Crittografia con Python

Corso introduttivo
Marzo 2015

Con materiale adattato dal libro "Hacking Secret Cypher With Python"
di Al Sweigart (http://inventwithpython.com/hacking/index.html)

Prof. Alessandro Bugatti

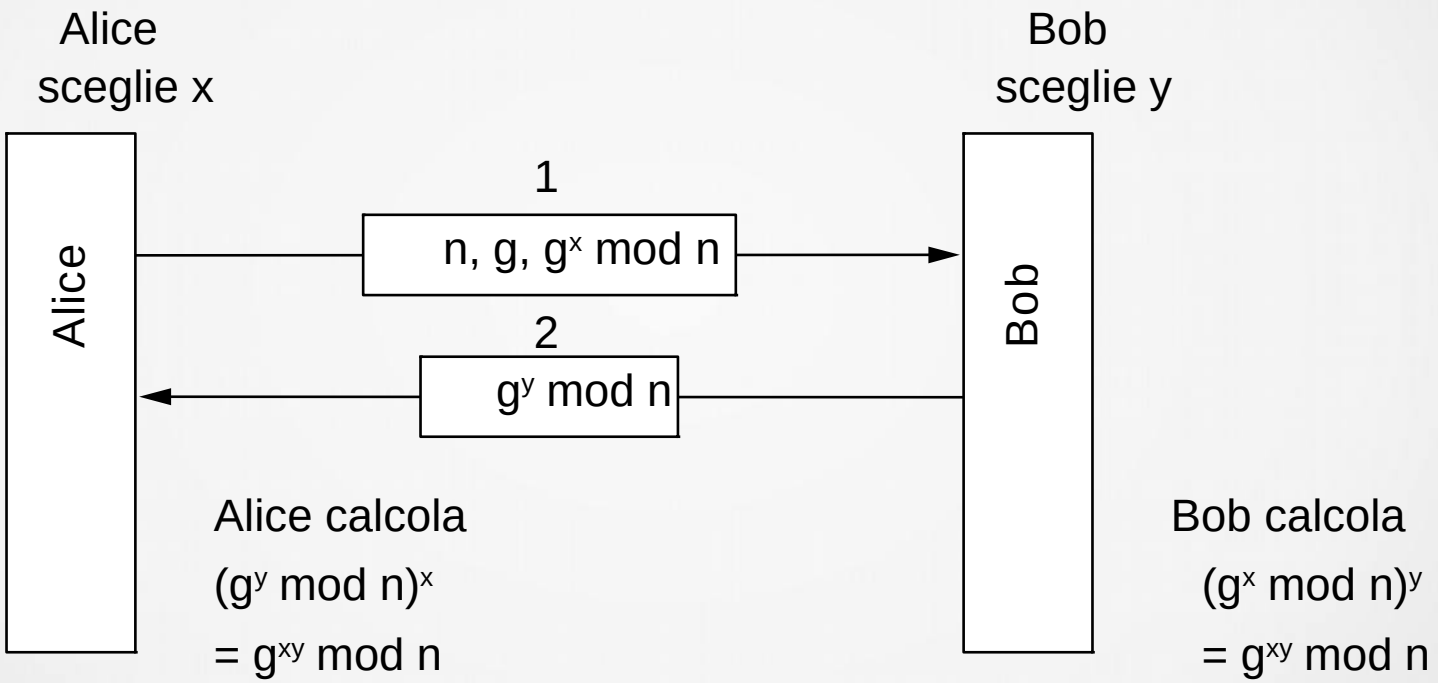
Ci eravamo lasciati così...

- Due volontari:
- Prendete i numeri 753 e 413
- Scegliete un numero a caso x (e ricordatevelo) minore di 413
- Elevate 753 alla x e fate modulo 413
- Dite a voce alta il numero ottenuto, chiamiamolo y
- Prendete y e elevato a x modulo 413

Scambio di chiavi di Diffie-Hellman

- **Problema**: come condividere la chiave (o le chiavi) senza incontrarsi personalmente?
- **Soluzione**: algoritmo di Diffie-Hellman
  - Vengono scelti due numeri primi n e g molto grandi e resi pubblici
  - Alice e Bob scelgono indipendentemente due numeri x e y
  - Attraverso uno scambio di messaggi creano una chiave condivisa

Scambio di chiavi di Diffie-Hellman

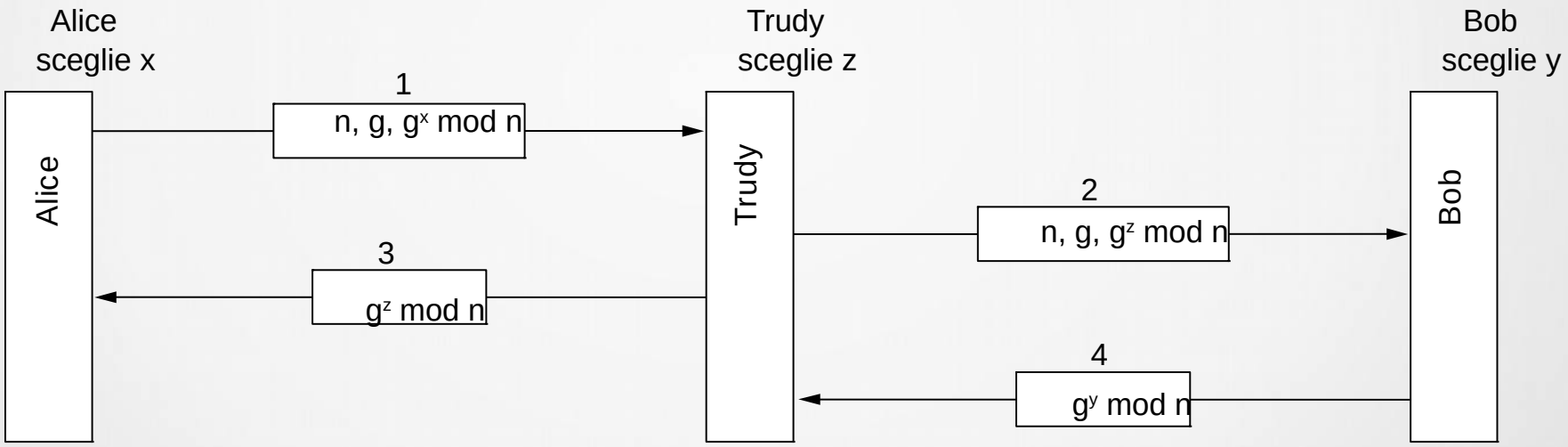


Esempio

- n=47, g=3, x=8, y=10
- Messaggio da Alice a Bob
  - 3^8 mod 47=28 quindi (47,3,28)
- Messaggio da Bob ad Alice
  - 3^{10} mod 47=17 quindi (17)
- Chiave condivisa
  - 17^8 mod 47= 28^{10} mod 47 = 4

Scambio di chiavi di Diffie-Hellman

- **Problema:** come essere sicuri dell'identità di chi manda il primo messaggio?



Attacco "Man in the middle"

Algoritmi a chiave pubblica

- Problema degli algoritmi a chiave segreta: distribuzione e sicurezza delle chiavi
- Diffie e Hellman, dopo aver proposto l'algoritmo visto in precedenza, nel 1976 proposero un metodo di cifratura nuovo in cui chiave di codifica e di decodifica sono differenti
- Non possono essere derivate l'una dall'altra

Nel mondo reale

- Distribuisco un lucchetto uguale in ogni ufficio postale, di cui solo io ho la chiave
- Chi vuole spedirmi un messaggio segreto va con il suo scrigno in un ufficio e prende uno dei miei lucchetti e chiude lo scrigno
- Quando mi arriva lo scrigno prendo la mia chiave e lo apro
- Con i lucchetti non è realistico…

L’algoritmo RSA

- Inventato nel 1978 da Rivest, Shamir, Adleman
- Utilizza l’aritmetica modulare
- Si fonda sul fatto che sia estremamente difficile fattorizzare numeri molto grossi (l’esperienza accumulata dai matematici ha dimostrato che è così)
- RSA (Rivest, Shamir, Adleman)

L’algoritmo RSA

- Funzionamento
  - Scegliere 2 grandi numeri primi p e q ($>10^{100}$)
  - Calcolare $n = p \times q$ e $z = (p-1) \times (q-1)$
  - Scegliere un numero primo rispetto a z e chiamarlo d
  - Trovare e tale che $(e \times d) \bmod z = 1$
- (e,n) chiave pubblica e (d,n) chiave privata

L’algoritmo RSA

- Il messaggio P viene scomposto in blocchi P_i tali che $0 \leq P_i \leq n$
- Per ogni blocco $C_i = P_i^e \pmod{n}$
- Per decodificare $P_i = C_i^d \pmod{n}$
- La difficoltà sta nell’impossibilità di scomporre *n* nei numeri *p* e *q* di cui è il prodotto

L’algoritmo RSA: esempio

p=3, q=11⇒ n=33 z=20 d=7 e=3

Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	$P^3 \pmod{33}$	C^7	$C^7 \pmod{33}$	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	1	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	5	E

Sender's computation

Receiver's computation

L’algoritmo RSA

- Con p e q dell’ordine di 10^{100} è possibile avere blocchi di 664 bit, quindi 83 caratteri da 8 bit, contro gli 8 del DES
- Con 200 cifre 4000000000 di anni a forza bruta
- RSA è molto più lento di algoritmi a chiave privata come AES, quindi nella realtà vengono usati sistemi ibridi, nei quali RSA viene usato per scambiare la chiave di sessione e AES o analoghi per la cifratura successiva

Esempio di numero con 640 bit

- RSA-640 =
  3107418240490043721350750035888567930037346022842727545720161948823206440518081504556346829671723286782437916272838033415471073108501919548529007337724822783525742386454014691736602477652346609
- RSA-640 =
  1634733645809253848443133883865090859841783670033092312181110852389333100104508151212118167511579

  ×

  1900871281664822113126851573935413975471896789968515493666638539088027103802104498957191261465571

RSA per la cifratura

- Bob crea una copia chiave privata – chiave pubblica e distribuisce la sua chiave pubblica sulla rete
- Alice cifra il messaggio con la chiave pubblica di Bob
- Alice spedisce il messaggio cifrato a Bob, che è l'unico che lo può leggere perché solo lui ha la chiave privata che permette la decodifica

RSA per la firma

- Alice crea una copia chiave privata – chiave pubblica e distribuisce la sua chiave pubblica sulla rete
- Alice firma il messaggio cifrandolo con la sua chiave privata (o allegandogli un digest codificato con la sua chiave privata) e lo spedisce a Bob
- Bob usa la chiave pubblica di Alice per decodificare il messaggio e verificare l'dentità di Alice, che è garantita poiché è l'unica a poter creare il messaggio

RSA per cifratura e firma

- Si combinano i due passaggi precedenti
- Alice firma il messaggio con la sua chiave privata, poi con la chiave pubblica di Bob e poi lo spedisce
- Bob prima decodifica il messaggio con la propria chiave privata e successivamente ne verifica il mittente usando la chiave pubblica di Alice

Come si trovano numeri primi grandi

- Magari non è evidente, ma trovare numeri primi grandi non è banale...
- È vero che sono infiniti, ma più diventano grandi più sono sparsi
- Inoltre bisognerebbe scegliere primi sempre diversi, quindi bisogna produrli ogni volta

Algoritmo di Miller-Rabin

- Come altri algoritmi di questo tipo non è deterministico
- Si basa sulla verifica di alcune proprietà che un primo deve soddisfare
- Purtroppo anche alcuni numeri, detti pseudoprimi, soddisfano queste proprietà.

Algoritmo di Miller-Rabin

- Se si vuole verificare se **N** è primo, si sceglie un numero **a** e si verificano alcune proprietà.
- Se queste proprietà risultano soddisfatte allora **N** è probabilmente primo, altrimenti è sicuramente composto
- La probabilità che un numero composto passi il test per **a** è ¼ (nel peggiore dei casi)
- Basta ripetere il procedimento per diversi valori di **a** per ottenere una probabilità grande a piacere di aver trovato un numero primo