2pq = (p+q)^2 - 4pq = (p+q)^2 - 4n\,. \end{aligned} \tag{17}$$

La parte destra della (17) è composta da tutti termini noti, quindi se ne ricava il valore di $(p-q)$ estraendone la radice quadrata.

Ora che sia $p+q$ sia $p-q$ sono noti, è facile ottenere p e q:

$$\begin{aligned} p &= \frac{(p+q)+(p-q)}{2} \\ q &= (p+q) - p\,. \end{aligned}$$

In questo modo da $\Phi(n)$ si calcolano p e q in tempo polinomiale.[1] Questa completa la dimostrazione che la complessità computazionale del calcolo di $\Phi(n)$ è pari a quella della fattorizzazione di n (a meno di un polinomio).

[1] Non è costante perchè si deve operare con interi di lunghezza illimitata.