

**FATTORIZZAZIONE VELOCE (POLINOMIALE),
PROBLEMA P=NP, RH E (IN)SICUREZZA DELLA
CRITTOGRAFIA RSA**

Abstract

In this work we show as Speed factoring, RH end RSA are connected bay P=NP millennium problem.

Riassunto

In questo secondo ma breve lavoro sulla fattorizzazione veloce (Rif.1), riprenderemo la relazione tra fattorizzazione veloce come possibile problema in P, poiché riteniamo vera la RH (Rif.2 e 3), e la estendiamo anche alla crittografia RSA, poiché se la RH è vera la fattorizzazione veloce è un problema in P e quindi la crittografia RSA non è più sicura.

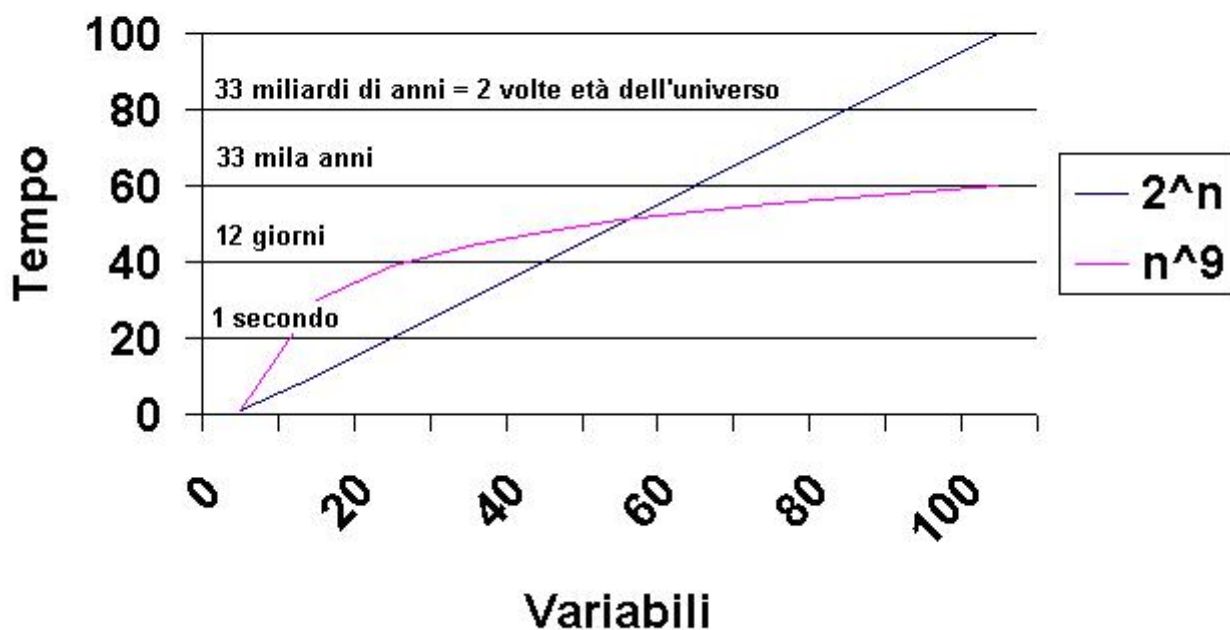
Occorre quindi complicare un po' o rinnovare la crittografia RSA, per allungare enormemente i tempi di calcolo)

Qui riportiamo l'articolo di del Prof. Luigi Salemi:

sul sito <http://www.visainformatica.it/3sat>

- [Lo Spirito della Prova 24.09.10](#)
- [Dimostrazione 11.09.10](#)
- [Spirit of the Proof 24.09.10](#)
- [Proof 13.09.10](#)
- [Eseguiibile 15.09.10](#)
- [Sorgenti Pascal Source 15.09.10](#)
-

Tempi di esecuzione x n. Variabili



In figura il confronto tra 2 algoritmi che lavorano in tempo $O(2^n)$ e $O(n^9)$

rispetto al n. delle Variabili ipotizzando che entrambi siano capaci di esaminare 1.000.000 di casi al secondo. Per motivi evidenti i tempi sono riportati in scala logaritmica.

Se siete arrivati qui probabilmente sapete già la differenza tra la classe dei problemi "P" e "NP", se vi siete persi e mi avete raggiunto per errore vi dico che nella classe P sono contenuti i problemi per i quali si conosce un algoritmo che li risolve in tempo Polinomiale, mentre nella classe NP sono contenuti i problemi per i quali si conosce solo un algoritmo di risoluzione in tempo Esponenziale (ma beffardamente l'algoritmo di verifica lavora in tempo Polinomiale). Il grafico chiarisce in modo immediato quanto grande sia la differenza, in relazione ai tempi, tra le 2 tipologie di algoritmi.

Ciò che il grafico non dice è che i problemi più interessanti (es.: quello del commesso viaggiatore o dei percorsi minimi, quello dello zaino o sub somma, la scomposizione in fattori primi [su cui si basa quasi tutta la crittografia esistente]) sono nella classe NP. La buona notizia è che ogni tanto un problema da NP si trasferisce in P perché si trova un algoritmo più efficiente che lavora in tempo Polinomiale, e questo il caso della "verifica di primarietà" che nel 2002 si è trasferito in P per merito di 3 matematici indiani Manindra Agrawal, Neeraj Kayal e Nitin Saxena.

Da un bel po' di anni si cerca di provare se le classi P e NP siano effettivamente distinte (ovvero esiste almeno un problema in NP che mai si potrà trasferire in P) o se viceversa le 2 classi in realtà coincidono, ma noi non siamo stati ancora capaci di trovare l'algoritmo unificatore, quello per cui ogni problema di NP si possa risolvere in tempo Polinomiale

Leonid Levin e Stephen Cook hanno scoperto separatamente, all'inizio degli anni '70, che tutti i problemi della classe NP si possono ricondurre ad un unico problema denominato "SAT" in cui occorre risolvere una espressione booleana trovando, se esiste, una n-upla di valori True/False che soddisfi la espressione. Come dire risolto SAT

risolti tutti, peccato che anche SAT sia un problema della classe NP.

Subito dopo si è visto che SAT si può ricondurre a "3SAT", un problema in cui bisogna trovare, se esiste, la soluzione di una espressione booleana che è formata dalla congiunzione di Clausole; ogni Clausola essendo composta dalla disgiunzione di esattamente 3 (da cui il nome 3SAT) Variabili booleane eventualmente negate. Es: $(A1 \text{ or } \sim A2 \text{ or } A3) \text{ and } (\sim A1 \text{ or } \sim A3 \text{ or } A4) \text{ and } ..$

Penso di avere trovato un Metodo che risolve in tempo Polinomiale ogni problema 3SAT. Se ho ragione allora $P=NP$ e questo comporta qualche conseguenza negativa (la crittografia tradizionale diventa non sicura), ma soprattutto tante positive in svariati campi della scienza e della tecnica. Penso che i risultati della ricerca vanno condivisi in tempo reale, da qui l'idea di realizzare questa pagina di segnalazione. Ogni commento è ben accetto.

[Luigi Salemi](#)

Commento

Pensiamo che il prof. Luigi Salemi abbia proprio ragione, (e quindi i suoi algoritmi potrebbero in seguito far rientrare la fattorizzazione da NP in P) anche per un altro motivo: poiché pensiamo proprio che la RH sia vera (Rif. 2 e 3), e se la RH è vera la fattorizzazione sta in P, la crittografia RSA non è più sicura (vedi Nota finale), e ancora per la nostra congettura

sui numeri RSA (Rif 4), ulteriormente perfezionabile.

Quindi i gestori del sistema RSA farebbero bene e presto a migliorarlo o meglio ancora a sostituirlo con un altro ancora più efficiente, e possibilmente inattaccabile anche dai futuri computers quantistici (10 000 più veloci rispetto ai computer attuali); una soluzione tampone sarebbe quella di usare numeri primi p e q con qualche milione di cifre anziché poche centinaia come si è fatto finora.

Riferimenti

- 1) Fattorizzazione veloce e problema $NP = P$),**
- 2) Dai multipli di 6 alla Riemann Hypothesis**
- 3) La funzione $\sigma(n)$ e la RH**
- 4) Congettura sui numeri RSA**
- 5) Appunti sulla fattorizzazione del numero RSA – 190**

Caltanissetta 27.12.2010

NOTA. (Da “La fattorizzazione veloce e il problema P=NP”

in sezione “Articoli sulla Fattorizzazione” del nostro sito

www.gruppoeratostene.com ”):

“...Circa il problema del millennio $P = NP$ (del quale la fattorizzazione è uno tra il migliaio di casi simili), esso è vero se la fattorizzazione fosse un problema di tipo NP – completo (questo però non si sa ancora bene), e se anche la RH fosse vera, poiché in questo caso esisterebbe un polinomio per la fattorizzazione veramente veloce. Tale polinomio potrebbe anche essere il nostro oppure un suo ancora più efficiente derivato), ancora però da scoprire (ma il nostro potrebbe già spianare la strada) oppure ancora un altro polinomio su basi diverse, ma anch’esso ancora tutto da scoprire. Il vero problema della fattorizzazione dei numeri RSA, rimane ancora la grandezza dei medesimi , di centinaia di cifre decimali e almeno 2048 bit, considerati sicuri). Con il nostro polinomio la difficoltà passa però dal numero di cifre di N (chiave pubblica) e dei due numeri primi p e q (chiave privata), al numero di cifre della semidifferenza, molto più piccolo, e quindi con minore tempo di calcolo (Rif. e citazioni nel Mathbuilding dell’Ing. Rosario Turco, si veda la sezione Link del nostro sito)...”.