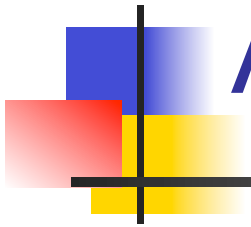
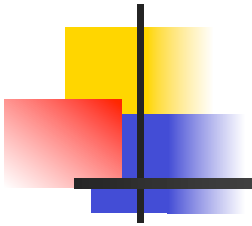


Authentication, Passwords

Robert H. Sloan



authenticate |ôˈθentiˌkāt|

verb [trans.]
prove or show (something) to be true or genuine : *they were invited to authenticate artifacts from the Italian Renaissance.*

• [intrans.] Computing (of a user or process) have one's identity verified.

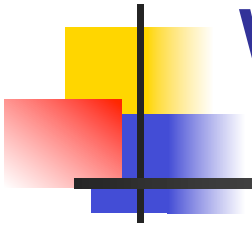
DERIVATIVES
authentication |ôˌθentiˈkā sh ən| noun
authenticator |-ˌkātər| noun
ORIGIN early 17th cent.: from medieval Latin ***authenticat-*** ***'established as valid,'*** from the verb ***authenticare***, from late Latin ***authenticus 'genuine'***.

Authentication is key

- Privacy (i.e., confidentiality) and anonymity are important for our social, business well being, but ***authentication is essential for survival***.
  - Who and what to trust and not to trust!
  - Human–Humand and Human–physical world interactions: sight, sound, smell, observation of body language, etc.

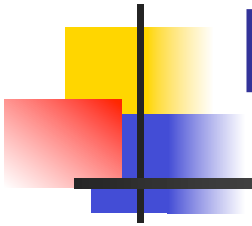
Hog dog!

- Say you want a Chicago-style hot dog
- Maybe you go to Carm's
- For sure, *authentication* is key. . . .



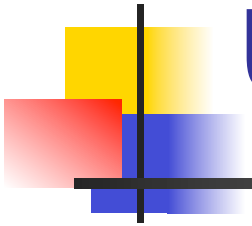
Why a hot dog?

- What's the point of the story of getting a Chicago-style hot dog?
  - Simple: Human-Human authentication is (relatively) easy
  - The hard cases are:
    - Human–Computer System across network
    - Computer System–Computer System



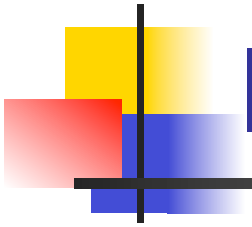
Protocols

- Passwords are most common way to authenticate human to computer system;
  - Can be considered part of (simple) protocol.
- But fancier things, or both principals devices, definitely require protocol
  - E.g., Key fob–car; IFF system
  - More on protocols later.



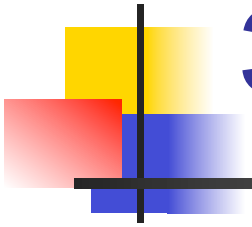
User authentication

- User authentication is absolutely crucial
- If you can impersonate someone else (be authenticated as them), you can do anything they can do
- If you can impersonate anyone (totally breaking authentication), you can do (almost) anything on the computer
- Usually hard part of taking over a computer is getting in as any one legitimate user



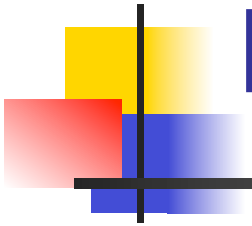
Identification, authentication

- Identification is subject (recall subject = physical person per Anderson) claiming to be a specific identity
- Authentication = process of validating that subject is who she claims to be.



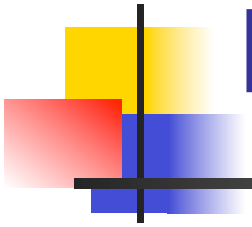
3 Ways to Authenticate

- Authentication is normally done by one or more of:
  1. What you know (typically a password)
  2. What you have (typically a chip/card of some sort)
  3. What you are (biometrics)
- All of these can fail!



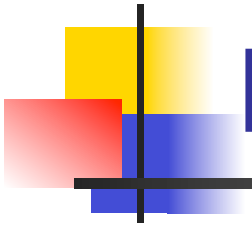
Must balance Errors

- Since authentication errors, must balance:
  - False Acceptance Rate (FAR) (fraud)
  - False Rejection Rate (FRR) (insult)
- Rule of thumb: choose setting where these two are equal (“Crossover Error Rate”) but depends on what is being authenticated.



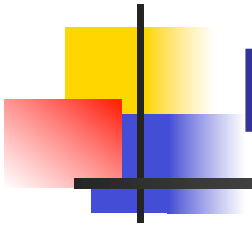
Passwords

- Most commonly used, cheapest, and clearly insecure these days
- Problem is clash of security requirements versus human capability
  - Psychologists tell us humans can’t remember infrequently used, frequently changed, many similar items and recall much harder than recognition.



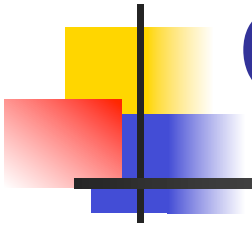
Problems with passwords

- Users may disclose them to third parties: phishing, sharing with friend
- Too hard to remember and either
  - Forgotten
  - Written down
- Maybe be entered inaccurately
- Common thread: cog. psych. issues



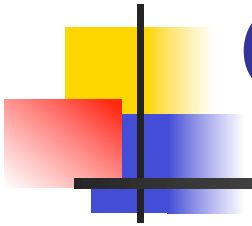
Passwords (cont.)

- Dictionaries have < 2^{18} entries.
  Common improvements
  - Restrict rate at which passwords can be retried
  - Monitor failed logins
  - Suggest recipes for difficult to guess choices (entire phrase, pass phrase initials)
  - Compare with directories and published lists
  - Requirements:



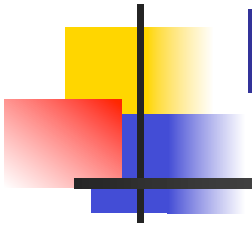
Classic Password desiderata

- Make them hard to guess: No words in dictionary, no personal info (Birth date, SSN of you or family)
- Use ≥1 digit/punctuation mark & MixED CaSe & at least 6 (8?) chars long
- Do not reuse
  - Else distinct security protocols become entwined!
- Memorize; never write them down
- Change periodically



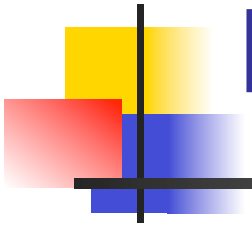
Guideline problem

- Password guidelines of previous slide are impossible to carry out
- Nobody can memorize that many distinct high-quality passwords
  - Typical person who does a lot online has 50–150 web accounts. (I have about 130.)
- I know Turing Award winners in crypto/ security who do not follow these guidelines!
- Pass*phrases* may be help some



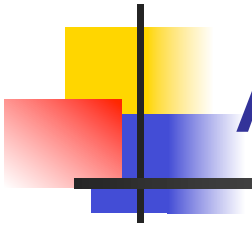
Inside an organization

- Want an aggressive enough password policy to ward off dictionary attacks (try all from list of common passwords)
- Key question is “Can you convince your users not to reuse their passwords elsewhere?”
- Helps if you can give them Single Sign-On (SSO)



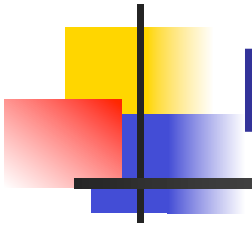
Password attacks

- Dictionary/Brute Force attacks: Hence length & character diversity requirements
  - And retry counters, but must balance with difficulty people have entering passwords
- Eavesdropping attacks (including “shoulder surfing”): Be careful when entering in person; design systems never to transmit passwords in clear over LAN
- Bogus machines/Spoofing: Need a *trusted path*



And the attack of the moment

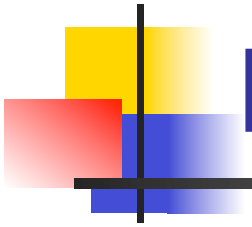
- Phishing: Trying to get you to enter your user name and password somewhere you shouldn’t
- Especially though not only: online banking
- Some suggest never getting to bank site by clicking link in email.



But

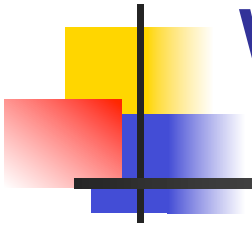
- Both banks I have some electronic dealings with regularly send me emails with a link I might want to click.
- Worst: from Everbank for a survey about their customer service: Link's text is: Please teak our short online survey today; URL is

http://www.surveymk.com/s.aspx?sm=8JmLvUhJxuXsj4Gi2wpL8w_3d_3d&c=19697%7CBANKING%5CCReinesc



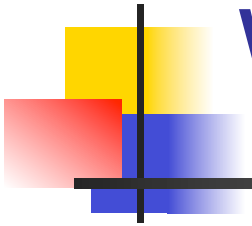
Passwords: Quis costodiet?

- Confidentiality of password database: Sure don't want it easily read!
- Don't even want to facilitate brute force attacks against other computer stored value that is function of the password
- Should generate a random nonce ("salt") and store a hash of password with the nonce.



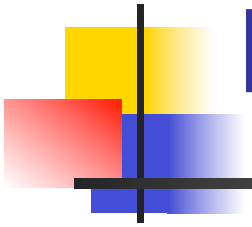
What you have

- Keys
- Cards/Chips
  - Time-generated number
  - Dumb cards: Returning same thing every time
  - Smarter Cards: Challenge and Response
- Computer itslef



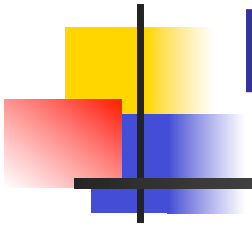
What you have attacks

- Stealing or finding
- Copying
- “Side channel”:
  - Measure power consumption of smart card (it takes more power to read bit=1 than bit=0 of secret key because ultimately something electronics)
  - Or timing, radiation, etc.



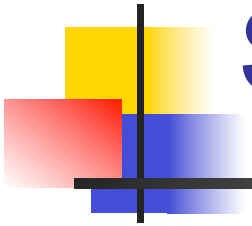
Biometrics

- Most expensive to maintain
- Inherently imperfect even with perfect users
- Main types:
  - Fingerpring/palm scan (but gelatin molds)
  - Hand geometry
  - Retina/iris scan (very high accuracy)



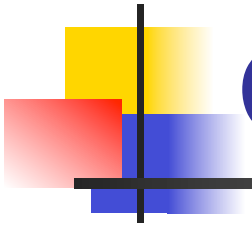
Biometric techniques (cont)

- Voice print
  - can be distorted by colds, defeated by recordings
- Keyboard dynamics
  - Can record and playback



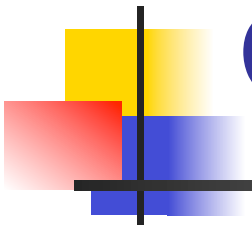
Social engineering

- A whole universe of clever attacks



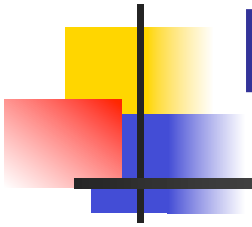
CAPTCHA's

- Authenticating merely one's non-botness.



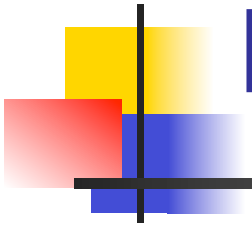
Coda: Kerberos

- Computer network authentication protocol, developed at MIT, today distributed as free software by MIT
  - Named for monstrous 3-headed dog guarding Hades
- Classified as a monition by US and therefore illegal to export until crypto policy change around 2000 in light of *Bernstein v. U.S.*
- Used in Windows 2000 onward; Mac OS X



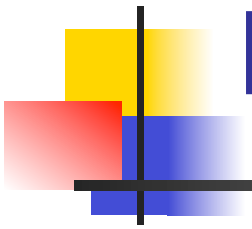
Kerberos Protocol

- Based on Needham-Schroeder Protocol we'll examine later (but bugfree)
- Trusted 3rd party, **Key Distribution Center (KDC)**, has 2 logically separate entities:
  - **Authentication Server (AS)**, to which users log on
  - **Ticket Granting Server (TGS)** gives tickets allowing access to resources (e.g., files)



Protocol itself (idea)

1. Alice logs onto AS using password, and gets session key/ticket K_{AS} to talk with TGS
2. To get service B, Alice contacts TGS, using K_{AS} to prove she's Alice.
3. If Alice is entitled to B, TGS sends Alice key with time stamp and lifetime Alice uses to authenticate with resource B.



Kerberos Weaknesses

- Requires clock synchronization; complex deliberate attack could even attack the clocks
- Single point of failure: When the Kerberos server is down, nobody can log in.