

I possibili pericoli per la crittografia RSA

(e i relativi rimedi)

.....

Gruppo Eratostene

Abstract

In this paper we show some dangers (new polynomial algorithms, quantum computers) for RSA cryptography

Riassunto

In questo lavoro citeremo diversi articoli sui possibili pericoli attuali e futuri (nuovi algoritmi polinomiali basati su nuove scoperte matematiche sui numeri primi, i non tanto futuri computer quantistici, ecc.) per la crittografia RSA; e, ove possibile, indicheremo qualche rimedio nel caso che l'uno o l'altro di tali pericoli la mettesse in serie difficoltà.

Introduzione

In questo lavoro riporteremo brani di lavori che adombrano pericoli per la crittografia RSA. Alla fine proporremo qualche possibile rimedio a tali veri o presunti pericoli, al fine di salvarla, consentendole di

assolvere il suo importante compito di garantire comunicazioni riservate, siano esse di natura economica, militare, ecc.

a) Cominceremo con il **Prof. Umberto Cerruti**, il quale sostiene che la fattorizzazione veloce (polinomiale), sottoproblema del problema del millennio $P = NP$, è indipendente dall'ipotesi di Riemann (**RH**, altro importante problema del millennio), e potrebbe anche avere ragione:

Riportiamo dal suo sito:

www.dm.unito.it/~cerruti/luglio04-gennaio28.html :

- **"18 Settembre 2004**

Congettura di Riemann e sicurezza mondiale

Sul numero di settembre di Le Scienze c'è un interessante articolo di Graham P. Collins sulla dimostrazione, da parte del matematico russo Grigory Perelman, della famosa congettura di Poincaré. Si tratta - come viene spiegato sulla stessa rivista - di uno dei sette "problemi del millennio", pubblicati sul sito del

[Clay Mathematics Institute](http://www.claymath.org/).

Per ognuno di essi viene offerto al solutore un premio di un milione di dollari.

Tra i magnifici sette appare, e non poteva essere altrimenti, la **congettura di Riemann**, più frequentemente detta *ipotesi di Riemann*. Il matematico Louis de Branges de Bourcia, che lavora presso la Purdue University, sostiene di avere provato la verità dell'ipotesi di Riemann, ma la comunità degli esperti non è completamente convinta della validità della dimostrazione.

Sembra, dai commenti della stampa internazionale, che la possibilità dell'esistenza della dimostrazione, faccia **paura**. Qualche esempio:

[Solution to arcane maths problem leads to doomsday projections](#)

11 Settembre 2004, The Times of India. ([Copia locale](#))

[Math problem solved=No Net transaction safe](#)

7 Settembre 2004, Hindustan Times ([Copia locale](#))

[Maths holy grail could bring disaster for internet](#)

7 Settembre 2004, Guardian ([Copia locale](#))

Vorrei cercare di spiegare, per quanto possibile brevemente e in modo semplice, da dove nascono questi timori e fino a che punto essi siano giustificati.

Dalla matematica salvezza e perdizione

Tutto il commercio elettronico ed i rapporti tra banche si fondano sulla *crittografia*. Quando, per esempio, acquistiamo qualcosa in rete, veniamo inseriti in un canale particolare, sicuro. Il male intenzionato che riuscisse ad intercettare il documento che inviamo al server non sarebbe in grado di decifrare il numero della nostra carta di credito, perché esso è stato crittato. Tutti i codici classici si basano su una chiave segreta. Il vostro computer ed il server ospite si scambiano una chiave, che viene creata sul momento, nuova ogni volta. Per passarsi la chiave occorre evidentemente un protocollo di trasmissione che metta in cifra il messaggio *senza* avere a sua volta bisogno di effettuare uno scambio di chiavi, altrimenti si entrerebbe in un circolo vizioso. I metodo usati in questa prima ed essenziale fase vengono detti *cifrari a chiave pubblica*. Nel caso di un cifrario a chiave pubblica, chiunque può conoscere la chiave: non riuscirà lo stesso a rompere il codice.

In sostanza, tralasciando diverse altre componenti troppo tecniche, in una transazione commerciale on-line vengono utilizzati due codici:

uno standard (si dice simmetrico) che necessita di una chiave *segreta* ed è molto veloce, l'altro a chiave pubblica, in genere più lento perché esegue calcoli pesanti. Per esempio, come codice simmetrico si potrebbe usare l'AES (Advanced Encryption Standard) e come codice a chiave pubblica l'RSA (acronimo dei nomi dei suoi inventori: Rivest, Shamir e Adleman). Il lettore che desideri chiarimenti in proposito può consultare, in questo stesso sito, le pagine su [Crittografia e numeri primi](#).

E' chiaro che se si rompe il codice a chiave pubblica, il quale custodisce la chiave del codice simmetrico, tutto viene a cadere.

Uno degli algoritmi più usati è proprio lo RSA. La sua forza si basa sul fatto che

- 1) E' facile trovare numeri primi grandi**
- 2) E' difficile fattorizzare un intero prodotto di due numeri primi grandi**

Ovviamente il termini "facile", "difficile" e "grande" sono relativi alla conoscenza scientifica e alla tecnologia disponibili. Attualmente sono grandi primi di alcune centinaia di cifre decimali. Facile rappresenta un tempo di secondi e difficile di secoli. Concludendo: l'intero sistema di sicurezza mondiale entrerebbe in gravi difficoltà se qualcuno trovasse un algoritmo veloce per la fattorizzazione di

interi.

Che cosa c'entra tutto questo con la congettura di Riemann?

Sentiamo le parole del professor Marcus du Sautoy, autore di un bestseller pubblicato in Italia da

Rizzoli, con il titolo "L'enigma dei numeri primi":

L'intero commercio elettronico dipende dai numeri primi. Io ho descritto i primi come atomi: ciò che manca ai matematici è uno spettrometro di primi. I chimici hanno una macchina che, se ci mettete una molecola, vi dice di quali atomi è composta. I matematici non hanno inventato una versione analoga di essa. Se l'ipotesi di Riemann è vera, essa non produrrà di per sé uno spettrometro. Ma la sua dimostrazione potrebbe farci capire meglio il funzionamento dei numeri primi, e quindi la dimostrazione potrebbe essere trasformata in qualcosa che potrebbe produrre questo spettrometro di primi. Se ciò accadrà, metterà in ginocchio l'intero commercio elettronico nello spazio di una notte.

Si tratta di affermazioni prudenti, piene di "se" e "potrebbe". Ma la stampa, con la sua solita sensibilità, si è lanciata con entusiasmo nell'impresa di spaventare gli utenti, prospettando un futuro prossimo in cui utilizzare bancomat e carte di credito sarà impossibile.

In uno degli articoli citati si legge:

Fino a che i numeri primi erano considerati casuali, potevano essere utilizzati senza problemi nelle moderne applicazioni di cifratura dei dati, che vanno dalle transazioni bancarie e il commercio elettronico all'uso delle carte di credito e al trasferimento di denaro su Internet. Una volta che la casualità sia provata falsa, però, finirà la cuccagna, nessun codice sarà più sicuro e nessuna transazione sarà protetta da intrusioni fraudolente. Alcuni pensano che per l'economia questo potrebbe essere l'equivalente di un asteroide che colpisca la Terra.

Siamo all'apocalisse! La matematica ha creato il paradiso della sicurezza informatica, dei codici inviolabili, e ora vuole distruggerlo!

Ma, alla fine, che cosa dice mai questa congettura di Riemann ?

- ...Dunque, certamente la successione dei primi non è casuale. Però, a
-
- partire da essa si generano sequenze con un comportamento
-
- *apparentemente* casuale, e la RH fornisce informazioni asintotiche
-
- sulla oscillazione di queste quantità.

Per concludere dirò che ritengo assai improbabile che la dimostrazione della RH possa portare a grandi rivoluzioni nel mondo delle comunicazioni riservate. Questo avverrebbe se - per esempio - facilitasse la fattorizzazione degli interi. Occorre però ricordare che da anni e anni vengono studiate *tutte le possibili implicazioni logiche* della RH...

- Il problema della fattorizzazione è diverso. Essa avviene o non

- avviene, semplicemente. Che io sappia nessuno ha ricavato metodi di
-
- fattorizzazione dalla ipotesi che RH sia vera. Non esiste un teorema
-
- del tipo
-
- **Se vale RH allora, dato un intero N , faccio**
-
- **questo e questo e lo fattorizzo velocemente.**
-
- Se esistesse lo si potrebbe usare: se N non si fattorizza nel tempo
-
- previsto la RH è falsa; se N si fattorizza siamo felici e rompiamo il
-
- codice RSA, anche *senza* avere dimostrato la RH. teorema può
-
- avere un enunciato elegante ed essere esteticamente bello; può
-
- avere una dimostrazione difficile e profonda, ed essere per questo
-
- di grande interesse matematico. Però la sua *forza*, la sua efficacia,
-
- non sta nella dimostrazione ma nelle sue possibilità di applicazione,
-
- nelle conseguenze che da lui si possono trarre, e queste dipendono
-
- solo dalla sua verità.

b) Libro Problemi del Millennio (Keith DEVLIN, “I Problemi del millennio” (Longanesi) pagina 166:

“Il problema dell’ apertura del codice per la crittografia RSA non è noto come NP completo (e probabilmente non lo è) quindi forse se ne potrebbe

sviluppare una soluzione senza dimostrare che $P = NP$. Viceversa, la dimostrazione di quell' entità implicherebbe immediatamente che il problema dell' apertura del codice potrebbe essere risolto in tempo polinomiale mettendo pertanto in discussione l' intero sistema di sicurezza su Internet. Poiché attualmente non conosciamo alcun modo che garantisca la sicurezza di comunicazioni aperte in internet senza dipendere dall' effettiva impossibilità di risolvere un problema NP, l' attuale dipendenza delle economie occidentali da comunicazioni elettroniche sicure in Internet non fa che confermare quanto sia alta la posta legata alla verifica di $P = NP$ ”

e a pag.168:

“...D'altra parte, come ho detto all'inizio di questo capitolo, fra tutti i problemi del Millennio, l'enigma $P = NP$ è quello che ha maggiori probabilità di essere risolto da un dilettante sconosciuto: non solo è relativamente facile capire che cosa dica il problema,, ma è anche possibile che per risolverlo basti semplicemente un' unica buona idea originale...”

Potrebbe essere il caso dell'Ing. Luigi Salemi, con i suoi lavori su 3SAT?
(Vedi paragrafo dell'Ing. Luigi Salemi)

Secondo il prof. Bottazzini, invece,

“Il problema della scomposizione di un numero in fattori sta in N_p , ma non

si a se stia anche in P (la risposta è positiva se l' ipotesi di Riemann è vera”

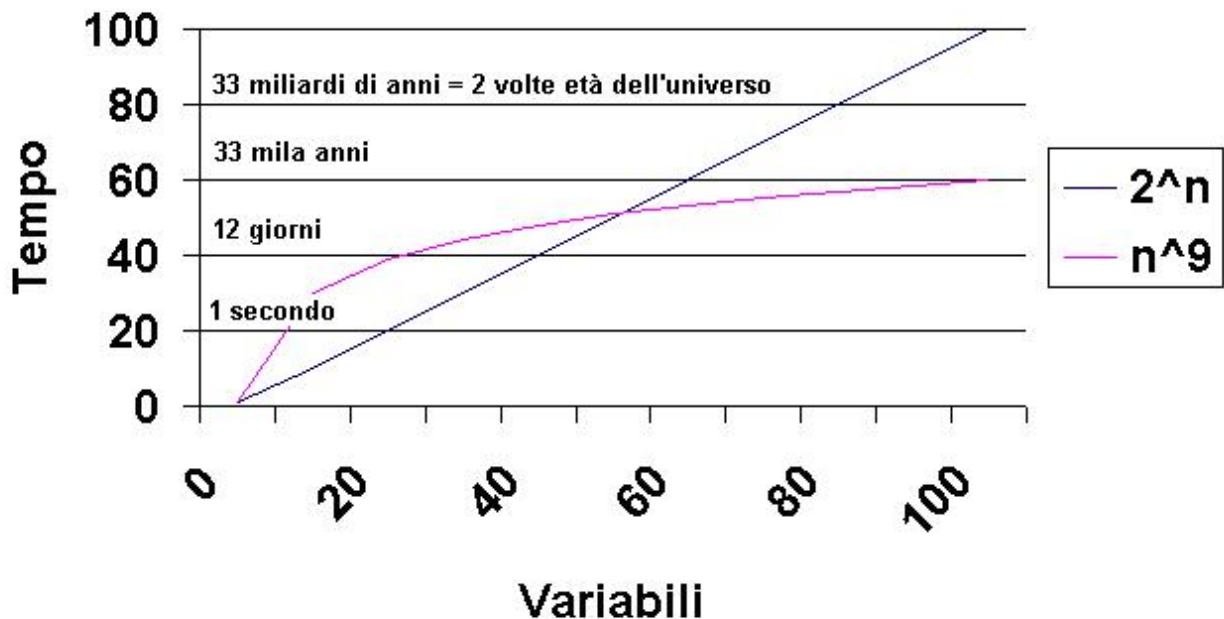
(in “Goldbach e le altre ipotesi tutte da dimostrare”, su “Il sole – 24Ore del 20 maggio 2000)

Poiché secondo noi la RH è vera (vedi “Articoli sui Problemi del Millennio” sul nostro sito www.gruppoeratostene.com), tale problema, che noi chiamiamo fattorizzazione veloce, sta in P, cioè deve esistere un algoritmo che risolva il problema in tempo polinomiale, e quindi questo è un pericolo per la crittografia RSA. Vedi successivo paragrafo c)

c) Ing. Luigi Salemi.

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

Tempi di esecuzione x n. Variabili



Download

Pagina 3SAT by Luigi Salemi

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

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 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: $(A_1 \text{ or } \sim A_2 \text{ or } A_3) \text{ and } (\sim A_1 \text{ or } \sim A_3 \text{ or } A_4) \text{ and } ..$ 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 accolto.

[Luigi Salemi](#)

d) Prof. Gerardo Iovane:

Brani tratti da “Fiera della conoscenza – 10-12-dicembre 2009 L’Aquila:

“ ASPETTANDO LA FIERA DELLA CONOSCENZA

La scoperta di Gerardo Iovane, lo studioso che ha rivoluzionato la teoria dei numeri primi, al convegno di venerdì 11 dicembre sulla sicurezza informatica presso la Scuola della Guardia di Finanza a Coppito

Comunicato Stampa del 5/12/2009

Una grande passione per le Scienze Esatte e l’Innovazione Tecnologica diventata oggi la sua professione, l’amore per l’interdisciplinarietà

scientifica che lo ha portato a curiosare nelle teorie evoluzionistiche e nella genetica e che lo ha messo di fronte a una scoperta eccezionale nel contesto dell'Info - Security e della Matematica, che per la prima volta verrà illustrata a L'Aquila, venerdì 11 dicembre, nell'ambito della Fiera della Conoscenza.

È Gerardo Iovane, professore associato di Analisi Matematica presso l'Università di Salerno. 37 anni, una laurea in Fisica Nucleare e Subnucleare, due dottorati di ricerca in Fisica e Matematica e lo studio elaborato nel 2008 e 2009 e pubblicato sulla prestigiosa rivista scientifica internazionale, specializzata sulla teoria del complessità e del caos, denominata Chaos, Solitons and Fractals, edita dalla Elsevier. Studio che rivoluziona la teoria computazionale dei numeri primi e le conseguenze in ambito della sicurezza delle informazioni e comunicazioni, attraverso una scoperta senza precedenti, ovvero quella che la sequenza dei numeri primi non è casuale, ma deterministica, che la loro natura non è fine a sé stessa, ma addirittura ha una dinamica naturale, appartiene a una specie, si organizza in famiglie. La teoria dei numeri primi fino ad oggi ha fatto da riferimento e base ad algoritmi applicati nell'ambito della trasmissione dei dati in tutto

il mondo, la sua “spiegazione” nell’accezione che ne dà il professor Iovane, di colpo rende vulnerabili i più sofisticati sistemi informatici e di security intelligence per chi approda a questa consapevolezza.

Tale scoperta e i suoi effetti saranno l’argomento centrale del convegno “Le nuove frontiere sulla sicurezza: vulnerabilità dei sistemi informatici e strumenti di security intelligence” (alle 10 – Sala Leonardo), promosso dai Servizi Innovativi di Confindustria Italia, sponsor E - Security – Gruppo Eltag Datamat. Oltre al professor Iovane, si confronteranno sul tema **Franco Silvi, Francesco Alfieri ed Ennio Lucarelli**, motori del settore di Confindustria; **Luisa Franchina**, delegata del Governo all’identificazione delle Infrastrutture Critiche Europee, **Domenico Santececca**, Direttore ABI, **Mario D’Intino, Mirco Ramazzo e Pietro Gentini** di E - Security e del Gruppo Eltag, Datamat e **Domenico Vulpiani**, dirigente generale della Polizia di Stato e Consigliere Ministeriale per la Sicurezza Informatica. Sicurezza e vulnerabilità stanno alla base dell’analisi proposta dal convegno, che è incentrato sull’utilizzo dei dati in campi molto avanzati e strategici e su quanto sia importante e complesso tutelarne la sicurezza nei contesti attuali.

“A minare la sicurezza delle informazioni ci possono essere cause esterne

come i diversi malware usati in cyber crime – illustra il professor Gerardo Iovane – oppure endogene, relative ai modelli matematici di riferimento. La scoperta che la sequenza dei numeri primi non risponde a schemi casuali, può creare un effetto domino capace di evidenziare la vulnerabilità degli algoritmi che stanno alla base degli attuali sistemi di sicurezza di riferimento a livello internazionale e della protezione delle infrastrutture critiche di un paese. La scoperta è solo l’inizio di un cammino che consentirà di affrontare tali fragilità e sviluppare dei modelli ancora più sofisticati e quindi più sicuri che diverranno una nuova base di partenza” Una scoperta resa possibile “rileggendo” la teoria dei numeri primi con gli occhi e le leggi della genetica. “Da qui il messaggio positivo che ne sta alla base – sottolinea il professor Iovane – perché se è grazie alle leggi della vita che il determinismo della sequenza dei numeri primi è stata compreso, per rendere più sicuri i futuri scenari dell’informazione bisognerà ricorrere a nuovi approcci che partono dall’uomo stesso e dalla scienze che lo studiano e lo misurano, come la biometria, l’information fusion, i sistemi dinamici complessi”.

In altre parole se l’uomo è stata la chiave per decifrare l’enigmatica sequenza dei numeri primi, l’uomo attraverso i suoi dati biometrici, può

essere una parte di nuove chiavi per cifrare le informazioni e renderle più sicure....”

Possibile rimedio: usare i dati biometrici come nuove chiavi di cifratura...

e) **Prof. Andrea Pasquinucci:**

dal sito www.ucci.it/docs/ICTSecurity-2002-01.pdf

1. **“ ICT Security n.1 Maggio 2002 p. 1 di 3**

La crittografia a chiavi pubbliche è a rischio?

Il mondo della sicurezza informatica è a rumore e vi sono molte discussioni fra esperti ed appassionati sullo stato della crittografia a chiave pubblica; la domanda più pressante è se è ancora sicuro utilizzare protocolli quali l’RSA. Cercherò qui di fare il punto della situazione e dare qualche informazione che aiuti a capire quello che sta succedendo.

Lo scorso novembre, il Professor Daniel Bernstein dell’Università dell’Illinois (Chicago), pubblicò un articolo scientifico (reperibile all’indirizzo <http://cr.yo.to/papers.html#nfscircuit>) in cui proponeva dei miglioramenti all’algoritmo usato per trovare i numeri primi, cioè non divisibili per altri numeri interi, divisori di un grande numero intero. La

“fattorizzazione in numeri primi di un numero intero” è un difficile problema matematico per il quale non si conosce alcun algoritmo efficiente quando il numero intero da fattorizzare è grande, ed anzi più il numero è grande più l’algoritmo è lento, in modo esponenziale. La maggior parte dei protocolli di crittografia a chiave pubblica, come il famoso RSA, sono basati sul fatto che con i computer attuali la fattorizzazione di un grande numero intero può richiedere tempi in media anche dell’ordine di migliaia di anni. La crittografia a chiave pubblica è utilizzata in molti tra i protocolli crittografici di maggior uso, come ad esempio https, ssh, IPSec, s/mime, pgp etc..

Con il suo articolo Bernstein in realtà proponeva alla National Science Foundation americana di finanziare una ricerca per stabilire quanto più velocemente si possano fattorizzare i numeri primi utilizzando le modifiche da lui proposte. E’ da notare che si tratta solamente di miglioramenti agli algoritmi noti, e non di un nuovo algoritmo che permette di decifrare facilmente, ad esempio, tutte le chiavi RSA. (In termini più tecnici, non cambia l’andamento esponenziale nel numero di bit della chiave del “costo” richiesto per la decifrazione, si ridurrebbe solamente l’esponente.) Inoltre la proposta di Bernstein richiede la

costruzione di circuiti appositi, possibile con le attuali tecnologie, ma dal costo odierno dell'ordine delle centinaia di milioni se non miliardi di dollari.

Con l'algoritmo tradizionale di fattorizzazione è stata decifrata recentemente una chiave RSA a 512 bit in 6 settimane utilizzando una rete di computer commerciali da ufficio (comunicazione a "Financial Cryptography 2002" di Nicko van Someren, nCipher **ICT Security n.1 Maggio 2002** p. 2 di 3 Inc.,UK).

Pertanto le chiavi simmetriche a 512 bit sono oggi considerate non sicure. Con l'algoritmo tradizionale le chiavi a 1024 bit sono però ancora sicure, ovvero non decifrabili in un tempo ragionevole con alcun computer esistente ora o nel prossimo futuro. Se i miglioramenti proposti da Bernstein all'algoritmo di fattorizzazione si rivelassero efficienti, le stime più pessimistiche indicano che le chiavi a 1024 bit potrebbero essere decifrate in qualche minuto o decina di secondi, utilizzando l'hardware costruito appositamente. Le chiavi RSA a 2048 bit sarebbero però ancora sicure per molti anni, anche tenendo conto della legge di Moore secondo cui la velocità dei computer raddoppia ogni 18 mesi.

In ogni caso questi sviluppi non sono inaspettati. Ad esempio già nel 1995

Bruce Schneier in *“Applied Cryptography”* prevedeva che nell’anno 2000 le più brevi chiavi RSA considerate sicure sarebbero state a 1024 bit, e nel 2005 a 1280 bit. Sugeriva inoltre a chi avesse particolari esigenze di sicurezza, di utilizzare a partire dal 2005 chiavi non inferiori a 2048 bit.

Quali conseguenze possono avere per le applicazioni pratiche di sicurezza informatica questi risultati? Le opinioni sono molto discordanti, posso però fare qui qualche osservazione. In primo luogo sono ben pochi i sistemi di sicurezza informatica ove l’anello debole è dato dalla lunghezza della chiave a crittografia pubblica, anche se essa è a 512 bit. I maggiori problemi sono di solito dati da vulnerabilità nel software (anche crittografico), ed errate progettazioni, implementazioni o gestioni dei sistemi di sicurezza.

Una seconda osservazione è che i software crittografici attualmente in commercio, e penso ad esempio ai browser web con SSL, raramente permettono all’utente o all’amministratore di richiedere che le chiavi crittografiche utilizzate non siano inferiori ad un certo numero di bit. Alla luce di quanto detto, questa possibilità diventerà alquanto importante nel prossimo futuro. Infine va ricordato che non vi è la certezza matematica che non esista un algoritmo “veloce” per la fattorizzazione dei numeri

interi, in termini matematici non è stato dimostrato che un problema “NP–HARD” non ammette una soluzione “P”. Pertanto vi è una piccolissima possibilità che qualcuno scopra un algoritmo “veloce” che renda del tutto insicura la crittografia basata sulla fattorizzazione in numeri primi. Inoltre vi è la possibilità che un giorno i “Quantum Computers” diventino realtà e siano in grado di fattorizzare qualunque numero intero in pochi secondi. Queste possibilità sembrano però molto lontane nel futuro, e quindi praticamente trascurabili nelle **ICT Security n.1 Maggio 2002** p. 3 di 3 applicazioni pratiche di oggi.

(Vedi successivo paragrafo sui computer quantistici, N.d.A.A.)

Per quanto riguarda la lunghezza delle chiavi a crittografia pubblica da utilizzare, se si sta progettando un sistema di sicurezza informatica è bene partire con chiavi a 1024 bit tenendo conto che, fra un paio di anni, potrebbe essere necessario fare un aggiornamento a chiavi tra i 1280 e 2048 bit. Poiché il “costo” di utilizzo di una chiave crittografica cresce linearmente, ma con un fattore 6 o 7, con il numero di bit, il sistema deve essere progettato in modo da poter “scalare” facilmente. Per i sistemi già esistenti se si utilizzano chiavi di lunghezza inferiore a 1024 bit, il consiglio è di pensare seriamente ad aggiornare il sistema per utilizzare

chiavi a 1024 bit entro i prossimi 10 mesi ed in questo caso prevedere già la possibilità di salire a 1280 o più bit nei prossimi 2 o 3 anni. La decisione dipende ovviamente dal costo globale dell'aggiornamento, sia per quanto riguarda l'hardware ma anche, ad esempio, le possibili conseguenze psicologico – pubblicitarie dell'uso di chiavi considerate insicure in una applicazione di commercio elettronico. Infine se la vostra richiesta di sicurezza è “assoluta”, come ad esempio per applicazioni militari, il suggerimento è di usare chiavi a 2048 o 4096 bit.

Andrea Pasquinucci
Consulente di Sicurezza Informatica

pasquinucci@ucci.it

Anno Personale Aziendale Governativo

1995 768 1280 1536

2000 1024 1280 1536

2005 1280 1536 2048

2010 1280 1536 2048

2015 1536 2048 2048

Lunghezza minima (in bits) delle chiavi RSA per uso Personale, Aziendale e Governativo consigliata da Bruce Schneier in *Applied Cryptography*,

1995

e) I computer quantistici

Riportiamo parzialmente da Wikipedia:

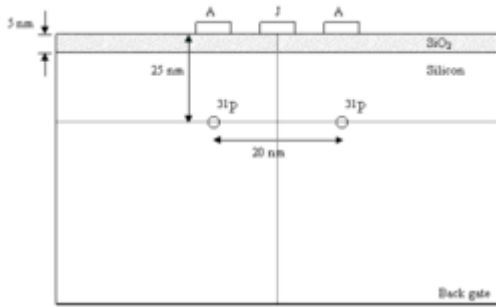
Computer quantistico

Un **computer quantistico** (o quantico) è un dispositivo per il trattamento ed elaborazione delle informazioni che per eseguire le classiche operazioni sui dati utilizza i fenomeni tipici della [meccanica quantistica](#), come la [sovrapposizione degli effetti](#) e l'[entanglement](#).

In un [computer](#) classico, la quantità di dati viene misurata in [bit](#), mentre in un computer quantico l'unità di misura è il [qubit](#). Il principio che sta alla base del computer quantico, è che le proprietà quantistiche delle particelle possono essere utilizzate per rappresentare strutture di dati, e che il complesso meccanismo della meccanica quantistica può essere sfruttato per eseguire operazioni su tali dati.

La prima idea di computer quantico la espose [Richard Feynman](#) nel [1982](#) pensandolo sulla base della sovrapposizione di stati delle particelle elementari.

Anche Eric Drexler indipendentemente rifletté sulla costruzione di computer molecolari (Nel suo libro *Engines of creation: Motori della creazione*).



Schema del computer di Kane

Nel [1985 David Deutsch](#) ne dimostrò la validità.

Nel [1994 Peter Shor](#) dimostrò che così sarebbe stato possibile **fattorizzare qualsiasi numero a grandi velocità...**

Si pensa siano 10 000 volte più veloci di qualsiasi computer normale, quindi non esattamente “quasi istantanei”.

Possibile rimedio: usare numeri primi di qualche decina di milioni di cifre, per mantenere all’incirca lo stesso tempo di calcolo su numeri di mille cifre oggi usati nella crittografia RSA.

Altro brano, da Punto informatico, sito punto-informatico.it/.../cos-crittografia-quantistica.aspx “Cos’è la crittografia informatica” di Andrea Pasquinucci:

“Cos'è la Crittografia Quantistica?”

di Andrea Pasquinucci - Breve semplice guida alla comprensione di un ramo della ricerca che apre prospettive notevolissime e che vede l'Italia in

prima linea. Raccontato da uno dei protagonisti

inShare Roma - Negli ultimi mesi si è sentito parlare molto di Crittografia Quantistica: annunci, press-realese, progetti, finanziamenti e prototipi commerciali. Verso la fine del 2003 sono comparsi sul mercato i primi due prototipi commerciali da parte di "MagiQ Technologies" (New York) e "id Quantique" (Ginevra). Inoltre altre aziende, quali NEC, Toshiba e Hewlett-Packard, stanno sviluppando propri sistemi di Crittografia Quantistica che presto appariranno sul mercato.

La Crittografia Quantistica ha già catturato l'interesse di governi, di militari ed agenzie di sicurezza, di banche ed istituzioni finanziarie.

Ad esempio, Visa International, l'azienda internazionale di carte di credito, sta sperimentando questa tecnologia ed altre banche e istituzioni finanziarie hanno annunciato il loro interesse. L'Unione Europea ha finanziato il progetto SECOQC, iniziato il 1 Aprile 2004 e da alcuni indicato come [il progetto "anti-echelon" europeo](#) per lo sviluppo sia della ricerca che della implementazione tecnologica e commerciale della Crittografia Quantistica (la Press Release è disponibile sul sito quantenkryptographie.at e la descrizione del progetto sul sito www.arcs.ac.at/quanteninfo).

Il progetto ha un budget di 11,4 Milioni di euro in 4 anni, vi partecipano

41 partner in 12 paesi europei, e per l'Italia vi sono l'Università di Pavia, il CNR, la Scuola Normale Superiore di Pisa ed il Politecnico di Milano.

Ma che cosa è la Crittografia Quantistica?

Per rispondere a questa domanda conviene fare un passo indietro. Già negli anni '70 i fisici teorici si chiedevano se fosse possibile utilizzare le teorie che descrivono le particelle elementari, atomiche e sub-atomiche, cioè la Meccanica Quantistica e la Teoria dei Campi, per realizzare direttamente qualche cosa di veramente nuovo. Infatti le leggi che regolano il mondo atomico e sub-atomico sono alquanto differenti da quelle a cui siamo abituati nella vita di tutti i giorni.

Questo le rende difficili da comprendere ma sono al contempo potenzialmente foriere di applicazioni impensabili altrimenti.

Ad esempio, quando si osserva una particella sconosciuta, si modificano sempre alcune delle sue proprietà: non è possibile una osservazione senza una interazione e la modifica dello stato della particella sconosciuta.

Ovviamente, nell'esperienza quotidiana le cose sono ben diverse: possiamo osservare quanto vogliamo un oggetto sconosciuto senza modificarlo affatto. Ed ancora, nel mondo delle particelle elementari non esiste la possibilità di una fotocopiatrice perfetta, non si possono fare copie esatte

di particelle se non in particolari ed eccezionali condizioni.

Uno dei primi risultati teorici è stata l'invenzione dei **Computer**

Quantistici. Questi sono elaboratori che funzionano seguendo la logica delle leggi della Meccanica Quantistica e quindi sono (potenzialmente) in grado di fare i conti in modo molto diverso da quello noto a tutti noi. In particolare gli elaboratori quantistici saranno in grado di risolvere alcuni difficili problemi matematici istantaneamente. Tra questi problemi vi sono quelli su cui si basano molti degli algoritmi crittografici moderni, quali ad esempio il famoso RSA.

In altre parole, se fosse possibile costruire oggi un elaboratore quantistico, questo sarebbe in grado quasi istantaneamente di ottenere da una chiave pubblica di qualunque lunghezza, la corrispondente chiave privata utilizzata dagli algoritmi Asimmetrici quali RSA. Questi algoritmi sono utilizzati oggi per l'identificazione delle parti e la creazione e scambio delle chiavi per cifrare le connessioni. **Poterli "rompere" vorrebbe dire rendere del tutto insicuri smart-card, firme e certificati digitali, navigazione in internet, email cifrate ecc.ecc. Al momento comunque non siamo ancora in grado di costruire un elaboratore quantistico, e le stime più ottimistiche indicano che ci vorranno ancora 20 anni."**

(Vedi paragrafo precedente sui computer quantistici)

f) **Nostra “Congettura numeri RSA”**

(In sezione “Articoli sulla Fattorizzazione” :

“Osservando bene alcuni numeri RSA già fattorizzati e i loro fattori (anche per piccoli numeri RSA, per esempio di 5 cifre, RSA-5, per esempio $29083 = 127 \cdot 229$) abbiamo ideato la seguente “**congettura sui numeri RSA**” :

“Per tutti i numeri RSA, p è compreso tra $n/2$ e n , con $n = \sqrt{N}$ ”

(e quindi è inutile cercarlo tra 3 ed $n/2$)

Se fosse vera, come cercheremo di dimostrare, si risparmierebbe almeno metà dei tempi di calcolo.

Per alcuni numeri RSA, p si potrebbe trovare vicino alla media aritmetica $(n/2 + n)/2$; un piccolo esempio (RSA - 5, cioè con cinque cifre, è : $29083 = 127 \cdot 229$, con $n = 170,53 \approx 170$

$$p' \approx (170/2 + 170)/2 \approx (85 + 170)/2 = 255/2 = 127,5 \approx 127 = p$$

Un altro esempio potrebbe essere il numero RSA (617), che inizia con le cifre 25 195... molto simili alle cifre di 29083 dell'esempio precedente , quindi potrebbe trovarsi anch'esso nel mirino della media aritmetica:

$n \approx \sqrt{25195...} \approx 158....$; $p' \approx (158.../2 + 158...)/2 = 118...$ le prime tre cifre, seguite da altre 305 cifre (questa è la nostra previsione , per quando RSA(617) sarà fattorizzato da altri ricercatori)

Ma è ancora molto difficile individuare questi numeri RSA particolari, piccoli o grandi che siano, e quindi questa congettura non sarebbe alla fine molto pericolosa per il noto sistema crittografico basato sui numeri RSA”.

Per il momento no, ma potrebbe essere dimostrata e perfezionata in futuro e quindi essere un po' più pericolosa di quanto si potesse immaginare oggi.

Rimedio: ampliare il rapporto tra i due numeri primi p e q di $N = p*q$, portandolo da $q/p \approx 4$ come valore massimo odierno (ma mediamente è di circa 2), ad un valore superiore di circa $q/p \approx 6$ oppure 7 nella scelta dei numeri RSA da usare in crittografia.

g) Ing. Rosario Turco, dal suo blog:

<http://mathbuildingblock.blogspot.com/>

“lunedì 10 novembre 2008

[L'RSA è attaccabile?](#)

L'RSA è attaccabile?

Una delle domande che spesso ci si pone è se l'RSA è attaccabile e se occorre necessariamente conoscere la funzione totiente di Eulero per poter risalire a monte fino ai due numeri primi che sono fattori del prodotto.

Nell'articolo sulle "Spalle dei giganti" menzioniamo almeno due metodi con cui poter risolvere il problema teoricamente (vedi articoli corrispondenti).

Uno è quello della soluzione dell'equazione $x^2 = a^2 \pmod n$ dove n è il semiprimo da fattorizzare in due numeri primi.

Il secondo è quello di utilizzare un'equazione completa di secondo grado ed il vincolo che la somma dei due primi da trovare sia un numero pari (Goldbach). Ebbene con questo metodo è possibile risolvere facilmente il problema. Anzi il tutto è generalizzabile ad un sistema di $m-1$ disequazioni anche nel caso di un composto costituito da m fattori di molteplicità qualsiasi (grazie alla funzione bigomega).

Ovviamente maggiore è il numero di cifre del composto da fattorizzare maggiore è il tempo necessario a tale operazione.

Per convincersene è possibile crearsi un generatore casuale di "prodotto di due numeri primi" (un tale prodotto è detto semiprimo o numero RSA) e scrivere un "breve" algoritmo per il crack del RSA.

Un modo per creare un generatore per un test di crack RSA (ma non è strettamente necessario, solo che vogliamo evidenziare la duplice utilità

delle congetture di Goldbach) è il seguente:

- si genera casualmente un numero pari, poi si trovano tutte le
-
- soluzioni di Goldbach ad esso legato
-
- si sceglie a caso una di tali coppie escludendo il caso $N=12n$ perchè
-
- produce coppie gemelle (il crack di un semiprimo RSA di poche cifre
-
- è facile nel caso di coppie di numeri primi gemelli e di quadrati
-
- perfetti)
-
- si calcola infine il prodotto $N=p*q$.

A questo punto sul prodotto generato si può testare il metodo del crackRSA.

Vi presentiamo con PARI/GP un algoritmo didattico, in tal senso, non ottimizzato.

L'algoritmo didattico ci rivela però cose importantissime:

1. Le tecniche di scomposizione che individuano numeri primi gemelli
2. attraverso quadrati perfetti, (i numeri gemelli rientrano come sottoproblema dei quadrati perfetti in generale), fanno capo comunque alla tecnica dell'eq. di secondo grado con condizione imposta con Goldbach
3. non è necessario dover impostare tutte le $m-1$ disequazioni poichè iterativamente si può utilizzare il risultato precedente con un'altra equazione di secondo grado (quindi scomponendo i termini a due alla volta e conoscendo il numero di fattori totali, anche ripetuti, con la funzione bigomega);
4. con la fattorizzazione comunque c'entra la RH, attraverso Goldbach; quello che si evita è la funzione totiente, non necessaria alla soluzione del problema, ma diventa necessaria la bigomega al suo

- posto. In ogni caso la ricerca di quadrati e/o radici quadrate è coinvolta;
5. crackRSA, come vi renderete conto, non bisogno di far riferimento ai numeri primi (non ci sono condizioni isprime(), grazie a Goldbach) ma solo ai quadrati perfetti;
 6. una funzionalità di scomposizione, basata su crackRSA , ha bisogno di condizioni isprime() per la bufferizzazione in uno stack; ma è solo un problema tecnico, per scegliere solo quei numeri primi che vanno nel vettore di output, mentre quello nello stack hanno bisogno di ulteriore scomposizione.

Nel seguito esaminiamo l'algoritmo.

La prima parte, non è strettamente necessaria e presenta un generatore casuale di primi con Goldbach. L'autore l'ha utilizzata per generarsi dei semiprimi casuali su cui usare crackRSA.

La seconda parte un algoritmo per attaccare un semiprimo RSA. Usando quest'ultimo iterativamente si può scomporre un numero con m fattori qualsiasi, con $m = \text{bigomega}(\text{numero})$.

Gli algoritmi crackRSA e scompFactor sono didattici. Difatti sono meno veloci, a parità di cifre, rispetto alla funzionalità factor fornita da PARI stesso.

Ma esaminiamolo e commentiamolo insieme.

Il sorgente completo (una vera Libreria per la Teoria dei Numeri e con altre funzionalità come Collatz ed altro) ed eventuali ottimizzazioni e nuove versioni è al link dell'autore che gentilmente ce lo ha fornito:

Prima parte - Generatore di $N=p*q$ per il test

/*****

* Goldbach

- ° Ritorna le coppie di numeri primi $(x, n-x)$,
- ° non ripetute, la cui somma è uguale a un numero pari n .

* Rosario Turco

*

*****/

{ {printGold(n) = local ();

if(n<=4, error("printgold(n) --> with n>4 and even"));

if(Mod(n,2) != 0, error("printgold(n) --> You must insert an even n>4"));

for(p=3, n, if(isprime(p) & isprime(n-p) & p <= n-p, print1("(" , p, ", " , n-p, ")", " ");)); }
{ gold(n) = local s; if(n<=4, error("gold(n) --> n>4 and even"));

if(Mod(n,2) != 0, error("gold(n) --> You must insert an even n>4"));
s=0;

for(p=3, n, if(isprime(p) & isprime(n-p) & p <= n-p, s++)); return(s) } /* * * It returns
all solutions of Goldbach in a matrix 2xn * * R. Turco */ {memGold(n) =
local (mat,col, j); if(n<=4, error("printgold(n) --> with n>4 and even"));

if(Mod(n,2) != 0, error("printgold(n) --> You must insert an even n>4"));

col = gold(n);

mat = matrix(2, col);

j=1;

```
for(p=3, n, if(
isprime(p) & isprime(n-p) & p <= n-p, mat[1,j] = p; mat[2,j] = n - p;
j++; ) ); return(mat); } /* * It chooses, in way random, a solution between
all * solutions of Goldbach and inserts it in a vector * * R. Turco */
{randGold(n) = local (mat,col, j, vec); if( n<=4, error("printgold(n) -->
with n>4 and even"));
```

```
if( Mod(n,2) != 0, error("printgold(n) --> You must insert an even n>4"));
```

```
col = gold(n);
```

```
mat = matrix(2, col);
```

```
mat = memGold(n);
```

```
j = random(col);
```

```
if( j==0, j++);
```

```
if( j>col, j--);
```

```
vec = vector(2);
```

```
vec[1]=mat[1,j];
```

```
vec[2]=mat[2,j];
```

```
return(vec);
```

```
}
```

```
/*
```

Genera due numeri primi ma scarta $N=12n$ per evitare
due numeri gemelli, altrimenti l'RSA è facile da
attaccare

R. Turco

```
*/
```

```

{getPrimes() = local (j, vec, test);

vec=vector(2);
test=1;

j=random(10^6); /* numbers of digit that you want */

while( test>0 | j<5 2==" 12==">4 & (j%2==0) & !(j%12==0), test=0;
break; );
);

vec=randGold(j);

return(vec);
}

```

```
/*
```

```
getRSA
```

Genera un semiprimo RSA dal prodotto di due numeri primi

che vengono scelti randomicamente da due soluzioni di

Goldbach.

Se si imposta getRSA(1) si ottiene la stampa dei due

numeri primi e del prodotto

R. Turco

```
*/
```

```
{getRSA(f) = local (vec, valore);
```

```

vec=vector(2);
vec=getPrimes();
valore=vec[1]*vec[2];
if( f==1, print("getRSA: p = ", vec[1], " q = ", vec[2], " p*q = ", valore));
return(valore);
}

```

Seconda parte - crack del RSA

Stupisce il numero esiguo di linee di codice che occorrono

```

/*
crackRSA(numero)
Dal semiprimo RSA n
deve trovare i due numeri primi tali che  $n=p*q$ 

```

Un modo casuale di usare la funzione è:

```
crackRSA(getRSA(1))
```

un altro è:

```
crackRSA(numero)
```

R. Turco

```
*/
```

```
{crackRSA(p) = local (s, vec, qp);
```

```
vec=vector(2);
```

```
qp = 4*p;
```

```
print("\ncrackRSA\n");
```

```
s=floor(sqrt(qp));
```

```
while( !(s^2>=qp) | (issquare(s^2 - qp) == 0) | (frac(s^2 - qp)!= 0.0) , s++);
```

```
vec[1]=floor((s+sqrt(s^2-qp))/2);  
vec[2]=floor((s-sqrt(s^2-qp))/2);  
  
return(vec);  
  
}
```

Come usare il tutto?

Per l'uso combinato Generatore del test e crackRSA
digitate:

```
crackRSA(getRSA(1))
```

L'1 serve a farsi scrivere un trace dei numeri primi attesi che crackRSA trovi.

Uso singolo di crackRSA, esempio con N=55
digitate:

```
crackRSA(55)
```

Aspettate!

Per decomporre in fattori un numero con m fattori mi servono davvero m-1 disequazioni? No, lo abbiamo visto prima. Devo però sapere quanti fattori, anche con molteplicità, ci sono.

Si può quindi fare la decomposizione in fattori di un numero a m fattori iterando su crackRSA.

Non solo, ma scompFactor trova anche i numeri primi! Restituisce difatti

un vettore nullo: [0].

Provate ad esempio: 4082777003

Eccovi il pezzo di funzionalità:

```
/*
 * If it returns [0], then it is a prime number.
 * scompFactor uses bigomega
 */
{scompFactor(p) = local ( col, f1, f2, f3, s, i, j, vecIn, vecOut, vecStack);

vecIn=vector(2);

col = bigomega(p);

vecOut=vector(col);
vecStack=vector(2*col);

f1=0;
f2=0;
f3=0;

s=1;
j=1;
i=1;
f3=0;

print("\n scompFactor\n");

while( s1 & s>1 & f1==1 & f2==1 & (vecStack[i-1]!= 0),
p=vecStack[i-1]; f3--; vecStack[i-1]=0; i--;
);

if( i==1 & s>1 & f1==1 & f2==1 & (vecStack[i]!= 0), p=vecStack[i]; f3--;
vecStack[i]=0;
```

```
);
```

```
if(p==1, s=col);
```

```
f1=0;
```

```
f2=0;
```

```
s++;
```

```
);
```

```
return(vecOut);
```

```
}
```

L'autore ci ha segnalato che, in realtà, esistono funzionalità più veloci come factor di PARI e che l'algoritmo era didattico e serviva ad arrivare a determinate conclusioni.

Conclusioni

In tutto questo apparentemente l'ipotesi di Riemann non è servita, anche se Goldbach è un sottoproblema della RH!!!

L'RSA regge solo grazie ad un numero di cifre del semiprimo elevate (almeno 100!). Sono

consigliabili comunque altri algoritmi come AKS, o 3DES o altri ancora, soprattutto non basati su un problema come la fattorizzazione che è noto ad una vasta platea.

L'RSA comunque regge solo grazie al tempo necessario a trovare la

soluzione dei due fattori e, quindi, l'unico bastione di difesa è legato proprio al numero di cifre che costituiscono il semiprimo da decrittare.

La fattorizzazione coinvolge anche il famoso problema se $P=NP?$ ”

Aggiungeremo in seguito questo articolo con ulteriori e importanti brani su eventuali altri pericoli per la crittografia RSA, ed eventuali altri rimedi nel frattempo emersi nella ricerca matematica dedicata alla crittografia RSA.

Gruppo Eratostene

Riferimenti

Tutti gli articoli delle sezioni “Articoli sui Problemi del Millennio” e “Articoli sulla Fattorizzazione” del nostro sito:

www.gruppoeratostene.com

Caltanissetta 1.3.2011