CryptoMythbusters: Miti e possibilità pratiche della crittografia moderna

Alessandro Barenghi

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

barenghi - at - elet.polimi.it

31 marzo 2014

Overview

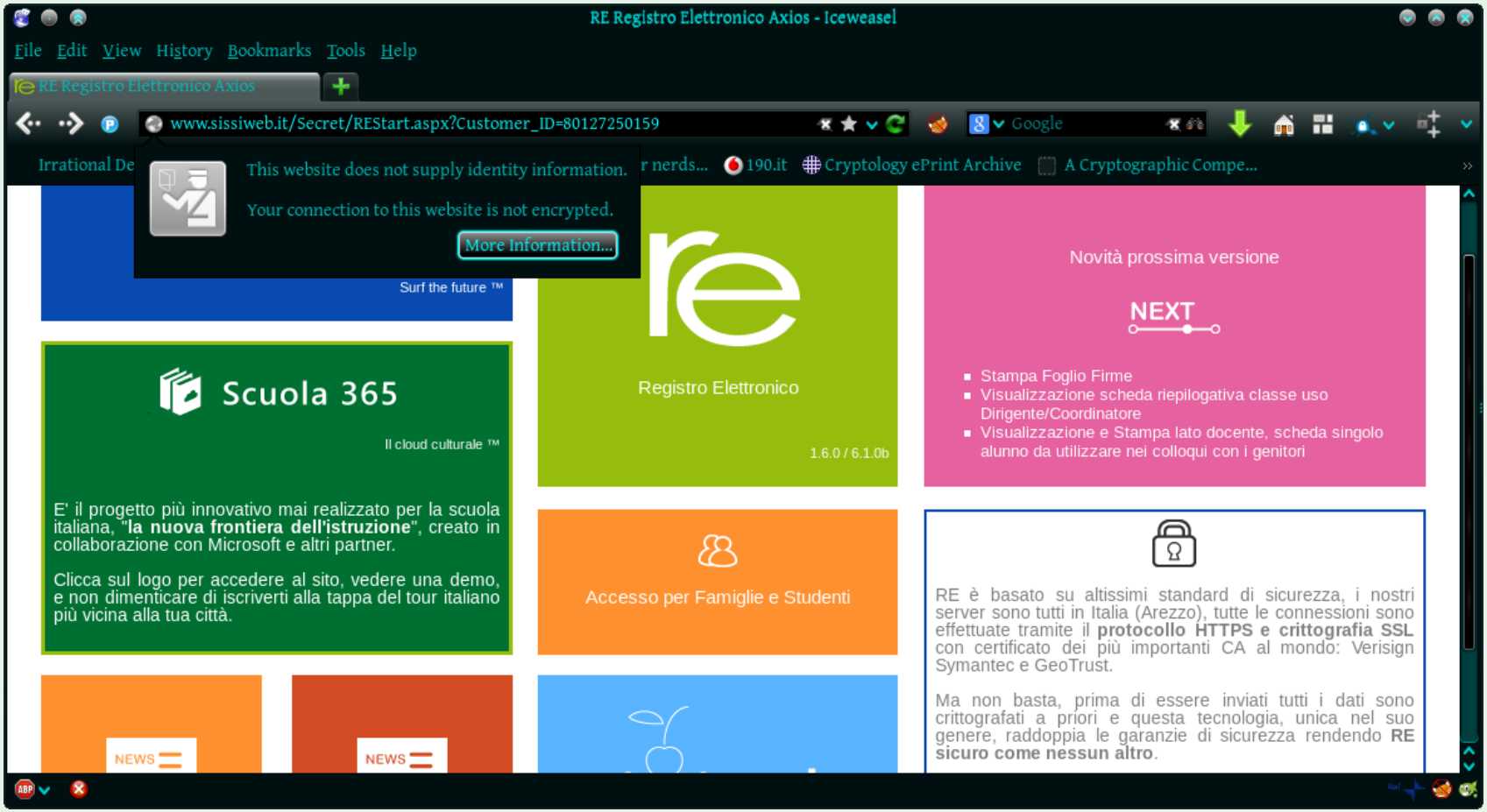
Di cosa parleremo oggi?

- La quantità di informazione stoccata/in movimento nel mondo moderno è impressionante
- La confidenzialità e riservatezza di questa marea di bit è un serio problema
- Utilizzare tecniche crittografiche è una soluzione efficiente, a patto che si eviti l' effetto "cargo cult"
- Puntiamo a far saltare un po' dei "sentito dire" e del FUD[a] riguardanti la crittografia

[a] https://en.wikipedia.org/wiki/Fear,_uncertainty_and_doubt

Overview

FUD in che senso?



Crittografia senza chiave

E' possibile avere confidenzialità senza chiavi?

- **Mito**: E' possibile avere un sistema che cifra dati garantendo confidenzialità senza una chiave.
- Esempio tipico: password manager che salva le password in modo sicuro senza chiedervi una master password
- Il risultato è un blocco di bit, apparentemente informe, salvato su disco fisso
- Claim: il grumo di dati non è intelligibile, quindi è sicuro

Crittografia senza chiave

E' possibile avere confidenzialità senza chiavi?

- **Mito**: E' possibile avere un sistema che cifra dati garantendo confidenzialità senza una chiave.
- Esempio tipico: password manager che salva le password in modo sicuro senza chiedervi una master password
- Il risultato è un blocco di bit, apparentemente informe, salvato su disco fisso
- Claim: il grumo di dati non è intelligibile, quindi è sicuro

Crittografia senza chiave

Back to 1883

- Per decodificare il grumo di dati basta avere una copia del programma (non utile, se voglio che molti usino il programma)
- Assioma fondamentale della crittografia: il metodo con cui si cifra è assunto pubblico (Auguste Kerchoff, 1883)
- E' segreto un solo parametro, la chiave
- La difficoltà di rompere un sistema di cifratura, è solamente quella di trovare/indovinare la chiave

Nano-tutorial di crittografia

Cifrario Simmetrico

- Esiste una sola chiave k_{sym}
- La chiave è nota solo a chi può accedere ai dati
- Si usa sia per cifrare i dati, che per decifrarli
- Molto efficiente (100MB/ - 10GB/s)

Cifrario Asimmetrico

- Esistono due chiavi k_{pub}, k_{pri}
- Non è possibile ricavare in tempi pratici k_{pub} da k_{pri}
- E' possibile decifrare il cifrato di k_{pub} solo con k_{pri}
- Circa 100 volte più lento di un cifrario simmetrico

Livello di sicurezza

Quantificazioni pratiche

- Quanto robusta è la cifratura in pratica? Dipende da quanto è difficile indovinare la chiave
  - Cifrario simmetrico: n bit di chiave $\rightarrow 2^n$ tentativi
  - Cifrario asimmetrico: sapendo k_{pub} è "più facile" indovinare k_{pri}: con n bit di chiave $\rightarrow 2^{\frac{n}{m}}$ tentativi (m dipende dal cifrario)
- Facile fare crescere lo sforzo di calcolo oltre i limiti del fattibile
- Limiti fisici della computazione intervengono a un certo punto
- Lunghezze di chiavi scelte tenendo conto dell' evoluzione delle capacità di calcolo[a]

[a] `http://www.keylenght.com`

Livello di sicurezza

Quanta energia mi serve per indovinare la chiave?

Sicuro?	Lunghezza della chiave [b] Simm. (AES)	Asimm. (RSA)	Asimm. (EC*)	Acqua scaldata 20C $\rightarrow$ 100C
NO	35	284	64	Cucchiaino da caffè
no	64	803	117	Piscina (50m)
sì	80	1233	148	Pioggia 1yr Olanda
Sì	114	2541	215	Tutta

- Convertendo l' intera massa dell' universo osservabile in energia arriviamo a indovinare delle chiavi da 256b Simm., 15489b Asimm. (RSA), 494b Asimm. EC*

Confidenzialità post Datagate

The NSA took my baby away

- Dopo la fuga di informazioni consentita da Ed Snowden, abbiamo le prove di un programma di sorveglianza su scala mondiale
- **Mito**: "NSA riesce a decifrare il traffico SSL!" e "NSA dedica attivamente sforzi, con successo, a rompere crittografia forte"
- Ma quindi cifrare i dati è inutile?

Confidenzialità post Datagate

The NSA took my baby away

- Dopo la fuga di informazioni consentita da Ed Snowden, abbiamo le prove di un programma di sorveglianza su scala mondiale
- **Mito**: "NSA riesce a decifrare il traffico SSL!" e "NSA dedica attivamente sforzi, con successo, a rompere crittografia forte"
- Ma quindi cifrare i dati è inutile?

Confidenzialità post Datagate

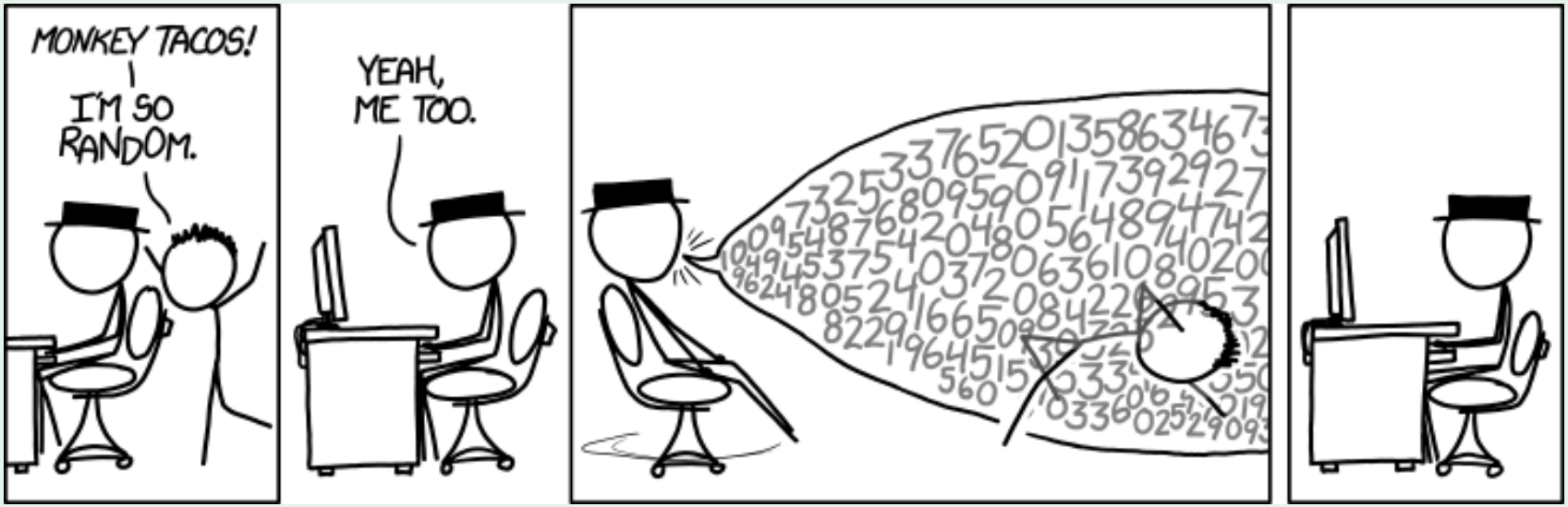
Ed Snowden, 10/3/2014

The bottom line, I have repeated this again and again, is that encryption does work. We need to think of encryption not as this sort of arcane black art. [...] it is a defense against the dark arts for the digital realm.

- Cifrario robusto $\rightarrow$ cifrario di cui sappiamo che trovare la chiave implica una quantità di risorse troppo alta
- Importante la scelta di chiavi solide $\rightarrow$ "difficile" da indovinare $\rightarrow$ estratta a caso tra le possibili chiavi per quel cifrario

Confidenzialità post Datagate

Casuale in che senso?



Courtesy of XKCD: `http://xkcd.com/1210/`

Confidenzialità post Datagate

Ed Snowden, 10/3/2014

[...] "we know that encryption algorithms we are using today work", typically it is the Random Number Generators that are attacked [...]

RNG FAILs

- Se l' RNG che decide la chiave è predicibile, non devo indovinarla!
- `Dual_ERBG`, standardizzato contiene una backdoor
  - Falla nota 4 anni prima del Datagate
- Debian ha usato un RNG con solo 16 bit di entropia per anni!

Confidenzialità post Datagate

RNG sicuri

- Un RNG puramente algoritmico si ripeterà a un certo punto
- `/dev/random` sotto Linux sfrutta la lettura di parametri fisici della macchina
- Molto difficile predirlo, ma può far attendere prima di produrre dati

This is my RNG. There are many like it, but this one is mine.

- Interfaccia USB, 3MB/s `http://goo.gl/xrgaUp`
- Interfaccia Seriale 9600B/s `http://goo.gl/6LBXcz`

Quantum Computers

I QC rompono tutta la crittografia moderna?

- Alcuni computer quantistici sono ormai commerciali: `http://www.dwavesys.com/`
- Sono in grado di effettuare alcuni calcoli per tutti i valori di un parametro a n bit in n operazioni
- Quindi, posso rompere qualunque cifrario: per trovare la chiave ci metto n esecuzioni anzichè 2^n
- NSA ha computer quantistici, quindi è inutile cifrare!

Quantum Computers

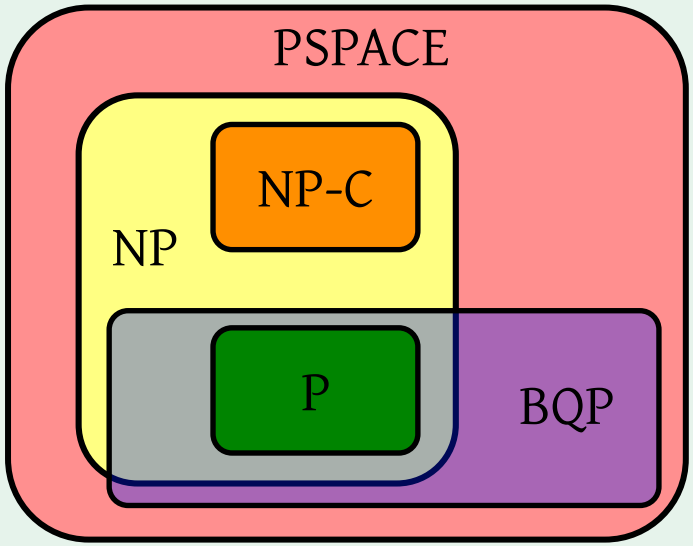
I QC rompono tutta la crittografia moderna?

- Alcuni computer quantistici sono ormai commerciali: `http://www.dwavesys.com/`
- Sono in grado di effettuare alcuni calcoli per tutti i valori di un parametro a n bit in n operazioni
- Quindi, posso rompere qualunque cifrario: per trovare la chiave ci metto n esecuzioni anzichè 2^n
- NSA ha computer quantistici, quindi è inutile cifrare!

Nano-tutorial di teoria della complessità

Classi di complessità: dato un input lungo n....

- P: Risolvo in n^k op.
- NP: Risolvo in n^k op., se indovino tutti gli `if-else`
- NP-C: Come risolvo lui, risolvo tutti gli NP
- PSPACE: Risolvo con al più n^k memoria letta-scritta
- BQP : Risolvo, con un QC, in n^k, "non sbagliando quasi mai"



Quantum Computers

In pratica

- Cifrari simmetrici: il problema di trovare la chiave è in NP-C
  - Miglior risultato su QC: passo da 2^n tentativi a $2^{\frac{n}{2}}$
  - Raddoppio la lunghezza della chiave, ed è fatta :)
- Cifrari asimmetrici attuali (f.i., RSA,DSA,ECDSA):
  - Hanno un algoritmo in BQP: n bit di chiave $\rightarrow n^k$ tentativi!
  - Troppo costoso aumentare la chiave a sufficienza
  - Soluzioni? Cambiare cifrari

Quantum Computers

Crittografia Post-Quantum

- La crittografia asimmetrica basata sulla difficoltà di fattorizzare interi/del logaritmo discreto cede con i QC
- Servono nuovi problemi difficili da usare come cifrario....
  - 1978: R. McEliece: cifrario basato sulla difficoltà di decodificare un dato con troppi errori
  - 1997: S. Goldwasser: cifrario basato su spazi vettoriali a coefficienti interi (Lattices)
- Se sono già resistenti ad attacchi con QC, perchè non li stiamo già usando?
  - Bassa efficienza: sono più lenti degli attuali cifrari asimmetrici
  - Chiavi molto grandi: circa $10\times$ più grosse delle chiavi asimmetriche attuali, qualche MB alla peggio

Offuscamento perfetto

Black-box security

- Recentemente è stato dimostrato essere possibile cifrare un programma in modo che non si comprenda cosa fa, ma lo si possa eseguire ugualmente
- Il tutto è stato riportato perfino da riviste tra cui Wired [a]
- Questo sistema è la risorsa definitiva del software proprietario!
- Possiamo scrivere un malware invulnerabile alle analisi perchè non verrà mai trovato!

[a] `http://www.wired.com/2014/02/cryptography-breakthrough/`

Offuscamento perfetto

Black-box security

- Recentemente è stato dimostrato essere possibile cifrare un programma in modo che non si comprenda cosa fa, ma lo si possa eseguire ugualmente
- Il tutto è stato riportato perfino da riviste tra cui Wired [a]
- Questo sistema è la risorsa definitiva del software proprietario!
- Possiamo scrivere un malware invulnerabile alle analisi perchè non verrà mai trovato!

[a] http://www.wired.com/2014/02/cryptography-breakthrough/

Offuscamento perfetto

Wait, Offuscamento perfetto, what?

- Il problema principale del mito precedente è cosa si intende per offuscamento perfetto
- Senso comune: ottenere un programma artificialmente contorto per evitare che si comprenda il senso di cosa fa guardando la sequenza di istruzioni
- Dato di fatto: dato sufficiente tempo e sufficienti sforzi, è sempre possibile capire cosa fa un programma come quello che abbiamo detto qui sopra
- Non c'è una difficoltà quantificabile, quindi qualcuno prima o poi ce la farà

Offuscamento perfetto

Un primo tentativo...

- Il primo passo per l' offuscamento perfetto è stato chiedersi: cosa vogliamo ottenere?
- Primo tentativo : Virtual Black-Box Obfuscation
  - Alice riceve un programma offuscato $Obf(P)$ da far girare sulla sua macchina
  - Bob può usare lo stesso programma solo mandando per posta cartacea gli input e ricevendo gli output
  - Obiettivo: Quando entrambi hanno finito di fare tutti i conti che vogliono, Alice non sa nulla più di Bob
- Entrambi possono invocare il programma quante volte vogliono e salvare tutte le coppie input-output

Not really...

... Con qualche problema

- 2001: Barak et al.: impossibile ottenere l' offuscamento Virtual Black-Box per programmi generici
- Come è possibile? Sketch della dimostrazione:
  - Il programma P da offuscare, se riceve `KLAATUBARADANIKTO` stampa `thisisreallysecret`, altrimenti non stampa nulla.
  - Terminato l' accesso al programma, Bob non è più in grado di dire se un input di P fa sì che P stampi `thisisreallysecret`.
  - Alice può ancora far girare $Obf(P)$ e, anche se non sa come funziona, dire se un input fa sì che stampi `thisisreallysecret`
- Il fatto stesso che Alice abbia una copia del programma fa sì che lei sappia più di Bob

Virtual black box obfuscation in pratica

Sì, ma —

- La dimostrazione precedente fa vedere che non è possibile offuscare un programma piuttosto banale...
- Purtroppo, il programma in questione non è un' eccezione:
  - L'offuscamento non è possibile per la stragrande maggioranza dei programmi utili
- Una delle due eccezioni è sono i programmi che calcolano un output di solo bit dagli input
- Esempio tipico: programma che computa 0 o 1 a seconda che la password fornita sia corretta

Indistinguishability Obfuscation

Una nozione più debole

- L' articolo di Wired fa riferimento quindi a un offuscamento diverso: l' Indistinguishability Obfuscation
- Informalmente, due programmi P_1 e P_2 grossi all' incirca uguale e con lo stesso comportamento ai morsetti, $Obf(P_1)$ non si distingue da $Obf(P_2)$
  - non si distingue $\rightarrow$ sono cifrati, devo fare 2^n calcoli
- Carino, ma... utilità pratica?
  - Cifrario simmetrico S con chiave k_{sym}, comportamento: trasforma un testo sensato in bytes pseudocasuali
  - Calcolo $Obf(S_{k_1})$, non si distingue da $Obf(S_{k_2})$:)
  - Distribuisco $Obf(S_{k_1})$, così tutti possono cifrare qualcosa
  - Solo io so decifrare, conoscendo k_1: ottengo un cifrario asimmetrico da uno simmetrico!

Conclusioni

Cosa portare a casa

- No free lunch: no cifratura senza chiave
  - Se non c'è una chiave, e posso invertirla, è una codifica
- Algoritmi crittografici solidi lo sono perchè so quanto ci metto a romperli, ed è troppo
- I computer quantistici richiederanno di cambiare cifrari asimmetrici e una lunghezza di chiave doppia per i simmetrici
- Offuscamento crittografico $\neq$ Offuscamento in senso comune
  - Difficoltà ben quantificata
  - Possibile totalmente solo per funzioni molto semplici
  - Indistinguishability obfuscation possibile per un generico programma