Qualcosa che sai: difendersi con un segreto

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

Educandato Setti Carraro Dalla Chiesa — 17 giugno 2016

Autenticazione & autorizzazione

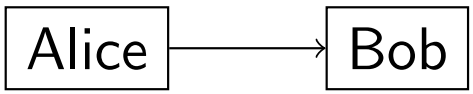
L'accesso ai servizi critici è controllato

- **Autenticazione:** chi è l'agente (che opera in nome di un principal
- **Autorizzazione:** l'agente autenticato ha il permesso?

Autenticazione

Autenticare significa verificare l'identità di un soggetto (non necessariamente umano)

Modalità di autenticazione



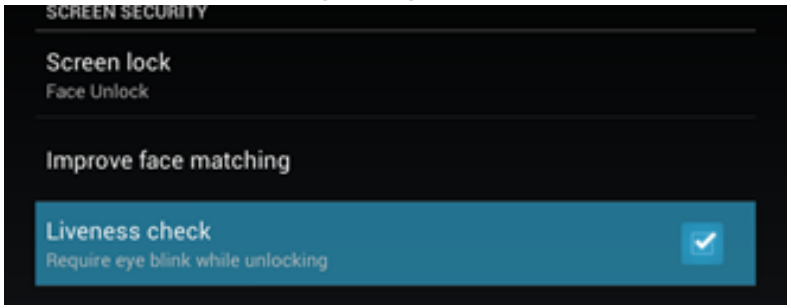
Modalità di base per l'autenticazione (di Alice) tramite rete:

1. password (ossia la conoscenza di un segreto)
2. locazione (logica o fisica) da cui proviene la richiesta di autenticazione o altre caratteristiche di Alice (biometria)
3. per mezzo di operazioni crittografiche su dati forniti dall'autenticatore (Bob).

Minacce e vulnerabilità

Alcune vulnerabilità sono intrinseche:

- Le password possono essere indovinate
- Locazioni e (bio)metria possono essere millantate



- I dati crittografici possono essere intercettati e riutilizzati (*replay attack*)

Queste minacce possono essere mitigate, ma mai eliminate del tutto. (In generale, non esiste nessuna 'sicurezza assoluta', ma solo relativa a specifici attacchi e limitatamente a quanto è disposto a 'spendere' l'attaccante)

Password guessing

- Una *password* può essere scelta in maniera prevedibile (anziché del tutto casuale) nell'insieme possibile.
- *Online guessing*: l'attaccante prova tutte le *password* possibili (*brute force*)
  `https://www.youtube.com/watch?v=G1ySd1Y0644`

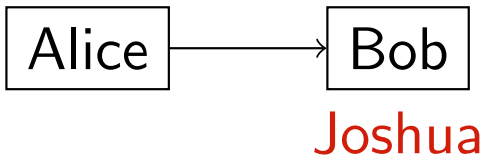
È difficile?

A volte le *password* vengono scelte in modo prevedibile.

- Le 20 piú comuni in inglese: `password1`, `abc123`, `myspace1`, `password`, `blink182`, `qwerty1`, `fuckyou`, `123abc`, `baseball1`, `football1`, `123456`, `soccer`, `monkey1`, `liverpool1`, `princess1`, `jordan23`, `slipknot1`, `superman1`, `iloveyou1`, `monkey`.
- Correlate con informazioni che ci riguardano: `mattia`, `unimi`, `comelico39`, `sara`, `...`
- Si stima che piú del 65% delle *password* abbiano meno di 8 caratteri.

La vita segreta delle *password:* http://www.nytimes.com/2014/11/19/magazine/the-secret-life-of-passwords.html (in Italiano su Internazione n.1081 12/18 dicembre 2014)

Offline guessing

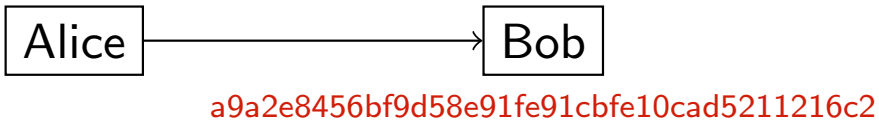


Offline guessing: che succede se l'attaccante ha accesso ai segreti di Bob?

Funzioni di *hash* crittografiche

Funzioni che è relativamente facile calcolare, ma che è impossibile invertire.

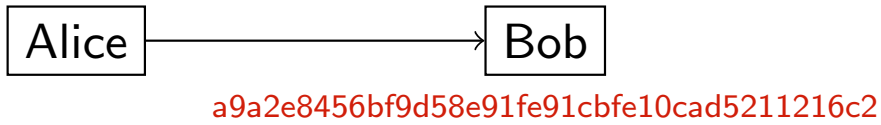
- `Joshua` $\rightsquigarrow$ `SHA1` $\rightsquigarrow$ `a9a2e8456bf9d58e91fe91cbfe10cad5211216c2`
- `joshua` $\rightsquigarrow$ `SHA1` $\rightsquigarrow$ `d6955d9721560531274cb8f50ff595a9bd39d66f`



Funzioni di *hash* crittografiche

Funzioni che è relativamente facile calcolare, ma che è impossibile invertire.

- `Joshua` $\rightsquigarrow$ `SHA1` $\rightsquigarrow$ `a9a2e8456bf9d58e91fe91cbfe10cad5211216c2`
- `joshua` $\rightsquigarrow$ `SHA1` $\rightsquigarrow$ `d6955d9721560531274cb8f50ff595a9bd39d66f`



- Sorpresa. . . !: `https://duckduckgo.com/?q=a9a2e8456bf9d58e91fe91cbfe10cad5211216c2`

Attacchi con dizionari

L'attaccante prova elenchi di parole (*dictionary attack*); si salano gli *hash* per rendere impraticabile la realizzazione di *rainbow table*.

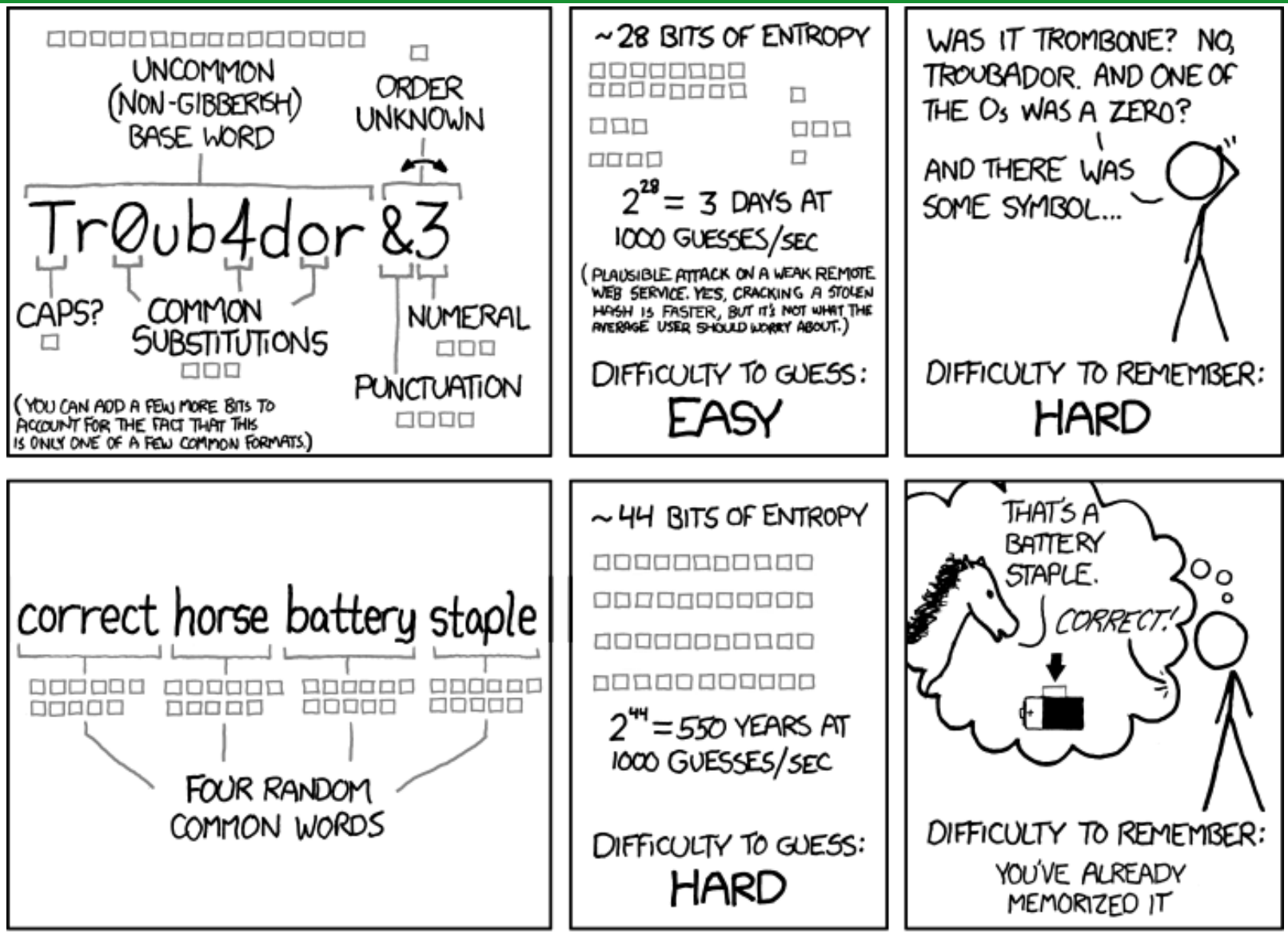
Utente	*salt*	*stored password*
Alice	42	$\mathrm{hash}(42 \vert \mathrm{password}_{\mathrm{Alice}})$

- Possibilità di intercettazione
- Utilizzo in occasioni differenti
- distribuzione iniziale delle credenziali (si fanno scadere al primo accesso)

Un PC normalissimo potrebbe essere in grado di tentare $10^6/s$, con cautele particolari si arriva a $10^9/s$. Da *password* a *passphrase*.

Parole	
45	3
183	4
986	5
3068	6
5597	7
10547	8
15032	9
18478	10
19730	11
14952	12
10066	13
5789	14
2466	15
1253	16
392	17
119	18
35	19
4	20

$26 \times 26 \times 26 \times 26 \times 26 \times 26 \times 26 \times 26$	$=$	$2,088 \cdot 10^{11}$	$\rightsquigarrow$	$2 \cdot 10^2 s$	$\approx$	3m
$4000 \times 4000 \times 4000 \times 4000$	$=$	$2,560 \cdot 10^{14}$	$\rightsquigarrow$	$2 \cdot 10^5 s$	$\approx$	71h
con le maiuscole 52^8	$=$	$5,346 \cdot 10^{13}$	$\rightsquigarrow$	$5 \cdot 10^4 s$	$\approx$	14h
con la prima maiuscola 8000^4	$=$	$4,096 \cdot 10^{15}$	$\rightsquigarrow$	$4 \cdot 10^6 s$	$\approx$	47d


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor&3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS:
EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER:
HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS:
HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER:
YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS

E se qualcuno intercetta la comunicazione?

Una buona (*i.e.*, lunga) *passphrase* difende bene dal *guessing* (*online*/*offline*). Ma l'attaccante potrebbe intercettare il segreto. Come fare?

La cifratura perfetta

La cifratura perfetta è stata inventata nel 1917 da J. Mauborgne e G. Vernam: One-time pad

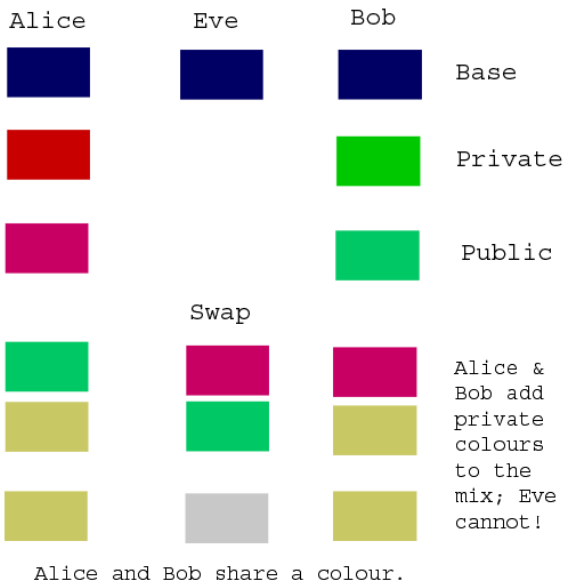
- Si produce uno stream di bit completamente random (tirando una moneta?) della stessa lunghezza del testo da cifrare
- Si fa lo XOR bit a bit con il testo in chiaro

$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$
$011 \oplus 110 = 101$ $101 \oplus 110 = 011$
$(c \oplus x) \oplus x = c$

Problema: la chiave deve essere veramente imprevedibile e usata una sola volta.

Ma come si fa a comunicare la password?

La prima soluzione "pubblica" per questo tipo di problemi è del 1976 (Diffie e Hellman, Premio Turing 2016).



(http://maths.straylight.co.uk/archives/108)

In realtà con l'aritmetica modulare

1. Alice e Bruno scelgono pubblicamente un numero primo p (grande) e un numero n
2. Alice sceglie un numero a caso x e calcola $A = n^x \mod p$ e manda A a Bruno
3. Bruno sceglie un numero a caso y e calcola $B = n^y \mod p$ e manda B ad Alice
4. Alice calcola $B^x \mod p = n^{yx} \mod p = S$
5. Bruno calcola $A^y \mod p = n^{xy} \mod p = S$

Alice e Bruno si trovano a condividere la chiave segreta S.

Esercizio

Codificare il protocollo Diffie-Hellman per scambiarsi una password che poi verrà usata per cifrare una parola tramite XOR. La parola codificata dovrà essere comunicata ad alta voce e qualcuno potrebbe intercettarla...

- potenza `Math.pow(b,e)`
- modulo `%`
- xor `^`
- `charCodeAt` trasforma un carattere nel codice numerico corrispondente
- `fromCharCode` trasforma un codice numerico nel corrispondente carattere

Il *single sign-on* (SSO)

L'idea di avere credenziali che permettono l'accesso a sistemi diversi è appetibile per una serie di ragioni

- Riduce il problema di trovare un buon segreto (sufficientemente casuale, ecc.)
- Riduce l'*overhead* totale di gestione degli accessi
- Permette la gestione centralizzata degli accessi, piú semplice da manutenere

(Aumenta la criticità delle credenziali, però)
Il SSO sembra particolarmente attraente in situazioni come i servizi *web* a bassa criticità, con decine di *password* da ricordare (e utenti poco sensibili al problema)

Lo schema OpenID

1. Alice si connette al sito `meraviglie.org`
2. Per accedere comunica che ha un account su `faccialibro.com/openid/alice`
3. Alice accede (con credenziali tradizionali) a `faccialibro.com` e produce un certificato d'accesso
4. `meraviglie.org` accetta il certificato d'accesso come credenziale d'autenticazione

Problemi di OpenID

I principali *threat*

- Facilità di allestire inganni di tipo **phishing**
- Privacy

Alternative migliori

In realtà sembra meglio avere credenziali multiple e gestirle con modalità come:

- PasswordSafe (`http://passwordsafe.sf.net`): un db criptato
- SuperGenPass (`http://supergenpass.com/`): un'unica *password* viene giustapposta ad un identificatore del sito e la vera *password* è ottenuta con una funzione *hash*. L'idea è ottima, la realizzazione ha diverse vulnerabilità: meglio usare alternative piú *crypto-savvy*, p. es. (`http://hpass.chmd.fr/`).

L'unico che conta di sicuro è il valore che voi attribuite ai dati

Un attaccante può sempre trasformare il valore che qualcosa ha per voi, in potenziale valore per un attacco. E le *password* possono essere un'arma dell'attaccante...

Ransomware

L'attaccante sequestra i vostri dati, rendendoli inaccessibili con tecniche crittografiche sofisticate; poi chiede un riscatto.

Una volta subito l'attacco, l'unica difesa è avere un accesso alternativo ai dati $\rightsquigarrow$ backup regolari e verificati!.