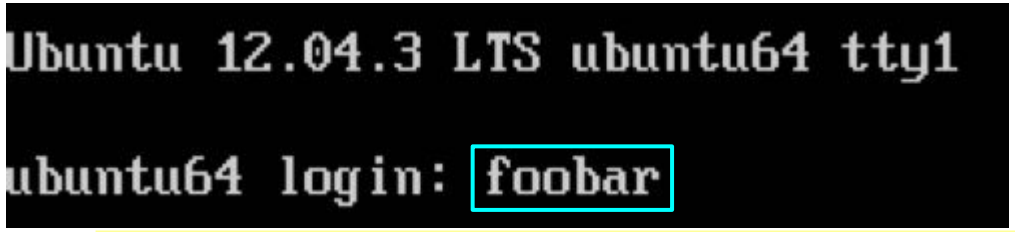
3. Authentication

Computer Security Courses @ POLIMI

Identification *vs.* **Authentication**

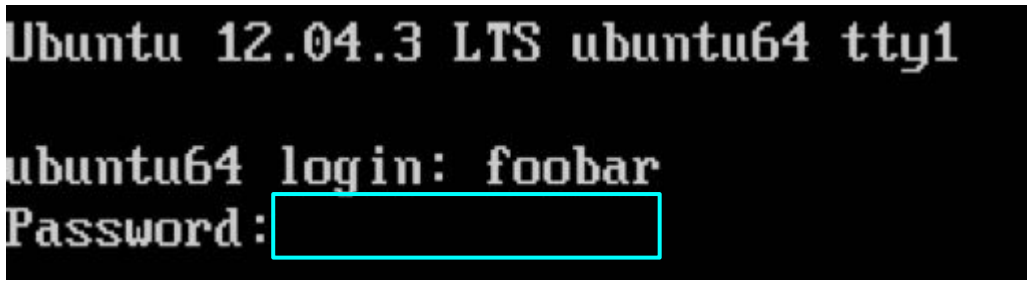
Identification: an entity declares its identifier

- **Examples:** *"I am Stefano"*, *"I am Michele"*



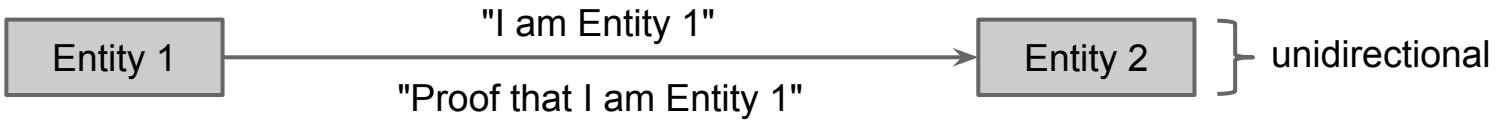
Authentication: the *entity* provides a *proof* that verifies its identity.

- **Examples:** *"Here is Stefano's ID card"*



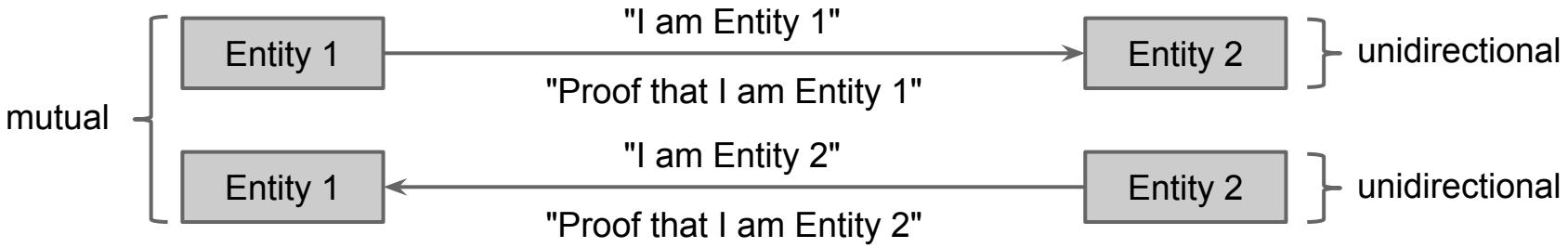
Authentication

Can be *unidirectional*



Authentication

Can be *unidirectional* or *bidirectional (mutual)*.



Can happen between any entity:

- Human to human
- Human to computer
- Computer to computer

Foundation for the subsequent *authorization* phase (see next class).

Three Factors of Authentication

Something that the entity ***knows*** (to know)

1. **Example:** password, PIN, secret handshake.

Something that the entity ***has*** (to have)

2. **Example:** Door key, smart card, token.

Something that the entity ***is*** (to be)

3. **Example:** Face, voice, fingerprints.

Humans:

Machines:

Multi-factor authentication uses two or three factors.

Three Factors of Authentication

Something that the entity ***knows*** (to know)

1. **Example:** password, PIN, secret handshake.

Something that the entity ***has*** (to have)

2. **Example:** Door key, smart card, token.

Something that the entity ***is*** (to be)

3. **Example:** Face, voice, fingerprints.

(to humans)

Humans: (3) more used than (2), more used than (1).

Machines:

Multi-factor authentication uses two or three factors.

Three Factors of Authentication

Something that the entity ***knows*** (to know)

1. **Example:** password, PIN, secret handshake.

Something that the entity ***has*** (to have)

2. **Example:** Door key, smart card, token.

Something that the entity ***is*** (to be)

3. **Example:** Face, voice, fingerprints.

Humans: (3) more used than (2), more used than (1).

Machines: (1) more used than (2), more used than (3).

Multi-factor authentication uses two or three factors.

Three Factors of Authentication

The three factors should be independent: trying to access on a website on a smartphone and getting the 2FA code on the smartphone itself has collapsed the two factors

Something that the entity ***knows*** (to know)

1. **Example:** password, PIN, secret handshake.

Something that the entity ***has*** (to have)

2. **Example:** Door key, smart card, token.

Something that the entity ***is*** (to be)

3. **Example:** Face, voice, fingerprints.

Humans: (3) more used than (2), more used than (1).

Machines: (1) more used than (2), more used than (3).

Usability

Multi-factor authentication uses two or three factors.

The "to know" Factor

Passwords and PINs

The *"to know"* Factor: Passwords

Advantages

- Low cost,
- ease of deployment,
- low technical barrier.

Disadvantages

The *"to know"* Factor: Passwords

Advantages

- Low cost,
- ease of deployment,
- low technical barrier.

Disadvantages

Secrets can be

a. stolen/snooped
b. guessed
c. cracked (enumerated)

The *"to know"* Factor: Passwords

Advantages

- Low cost,
- ease of deployment,
- low technical barrier.

Disadvantages

Secrets can be

a. stolen/snooped
b. guessed
c. cracked (enumerated)

The *"to know"* Factor: Passwords

Advantages

- Low cost,
- ease of deployment,
- low technical barrier.

Disadvantages

Secrets can be

a. stolen/snooped
b. guessed --> you exploit the fact that passwords are actually not uniformed distributed (birthdays, names of relatives) by guessing
c. cracked (enumerated)

The *"to know"* Factor: Passwords

Advantages

- Low cost,
- ease of deployment,
- low technical barrier.

Disadvantages

Secrets can be

a. stolen/snooped
b. guessed
c. cracked (enumerated) --> try all the combinations

The *"to know"* Factor: Passwords

Advantages

- Low cost,
- ease of deployment,
- low technical barrier.

Disadvantages

Secrets can be

a. stolen/snooped
b. guessed
c. cracked (enumerated)

Countermeasures *(i.e., costs)*

Enforce passwords that

- *change/expire* frequently
- are *long* and have a *rich character set*
- are *not related to the user*

Estimate the most likely attack in the scenario. ⟺ Accordingly choose the countermeasure(s) that are worth asking users to adhere to (remember indirect costs?)

Why are Countermeasures Costs?

Humans are not machines

- Inherently *unable to keep secrets*
- Hard to *remember* complex passwords
- Can't pick *unlimited* countermeasures
  - how to choose?

Why are Countermeasures Costs?

Humans are not machines

- Inherently *unable to keep secrets*
- Hard to *remember* complex passwords

Can't pick *unlimited* countermeasures

- how to choose?

Countermeasure guideline: important, may help, unimportant

Against **snooping**	Against **cracking**	Against **guessing**
complexity	*complexity*	*complexity*
change	*change*	*change*
being related to users	being related to users	*not being related to users*

Why are Countermeasures Costs?

Humans are not machines

- Inherently *unable to keep secrets*
- Hard to *remember* complex passwords
- Can't pick *unlimited* countermeasures
  - how to choose?

Countermeasure guideline: important, may help, unimportant

Against **snooping**	Against **cracking**	Against **guessing**
complexity	*complexity*	*complexity*
change	*change*	*change*
being related to users	being related to users	*not being related to users*

Why are Countermeasures Costs?

Humans are not machines

- Inherently *unable to keep secrets*
- Hard to *remember* complex passwords
- Can't pick *unlimited* countermeasures
  - how to choose?

Countermeasure guideline: important, may help, unimportant

Against **snooping**	Against **cracking**	Against **guessing**
complexity	*complexity*	*complexity*
change	*change*	*change*
being related to users	being related to users	*not being related to users*

Why are Countermeasures Costs?

Humans are not machines

- Inherently *unable to keep secrets*
- Hard to *remember* complex passwords
- Can't pick *unlimited* countermeasures
  - how to choose?

Countermeasure guideline: important, may help, unimportant

Against **snooping**	Against **cracking**	Against **guessing**
complexity	*complexity*	*complexity*
change	*change*	*change*
being related to users	being related to users	*not being related to users*

User Education and Password Complexity

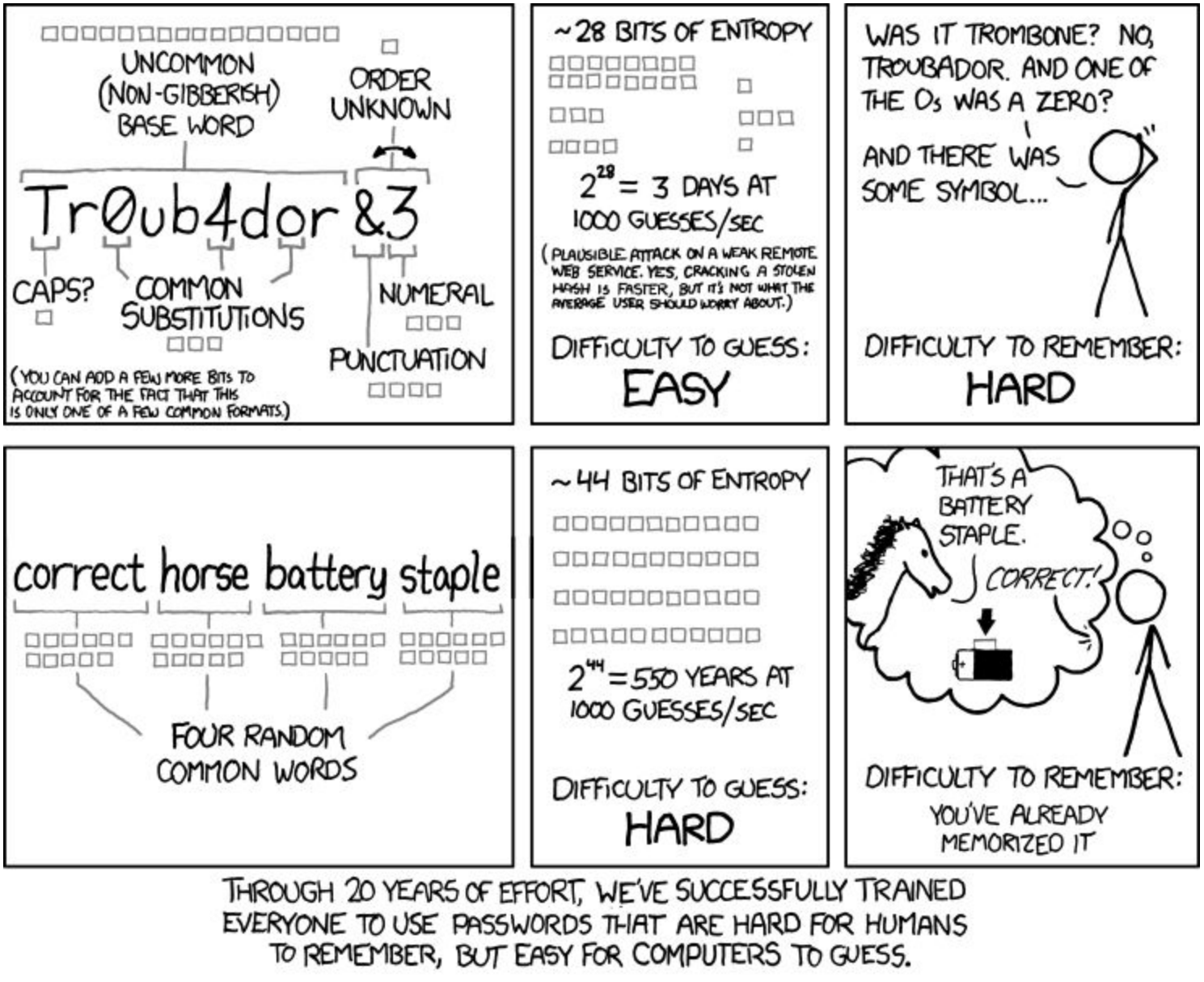
User education: "human" == "weak link".

- enforce *strong* passwords in the process.
- enforce password *expiration/change* policies
- use password meters to balance *usability*.

Password complexity

- must h4v3 4 r1ch, ch4r4ct3r, s3t!
- mUsT hAvE a MiXeD cAsE
- muuuuuuust beeeeeeeeee looooooooong enooooooough

It's better to have a long password composed of 4 or 5 independent extracted english words than a lower password but composed of capitals, special characters, numbers etc... It can be seen by computing the entropy, that is the number of bits needed for encoding it.



The limit in length of a password is non-sense from a security point of view. Usually it is done because there is a fixed amount of space where to store the password.

Password Meters

passfault

they look at the equivalent distribution that is they are trying to guess the distribution out of a simple server

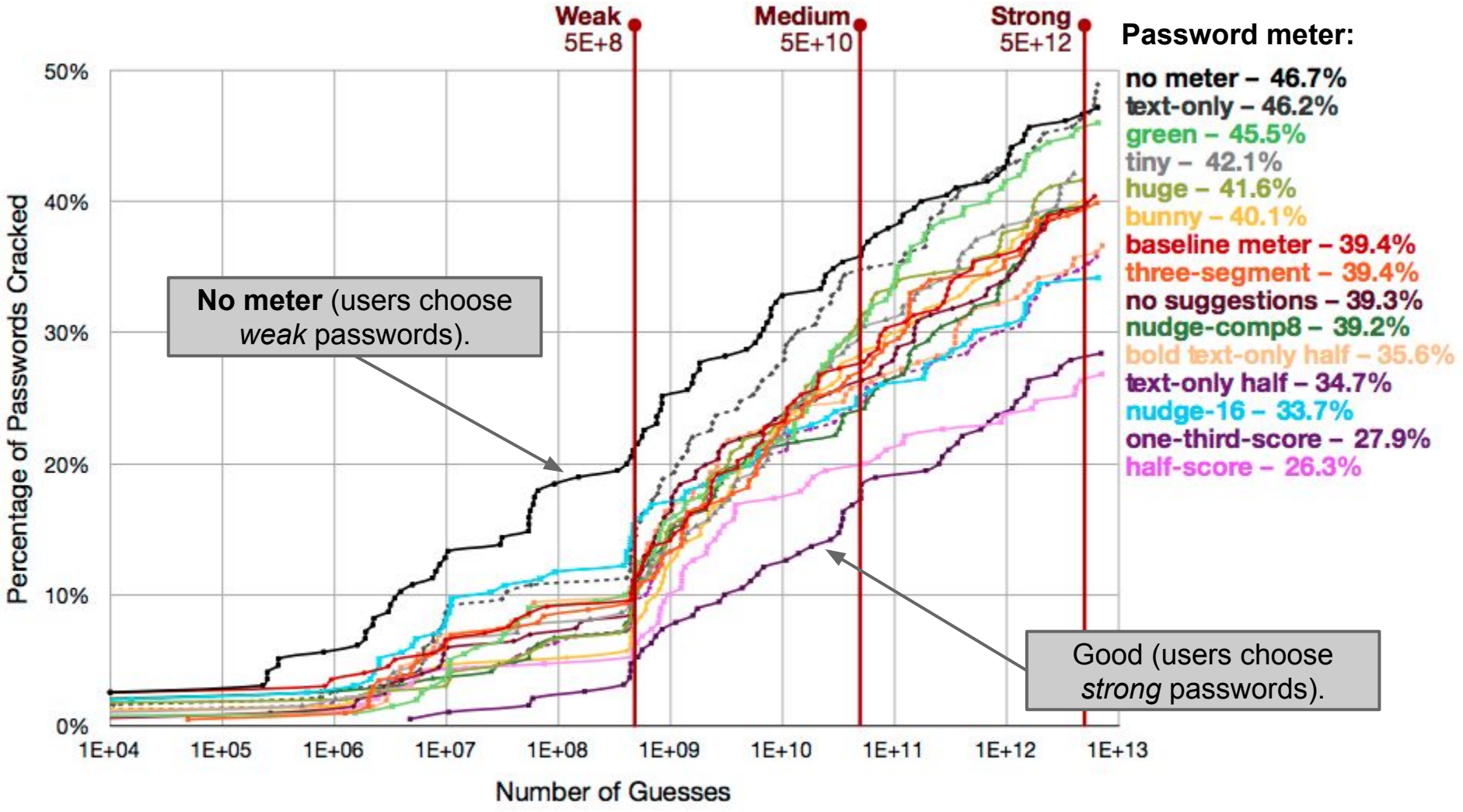


Figure 3: This graph contrasts the percentage of passwords that were cracked in each condition. The x-axis, which is logarithmically scaled, indicates the number of guesses made by an adversary, as described in Section 2.4. The y-axis indicates the percentage of passwords in that condition cracked by that particular guess number.

B. Ur, P.G. Kelley, S. Komanduri, J. Lee, M. Maass, M. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L.F. Cranor. How does your password measure up? The effect of strength meters on password creation. USENIX Security 2012.

You want to provide a proof to the server that you know the password without revealing the password, and this proof should be fresh every single time.

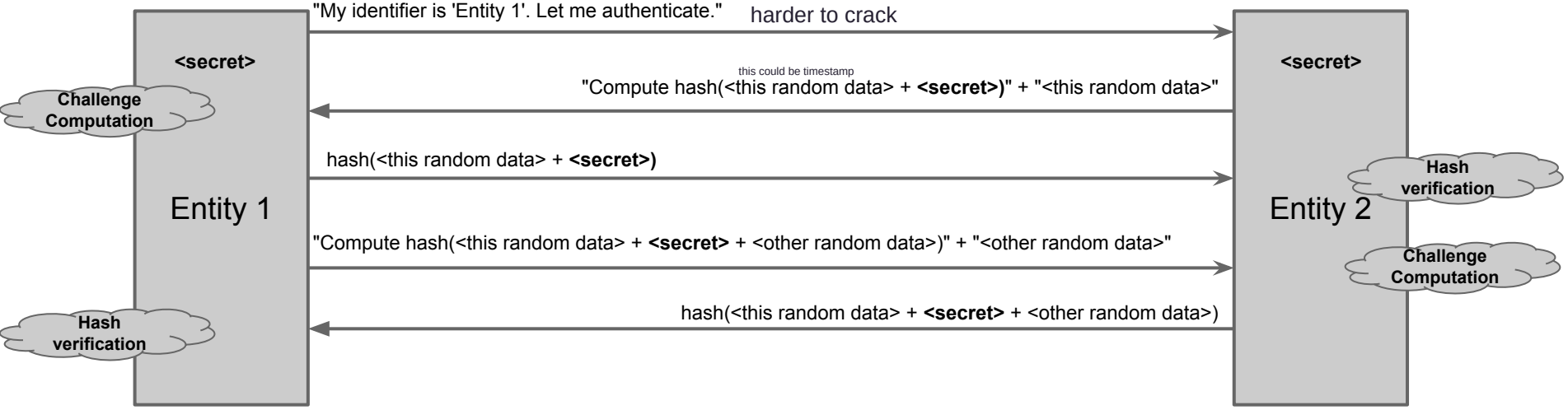
Secure Password Exchange

Authentication is about sharing a *secret*.

How to minimize the risk that secrets get *stolen*?

- use *mutual authentication* if possible
- use a ***challenge-response*** scheme
  - use random data to avoid *replay attacks*

With password salting, a random piece of data is added to the password before it runs through the hashing algorithm, making it unique and harder to crack



Secure Password Storage

OS stores a file with usernames and passwords.

An attacker could try to compromise the confidentiality and integrity of this password file

How to minimize the risk that secrets get *stolen*?

- Cryptographic protection
  - Never store passwords in clear (see next classes: hashing + salting to mitigate dictionary attacks)
- Access control policies (privileges to w/r)
- Never disclose secrets in password-recovery schemes.

Caching problem (information is held in intermediate storage locations)

The "to have" Factor

Tokens, smart cards, smart phones.

Password manager should keep passwords locally and not on a server
and should not store the main password in main memory, otherwise
if the system shuts down not gracefully, the password remains on memory
and could be used if someone has accesses to the database.

The *"to have"* Factor

User must prove that it *possesses* something.

Advantages

- Human factor (less likely to hand out a key),
- relatively low cost,
- good level of security.

Disadvantages

- Hard to deploy,
- can be lost or stolen.

Countermeasures

- none
- use with second factor.

Uses a mac, which is an abuse since mac is for integrity but since the verifier it's the one that generates the code (es: the bank), there is no problem

Example Classic Technologies

One-time password generators:

- *Secret key* + *counter* synchronized with the host.
- Client: MAC-compute(counter, key).
- Host: MAC-verify(counter, key).
- Check that the counter is the expected one.
- The counter changes every 30–60 seconds.

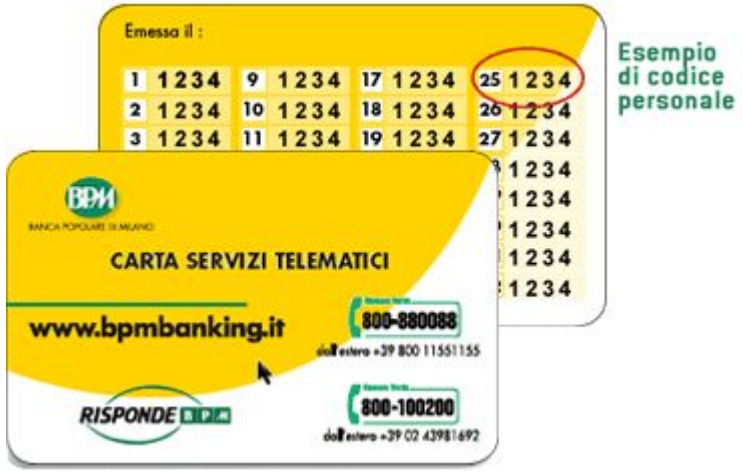
Application examples: online banking, admin console (e.g., Amazon AWS).

Smart cards (also w/ embedded reader in USB keys)

- CPU + non-volatile RAM with a *private key*.
- The smart card authenticates itself to host via a ***challenge-response*** protocol.
- Uses the *private key* to sign the challenge.
- The *private key* does not leave the device.
- Should be **tamper proof** to some extent.

Application examples: credit cards (+PIN).

Static OTP lists (cheaper alternative)



- Known to both client and host.
- Host chooses a challenge: random numbers (e.g., "second digit of the 14th cell").
- The client transmits the response (hopefully, over an encrypted channel).
- The host should not keep the list in clear (e.g., hashing).

Modern Technology: TOTP

Software that implements the same functionality of password generators

Modern Technology: TOTP

-> they compute mac on current time (current minute)

Software that implements the same functionality of password generators:

- Key difference
  - password generators are *closed, embedded* systems.
  - password-generation *apps* work on *general-purpose* sw/hw platforms.
- What if the device is infected by a malicious app: Dmitrienko et al., When More Becomes Less: On the (In)Security of Mobile Two-Factor Authentication, FC 2014

EUROPOL

SIM SWAPPING – A MOBILE PHONE SCAM

SIM swapping occurs when a fraudster, using social engineering techniques, takes control over your mobile phone SIM card using your stolen personal data.

HOW DOES IT WORK?

A fraudster obtains the victim's personal data through e.g. data breaches, phishing, social media searches, malicious apps, online shopping, malware, etc.

With this information, the fraudster dupes the mobile phone operator into porting the victim's mobile number to a SIM in his possession

The fraudster can now receive incoming calls and text messages, including access to the victim's online banking

The victim will notice the mobile phone lost service, and eventually will discover they cannot login to their bank account

https://www.europol.europa.eu/activities-services/public-awareness-and-prevention-guides/sim-swapping-%E2%80%93-mobile-phone-scam

The "to be" Factor.

Biometric authentication.

The *"to be"* Factor: Biometric.

User must prove that it has some specific *characteristics*.

Advantages

- high level of security,
- requires no extra hardware to carry around.

Disadvantages	Countermeasures
Hard to deploy,	none
probabilistic matching,	none
invasive measurement,	none
can be cloned,	none (see next slides)
bio-characteristics change,	re-measure often,
privacy sensitivity,	secure the process,
users with disabilities.	need alternate (weaker?)

Technology examples

Extract the characteristics (i.e., features) of:

- Fingerprints (http://www.freedesktop.org/wiki/Software/fprint/)
- Face geometry (https://code.google.com/p/pam-face-authentication/)
- Hand geometry (palm print)
- Retina scan
- Iris scan
- Voice analysis
- DNA
- Typing dynamics (http://flyer.sis.smu.edu.sg/ndss13-tey.pdf)

Example: Fingerprint

- *Enrollment*: reference sample of the user's fingerprint is acquired at a fingerprint reader.
- Features are derived from the sample.
  - *Fingerprint minutiae*: end points of ridges, bifurcation points, core, delta, loops, whorls, ?
  - For higher accuracy, record features for more than one finger and different positions.
- Feature vectors are stored in a secure database.
- When the user logs on, a new reading of the fingerprint is taken; features are compared against (similarity) the reference features. User is accepted if match is above a predefined threshold.

Main issue: false positives and false negatives

Consumer-level Biometric Auth

http://cryptome.org/fake-prints.htm

Manufacturer	Model	Technology	Date	Difficulty
Identix	TS-520	Optical	Nov. 1990	First attempt
Fingermatrix	Chekone	Optical	Mar. 1994	Second attempt
Dermalog	DemalogKey	Optical	Feb.1996	First attempt
STMicroelectronics	TouchChip	Solid state	Mar. 1999	First attempt
Veridicon	FPS110	Solid state	Sept.1999	First attempt
Identicator	DFR200	Optical	Oct. 1999	First attempt

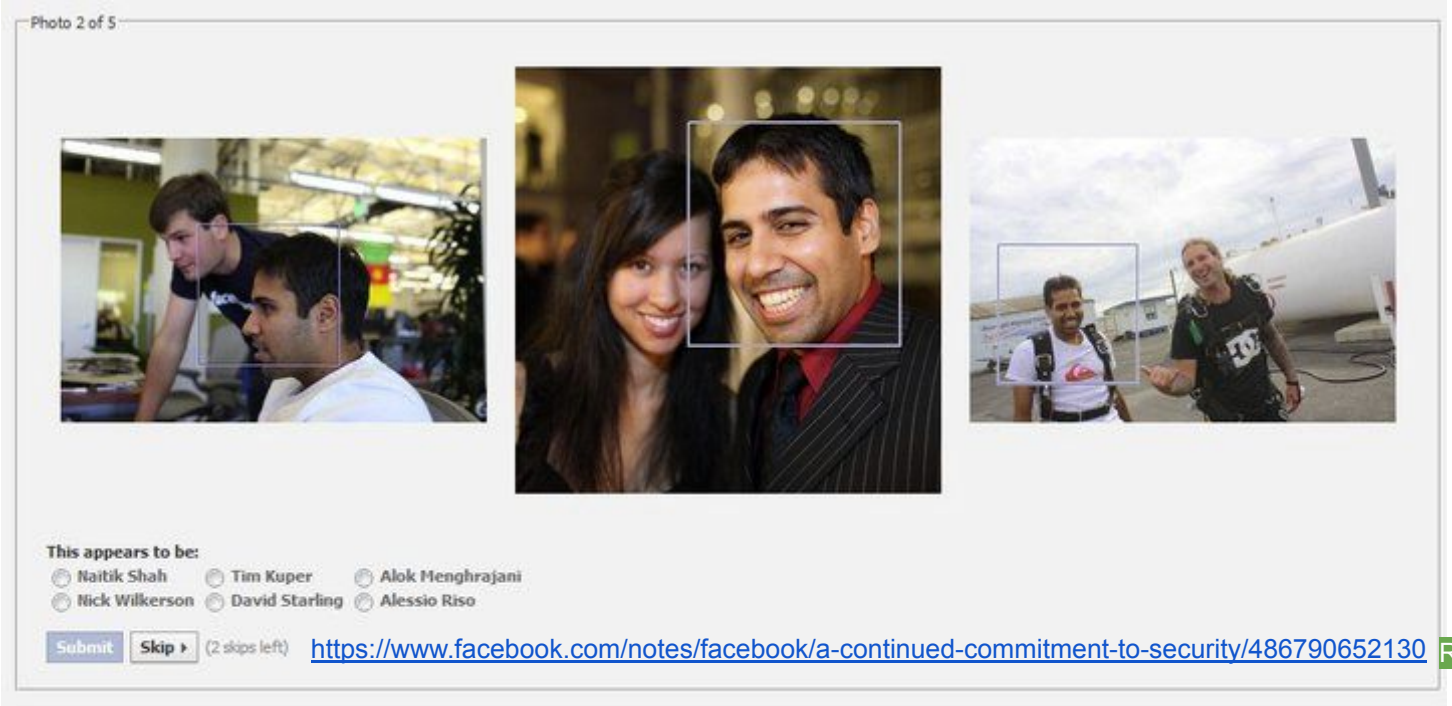
20 september 2013 RELEASED

21 september 2013 CRACKED

http://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid

The *"social"* Factor: Who you know.

Alice must prove that she *knows someone.*



https://www.facebook.com/notes/facebook/a-continued-commitment-to-security/486790652130 RELEASED

Papers

- H. Kim, J. Tang, and R. Anderson. *Social authentication: harder than it looks.* In Proceedings of the 2012 Financial Cryptography and Data Security conference.

CRACKED J. Polakis, M. Lancini, G. Kontaxis, F. Maggi, S. Ioannidis, A. Keromytis, S. Zanero, *All Your Face Are Belong to Us: Breaking Facebook's Social Authentication.* In Proceedings of 2012 Annual Computer Security Applications Conference.

Single Sign On

Problem: managing and remembering multiple passwords is complex.

- Users re-use passwords over multiple sites,
- Password policies replicated ($$$).

Solution: 1 identity, 1-2 auth. factors, 1 trusted host.

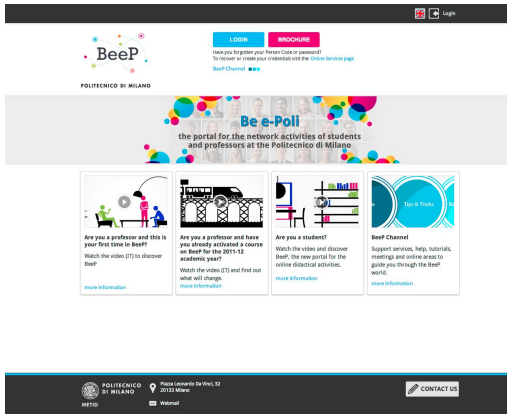
- elect a trusted host,
- users authenticate (sign on) on the trusted host,
- other hosts ask the trusted host if a user is authenticated.

Example: Shibboleth (AunicaLogin)

User (e.g., you)

1. I am "John G. Student"

6. You can access the service.



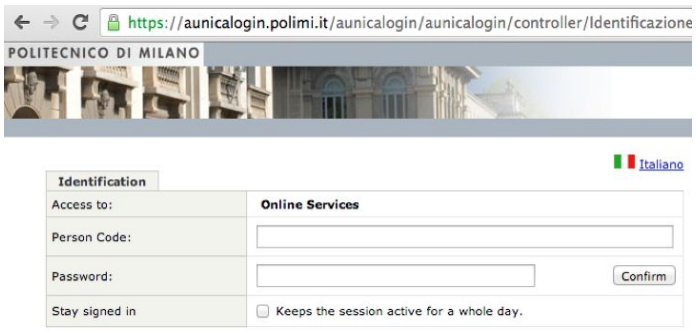
Service (e.g., BeeP)

3. I need your credentials.

2. Is "John G. Student" a student?

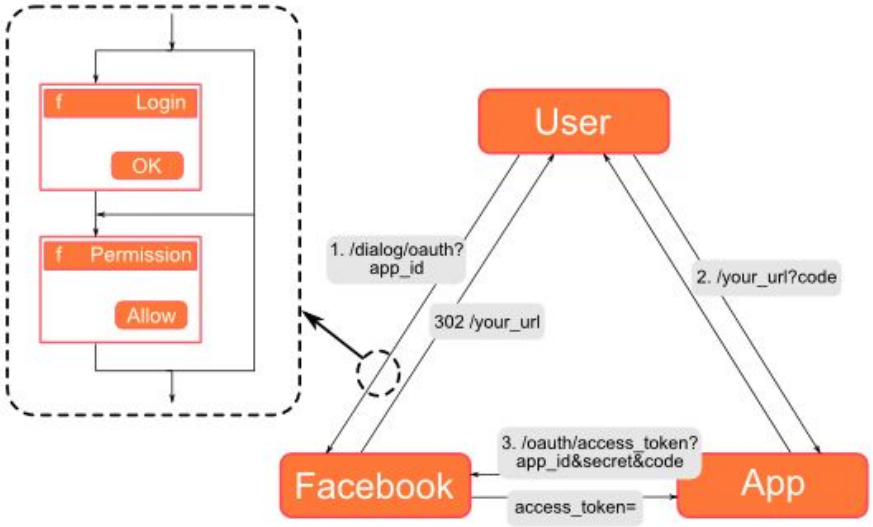
4. Here is my username and password

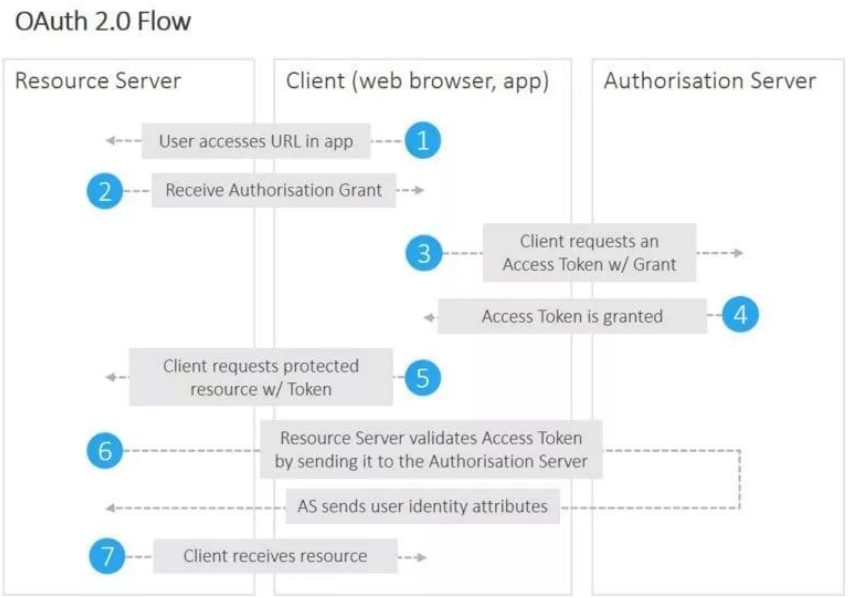
5. Yes. I confirm.



Identity Provider (e.g., AunicaLogin)

Example: OAuth2 Flow (Facebook)





Single Sign On: challenges

Single point of *trust*: the trusted server.

- If compromised, all sites are compromised.
- Password reset scheme must be bulletproof.
  - Email is the trusted element

    Kontaxis G. et al., *SAuth: Protecting User Accounts from Password Database Leaks*. In Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS), 2013.

Difficult to get right for the developers.

- The flow is complex to implement.
- Libraries exist, but they can be bugged.

http://homakov.blogspot.it/2014/02/how-i-hacked-github-again.html

Conclusions

Identification, *authentication* and *authorization* are three distinct, yet inter-dependent, concepts.

There are *three* types of authentication *factors*, which should be used in *combination*.

Passwords are increasingly showing their limits.

New authentication schemes are promising, but should be used with care.