UNIVERSITY OF AMSTERDAM

Jaap van Ginkel

Security of Systems and Networks

5 Oktober 2023, Authentication

The problem illustrated

Thanks to Ton Verschuren

Terminology

- Identification: (“who are you?“)
- Authentication: (“prove it!”) (AUTHN)
- Authorization: (“these you can do”) (AUTHZ)
- Different levels of authentication:
  - Weak (something you know)
  - Strong (something you have and something you know)
  - Biometrics (something you are)

Examples

- **Something you Know**
  - password
  - Address/birthday combination
  - Pin code
- **Something you Have**
  - Key
  - Bank card
  - Drivers license
  - Letter
- **Something you Are**
  - Finger print
  - DNA profile
  - Iris print

Access Control

- Two parts to access control
- **Authentication:** Are you who you say you are?
  - Determine whether access is allowed
  - Authenticate human to machine
  - Or authenticate machine to machine
- **Authorization:** Are you allowed to do that?
  - Once you have access, what can you do?
  - Enforces limits on actions
- Note: “access control” often used as synonym for authorization

Are You Who You Say You Are?

- How to authenticate human a machine?
- Can be based on…
  - Something you **know**
    - For example, a password
  - Something you **have**
    - For example, a smartcard
  - Something you **are**
    - For example, your fingerprint

Something You Know

- Passwords
- Lots of things act as passwords!
  - PIN
  - Social security number
  - Mother’s maiden name
  - Date of birth
  - Name of your pet, etc.

Trouble with Passwords

- “Passwords are one of the biggest practical problems facing security engineers today.”
- “Humans are incapable of securely storing high-quality cryptographic keys, and they have unacceptable speed and accuracy when performing cryptographic operations. (They are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed.)”

Why Passwords?

- Why is “something you know” more popular than “something you have” and “something you are”?
- **Cost**: passwords are free
- **Convenience**: easier for admin to reset pwd than to issue a new thumb

Keys vs Passwords

- **Crypto keys**
- Spse key is 64 bits
- Then 2^{64} keys
- Choose key at random…
- …then attacker must try about 2^{63} keys

- **Passwords**
- Spse passwords are 8 characters, and 256 different characters
- Then $256^8 = 2^{64}$ pwds
- Users do not select passwords at random
- Attacker has far less than 2^{63} pwds to try (**dictionary attack**)

Good and Bad Passwords

- Bad passwords
  - frank
  - Fido
  - password
  - 4444
  - Pikachu
  - 102560
  - AustinStamp
- Good Passwords?
  - jflej,43j-EmmL+y
  - 09864376537263
  - P0kem0N
  - FSa7Yago
  - 0nceuP0nAt1m8
  - PokeGCTall150

User name Password

- Weak authentication
- User Friendly
  - Works everywhere
- Very common
- Alternatives difficult
- Extended Life span
  - Awareness
  - Safe implementation

Common passwords

- 123456 1375
- Ficken 404
- 12345 367
- Hallo 362
- 123456789 260
- Schatz 253
- 12345678 215

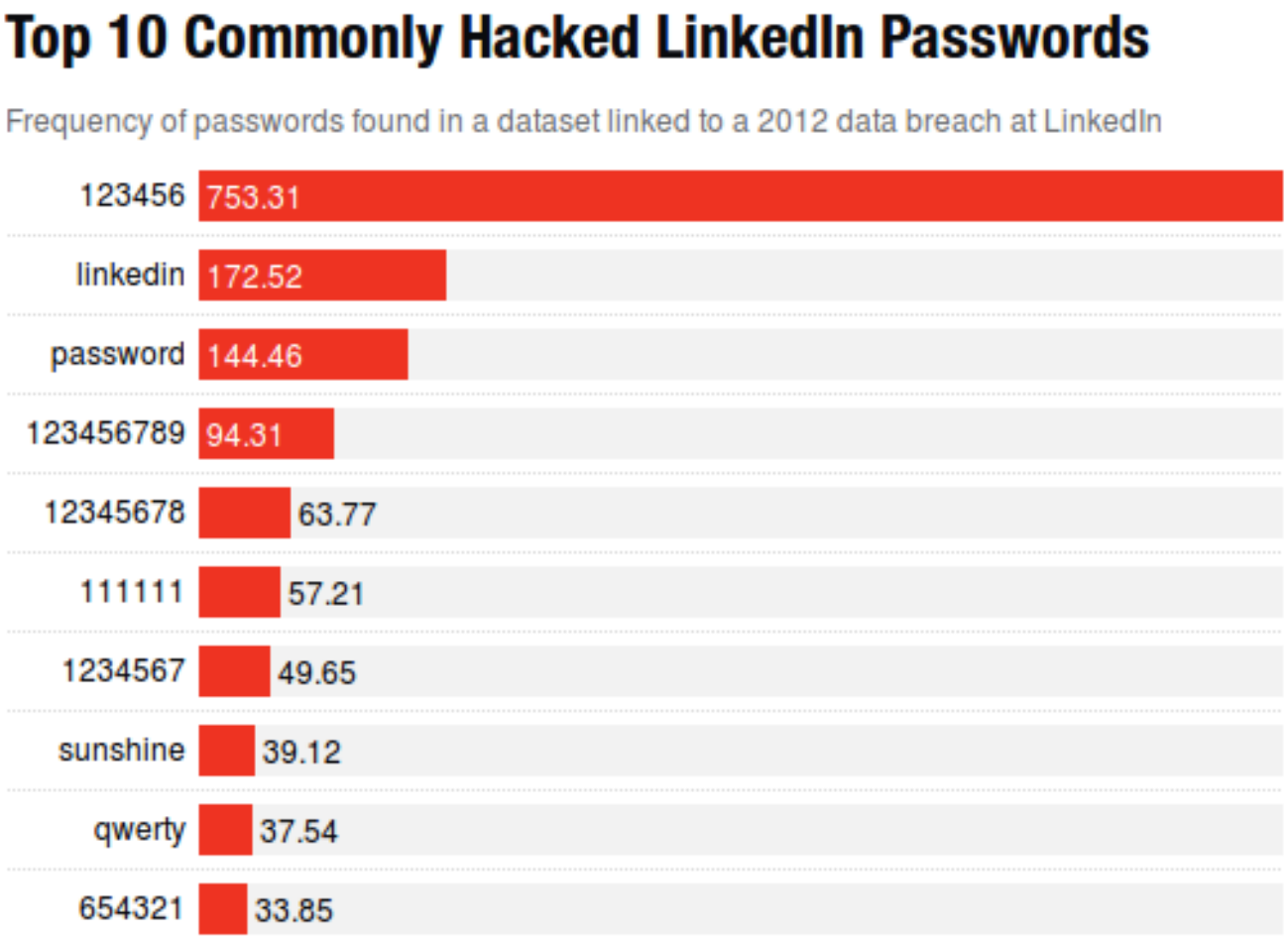
Common passwords

- 123456 1375
- Ficken 404
- 12345 367
- Hallo 362
- 123456789 260
- Schatz 253
- 12345678 215

Common passwords

- 123456 1375
- Ficken 404
- 12345 367
- Hallo 362
- 123456789 260
- Schatz 253
- 12345678 215

Common passwords



Chocolate passwords

- 2004 Research Liverpool Street Station
  - 70% gave up password for chocolate
- http://news.bbc.co.uk/2/hi/technology/3639679.stm

Password Experiment

- Three groups of users - each group advised to select passwords as follows
  - **Group A:** At least 6 chars, 1 non-letter
  - **Group B:** Password based on passphrase
  - **Group C:** 8 random characters

Password Experiment

- Three groups of users - each group advised to select passwords as follows
  - **Group A:** At least 6 chars, 1 non-letter
  - **Group B:** Password based on passphrase
  - **Group C:** 8 random characters

winner →

- Results
  - **Group A:** About 30% of pwds easy to crack
  - **Group B:** About 10% cracked
    - Passwords easy to remember
  - **Group C:** About 10% cracked
    - Passwords hard to remember

Attacks on Passwords

- Attacker could…
  - Target one particular account
  - Target any account on system
  - Target any account on any system
  - Attempt denial of service (DoS) attack
- Common attack path
  - Outsider → normal user → administrator
  - May only require **one** weak password!

Password Retry

- Suppose system locks after 3 bad passwords. How long should it lock?
  - 5 seconds
  - 5 minutes
  - Until SA restores service
- What are +'s and -'s of each?

Password File?

- Bad idea to store passwords in a file
- But we need to verify passwords
- Cryptographic solution: **hash** the pwd
  - Store y = h(password)
  - Can verify entered password by hashing
  - If Trudy obtains “password file,” she does not obtain passwords
- But Trudy can try a *forward search*
  - Guess x and check whether y = h(x)

Dictionary Attack

- Trudy pre-computes h(x) for all x in a **dictionary** of common passwords
- Suppose Trudy gets access to password file containing hashed passwords
  - She only needs to compare hashes to her pre-computed dictionary
  - After one-time work, actual attack is trivial
- Can we prevent this attack? Or at least make attacker's job more difficult?

Salt

- Hash password with **salt**
- Choose random salt s and compute

  y = h(password, s)

  and store (s,y) in the password file
  - Note: The salt s is not secret
  - Easy to verify salted password
  - But Trudy must re-compute dictionary hashes for each user
    - Lots more work for Trudy!

Password Cracking: Do the Math

- Assumptions:
- Pwds are 8 chars, 128 choices per character
  - Then $12^{88} = 2^{56}$ possible passwords
- There is a **password file** with 2^{10} pwds
- Attacker has **dictionary** of 2^{20} common pwds
- **Probability** of 1/4 that a pwd is in dictionary
- Work is measured by number of hashes

Password Cracking: Case I

- Attack 1 password without dictionary
  - Must try $2^{56}/2 = 2^{55}$ on average
  - Like exhaustive key search
- Does **salt** help in this case?

Password Cracking: Case II

- Attack 1 password with dictionary
- With **salt**
  - Expected work: 1/4 (2^{19}) + 3/4 (2^{55}) = $2^{54.6}$
  - In practice, try all pwds in dictionary…
  - …then work is at most 2^{20} and probability of success is 1/4
- What if **no salt** is used?
  - One-time work to compute dictionary: 2^{20}
  - Expected work still same order as above
  - But with precomputed dictionary hashes, the "in practice" attack is free…

Password Cracking: Case III

- Any of 1024 pwds in file, **without** dictionary
  - Assume all 2^{10} passwords are distinct
  - Need 2^{55} **comparisons** before expect to find pwd
- If **no salt** is used
  - Each computed hash yields 2^{10} comparisons
  - So expected work (hashes) is $2^{55}/2^{10} = 2^{45}$
- If **salt** is used
  - Expected work is 2^{55}
  - Each comparison requires a hash computation

Password Cracking: Case IV

- Any of 1024 pwds in file, **with** dictionary
  - Prob. one or more pwd in dict.: $1 - (3/4)^{1024} = 1$
  - So, we ignore case where no pwd is in dictionary
- If **salt** is used, expected work less than 2^{22}
  - See book, or slide notes for details
  - Approximate work: size of dict. / probability
- What if **no salt** is used?
  - If dictionary hashes not precomputed, work is about $2^{19}/2^{10} = 2^9$

Other Password Issues

- Too many passwords to remember
  - Results in password reuse
  - Why is this a problem?
- Who suffers from bad password?
  - Login password vs ATM PIN
- Failure to change default passwords
- Social engineering
- Error logs may contain “almost” passwords
- Bugs, keystroke logging, spyware, etc.

Passwords

- The bottom line…
- **Password cracking is too easy**
  - One weak password may break security
  - Users choose bad passwords
  - Social engineering attacks, etc.
- Trudy has (almost) all of the advantages
- All of the math favors bad guys
- Passwords are a **BIG** security problem
  - And will continue to be a big problem

No More Leaks

No More Leaks is a project under which the police share data with private partners to prevent login credentials being misused. Login credentials constitute personal data. The pages below provide more information on what the police and participating parties do with that data.

Password Cracking Tools

- Popular password cracking tools
  - Hashcat
  - Password Crackers
  - Password Portal
  - L0phtCrack and LC4 (Windows)
  - John the Ripper (Unix)
- Admins should use these tools to test for weak passwords since attackers will
- Good articles on password cracking
  - Passwords - Conerstone of Computer Security
  - Passwords revealed by sweet deal

CAPTCHA

Turing Test

- Proposed by Alan Turing in 1950
- Human asks questions to a human and a computer, without seeing either
- If questioner cannot distinguish human from computer, computer passes
- This is the **gold standard** in AI
- No computer can pass this today
  - But some claim they are close to passing

CAPTCHA

- **CAPTCHA**
  - **C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part
- **C**ompletely **A**utomated — test is generated and scored by a computer
- **P**ublic — program and data are public
- **T**uring test to tell… — humans can pass the test, but machines cannot
  - Also known as **HIP** == **H**uman **I**nteractive **P**roof
- Like an inverse Turing test (sort of…)

CAPTCHA Paradox?

- "…CAPTCHA is a program that can generate and grade tests that it itself cannot pass…"
- "…much like some professors…"
- Paradox — computer creates and scores test that it itself cannot pass!
- CAPTCHA purpose?
  - Only humans get access (not bots/computers)
- So, CAPTCHA is for **access control**

CAPTCHA Uses?

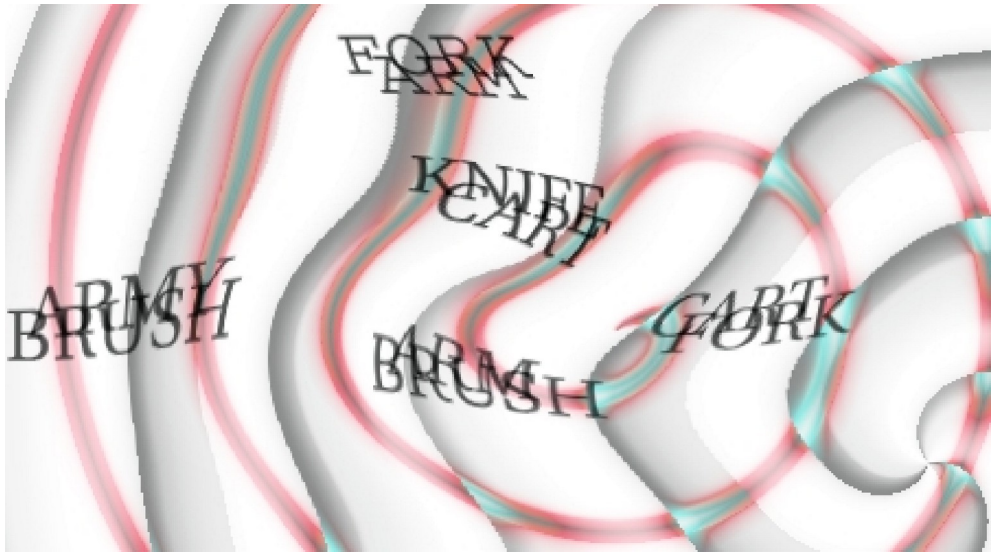
- Original motivation?
  - Automated bots stuffed ballot box in vote for best CS grad school
  - SJSU vs Stanford? No, it was MIT vs CMU
- Free email services — spammers like to use bots to sign up for 1000s of email accounts
  - CAPTCHA employed so only humans get accounts
- Sites that do not want to be automatically indexed by search engines
  - CAPTCHA would force human intervention

CAPTCHA: Rules of the Game

- Easy for most humans to pass
- Difficult or impossible for machines to pass
  - **Even with access to CAPTCHA software**
- From Trudy's perspective, the only unknown is a random number
  - Similar to Kerckhoffs' Principle
- Good to have different CAPTCHAs in case someone cannot pass one type
  - E.g., blind person could not pass visual CAPTCHA

Do CAPTCHAs Exist?

- Test: Find 2 words in the following



- Easy for most humans
- A (difficult?) OCR problem for computer
  - OCR — Optical Character Recognition

CAPTCHAs

- Current types of CAPTCHAs
  - Visual — like previous example
  - Audio — distorted words or music
- No text-based CAPTCHAs
  - Maybe this is impossible…

CAPTCHA's and AI

- OCR is a challenging AI problem
  - Hardest part is the **segmentation problem**
  - Humans good at solving this problem
- Distorted sound makes good CAPTCHA
  - Humans also good at solving this
- Hackers who break CAPTCHA have solved a hard AI problem (such as OCR)
  - So, putting hacker's effort to good use!
- Other ways to defeat CAPTCHAs???

Alternatives

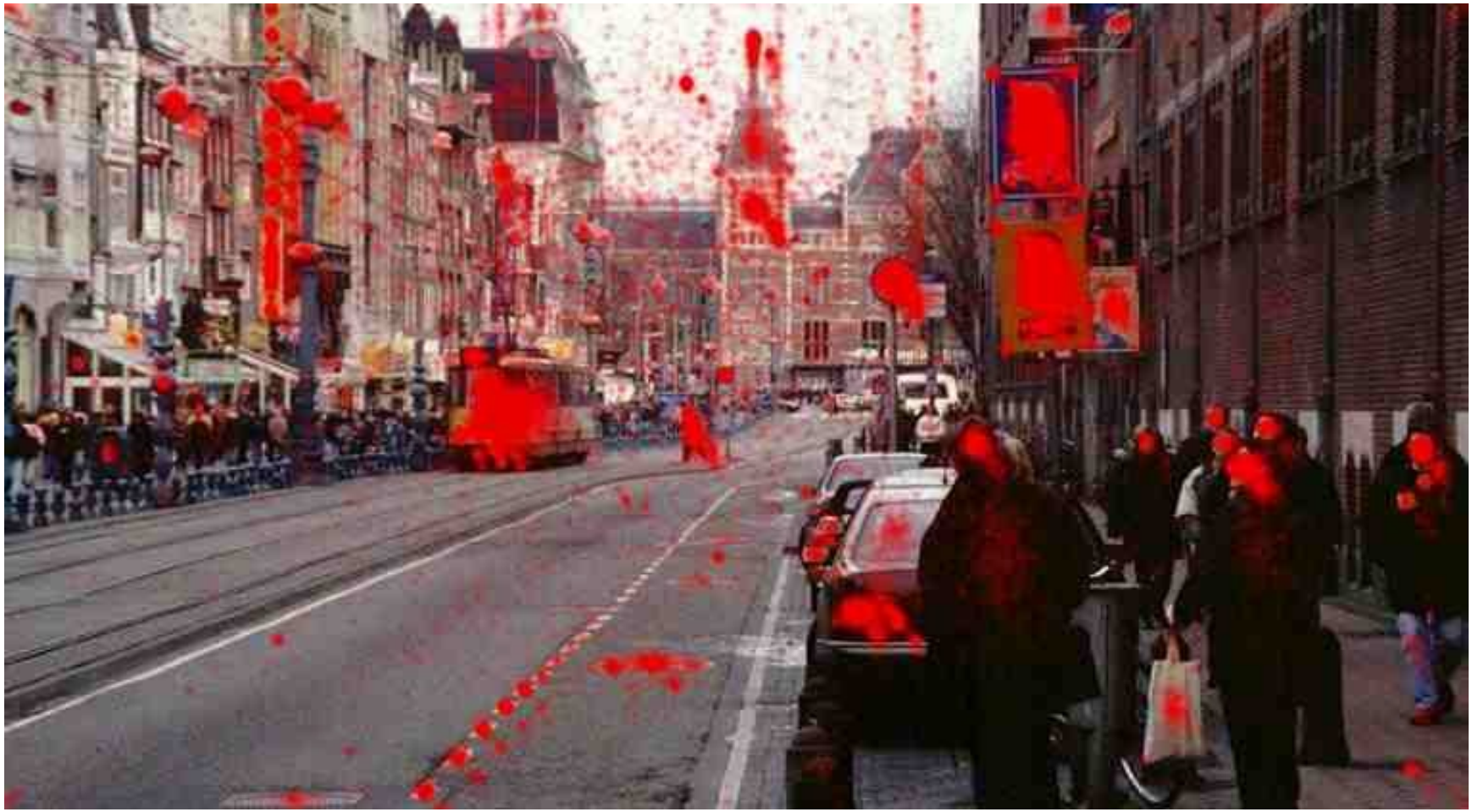
Passfaces

- Click here if you are doing the Passfaces demo for the first time

Passclicks

http://labs.mininova.org/passclicks/

But where do people click



Certificate based

- Public Key Infrastructure
- X.509 certificates
- Open standard
- Can be used in strong Authentication
- Complex for end user
- High cost
- Used for server side authentication
- Wide support

Smart cards

- Not many successful implementations
  - Card reader
  - Logistics
  - Expensive
- Standardisation poor
- On card generated X509 key pair

USB Tokens

- Smartcard with reader

SecureID

- One time pad
- Pin code
- Easy to integrate
- Clock sync

Yubikey

- External USB token
- RFID version
- Touch to generate
  one time password
- (HOTP/TOTP)
- Y2F FIDO Webauthn

- Similar from
  other brands

A-select

- Dutch Initiative
- SURFnet
- No open source
- Many platform2
- Harde authenticatie met Niegefoon en Niegebach
- DiGID

Shibboleth

- Sheveningen
- Lollapalooza
-
- Internet 2 middle ware initiative
- Good architecture
- Focus on privacy

Shibboleth

What is Shibboleth?

- Internet2/MACE project (open source)
- “inter institutional” authorization for web resources
- Authorization with privacy
- User data remains local
- More control to user and home organization
- More control for publishers

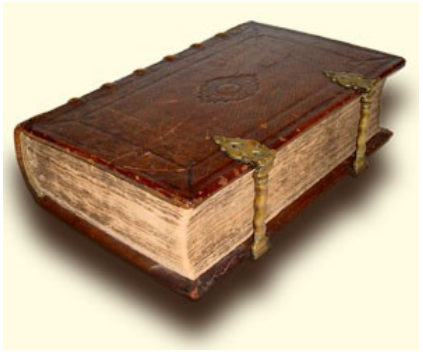
Crossing the Jordan


JORDAN RIVER

- Pronounciation password
- War between Ephraimites and Gileadites
- Bible: Judges 12:1-15
- 42.000 were killed

א.א.א. אגרונומים
שבולת
שרותי הדברה בע"מ

Old testament Bible



Then said they unto him, Say now Shibboleth: and he said Sibboleth: for he could not frame to pronounce it right. Then they took him, and slew him at the passages of Jordan: and there fell at that time of the Ephraimites forty and two thousand.

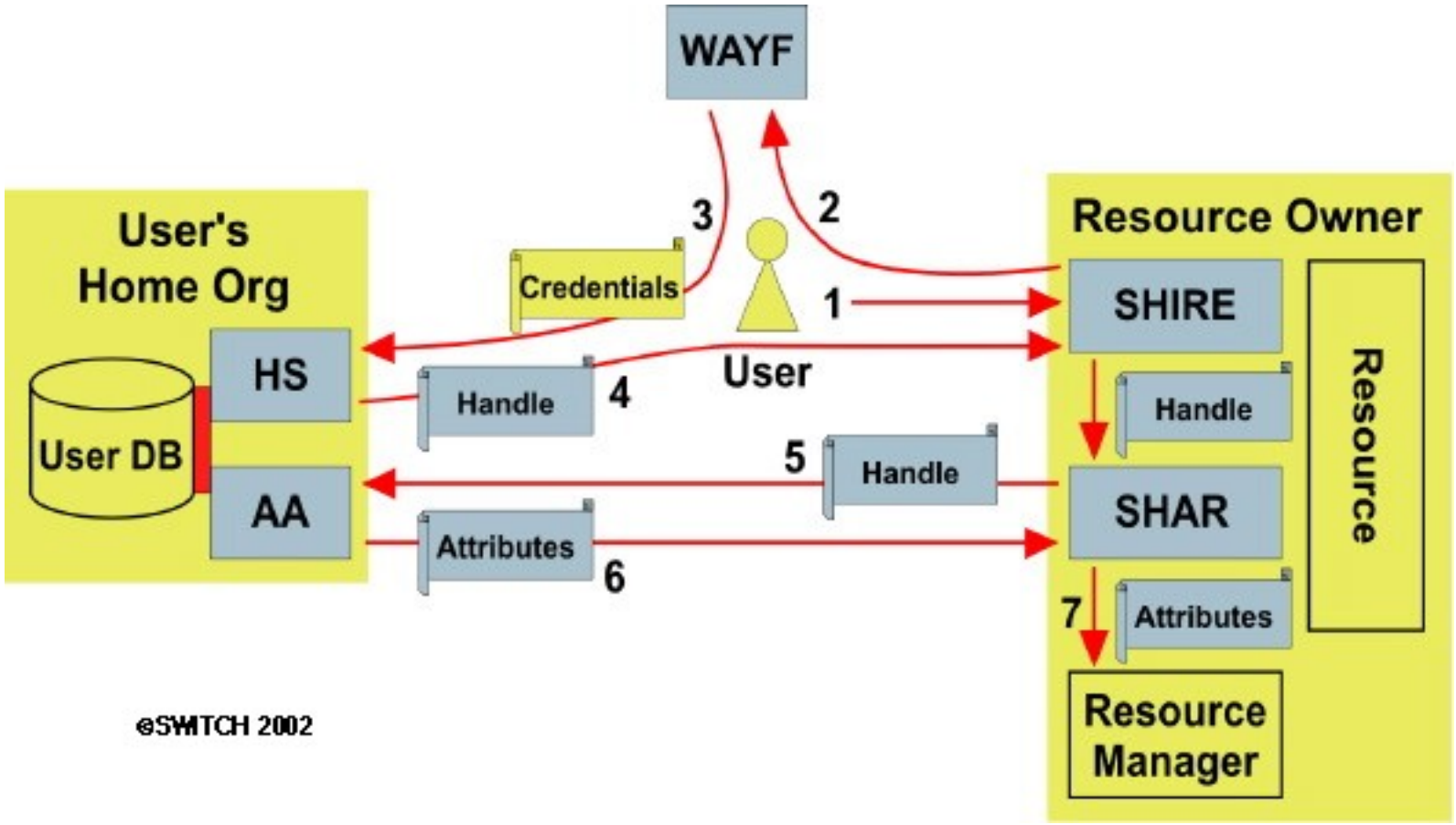
Shibboleth terminologie

Onderdelen:

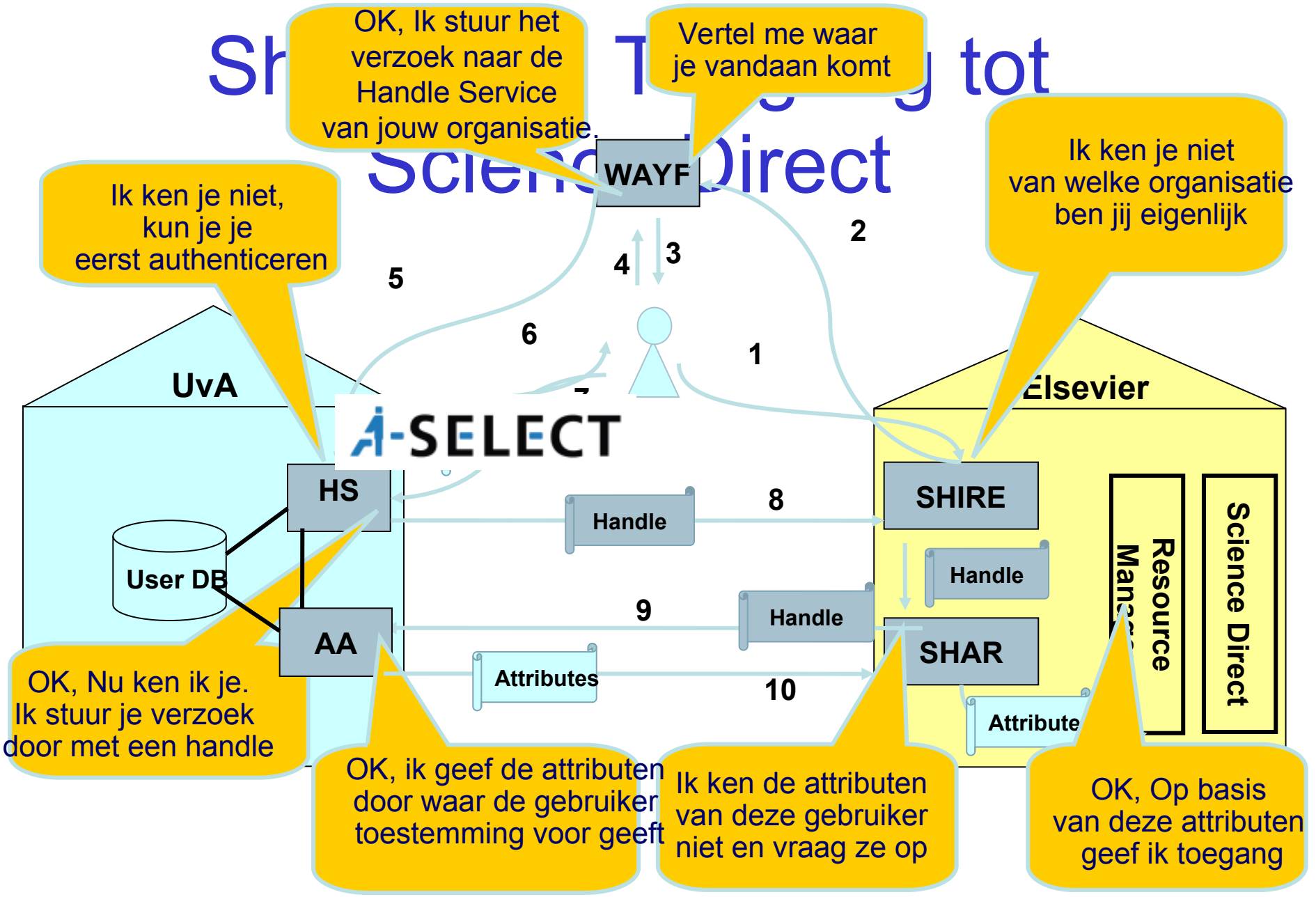
1. *Shibboleth Indexical Reference Establisher* (**SHIRE**).
2. *Handle Service* (**HS**)
3. *Where Are You From* (**WAYF**)
4. *Authentication System* (**AS**)
5. *Shibboleth Attribute Requestor* (**SHAR**)
6. *Resource Manager* (**RM**)

1. Security Assertion Markup Language (**SAML**)
2. *Attribute Release Policies* (**ARP**).
3. *Attribute Acceptance Policies* (**AAP**)

Shibboleth Architecture



Sh… T…g tot Science Direct

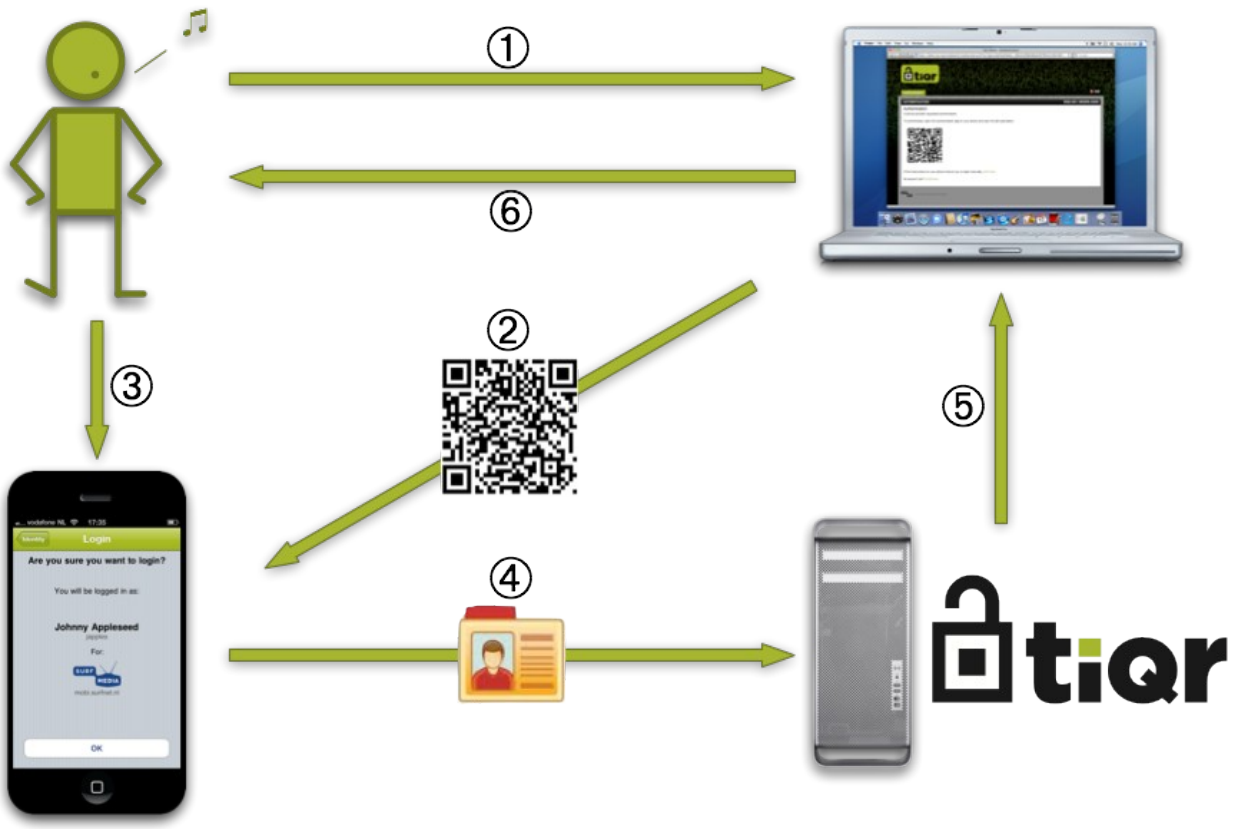


Demo

- Thanks to switch AAI
- Resource is
  - kohala.switch.ch
- WAYF is
  - wayf1.switch.ch
- Identity Provider is
  - maunakea.switch.ch

- https://www.switch.ch/aai/demo/

TIQR

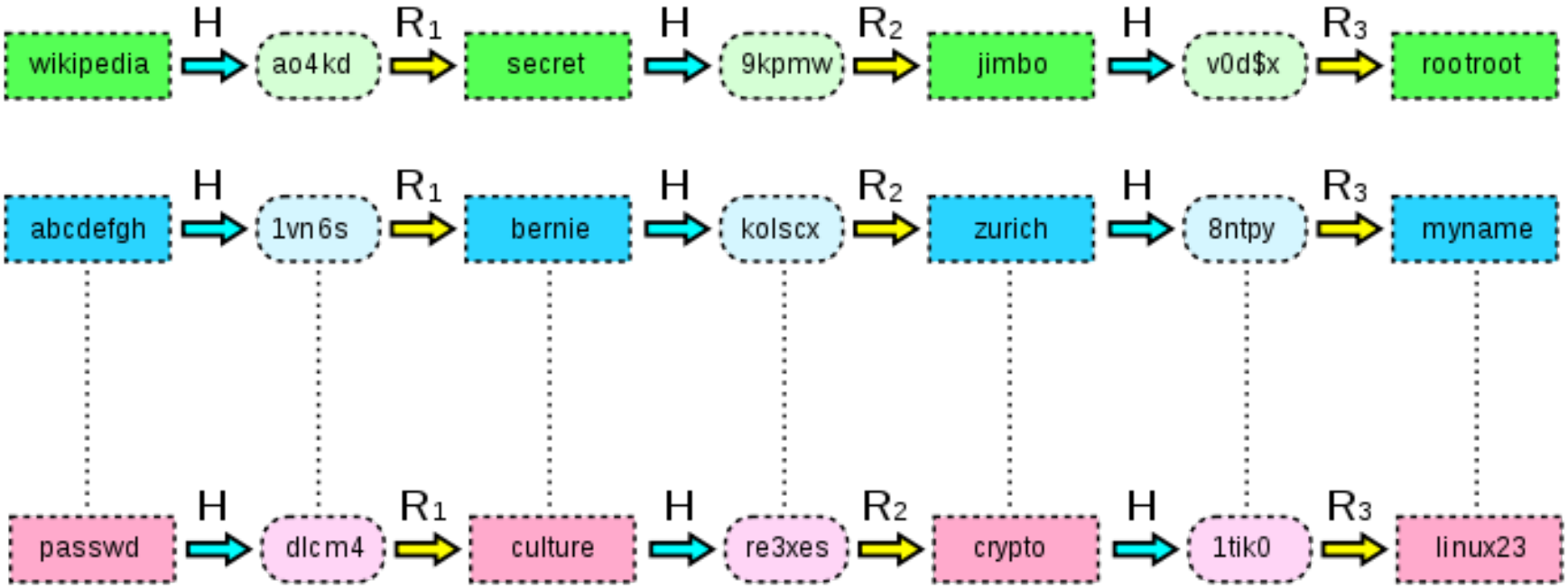

①
⑥
②
③
④
⑤
Login
Are you sure you want to login?
You will be logged in as:
Johnny Appleseed
For:
OK
tiqr

TIQR

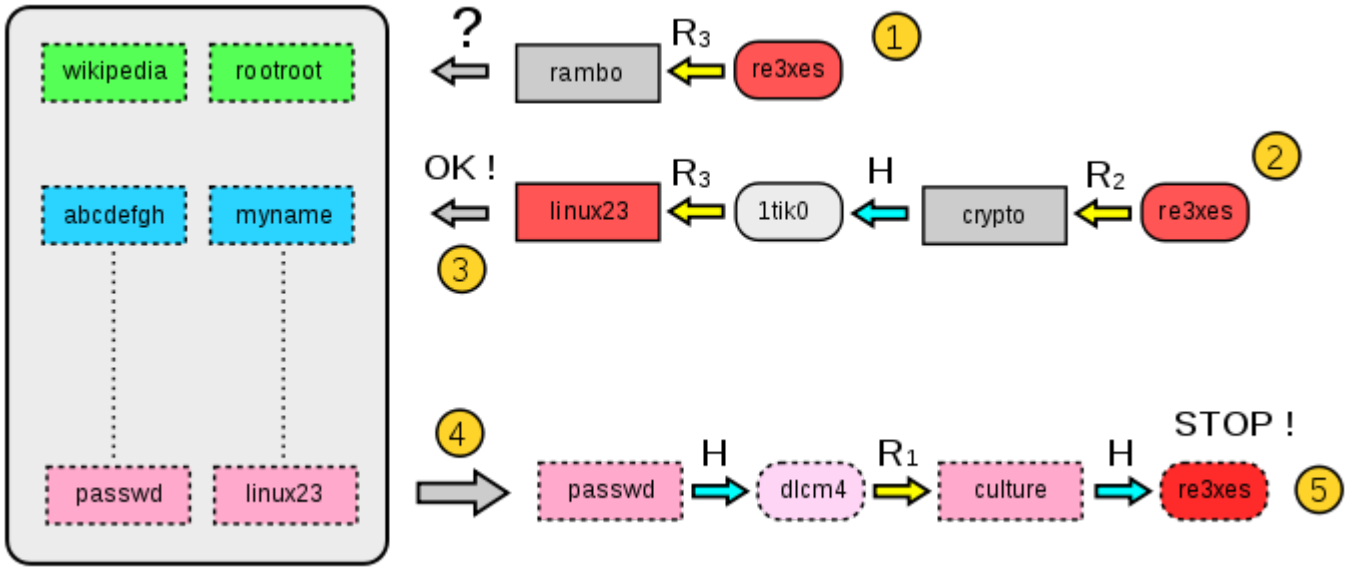
- Dutch initiative SURFn
- OAUTH
  - Initiative for Open Authentication
- OCRA
  - OATH Challenge-Response Algorithm

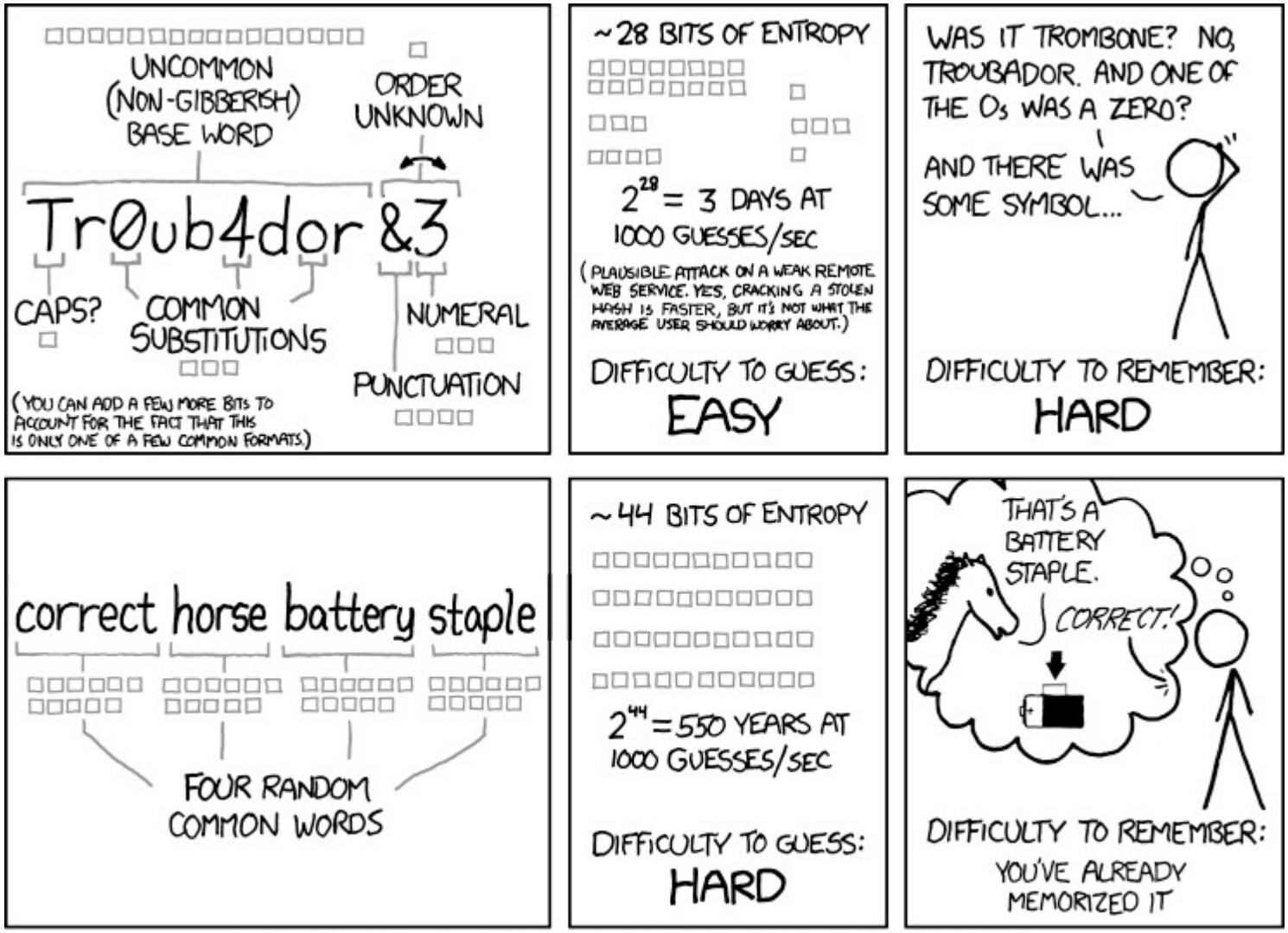
- OpenID provider (OP)
- OpenID relying party (RP)
- Microsoft Google Facebook Paypal

Rainbow tables



Cracking..


wikipedia
rootroot
abcdefgh
myname
passwd
linux23
?
rambo
R3
re3xes
1
OK !
linux23
R3
1tik0
H
crypto
R2
re3xes
2
3
4
passwd
H
dlcm4
R1
culture
H
re3xes
STOP !
5



THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Mind effectiveness