Authentication

Fall 2024

R. Sekar

History of Password Authentication

Identity and Authentication

- Access rights are granted on the basis of identity (principal)
- *Authentication* is to ensure that the principal is who it claims to be. It covers:
  - User Authentication
    - Main focus in this lecture
    - Primary problem within a single administrative domain where “the system” is trusted, but users are not
  - Authentication between systems
    - Primarily in the context of networked system, i.e., multiple domains with limited trust between them

Evolution of Password Schemes

- Early systems (1960-) stored plaintext passwords
  - Frustrated by hackers that were able to get to this file
- UNIX (1970s): store only one-way hashes of passwords
  - UNIX originally used DES, then shifted to MD5
- Use of salt to thwart offline attacks
  - a different random value used as input for hashing for each user
  - salt stored together with hashed password

Confidentiality of stored passwords

- Difficult to protect stored passwords
  - Accidental disclosures (temporary copies left behind, accidental misconfiguration of file permissions)
  - Motivated attacks on a high-value target
  - Illicit copies made by system staff
  - Stealing from backup tapes
- Solution
  - Don’t store plaintext passwords
  - Original proposal: store $DES^{25}_{\text{Password}}(0)$
  - Subsequently, use hashes (MD5crypt, SHA-512crypt)
  - For authentication, apply same process to user-supplied password, compare with stored value (in /etc/shadow)

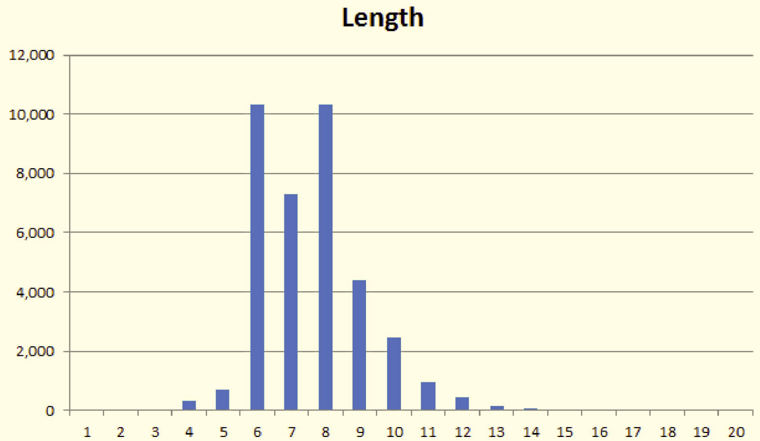
Password weaknesses [Morris, Thompson 79]

- In a collection of 3,289 passwords:
  - 15 were a single ASCII character
  - 72 were strings of two ASCII characters
  - 464 were strings of three ASCII characters
  - 477 were strings of four alphanumerics
  - 706 were five letters, all upper-case or all lower-case
  - 605 were six letters, all lower-case
  - 492 in various common dictionaries
- 86% of the 3,289 passwords were thus easy to crack
  - Cracked in seconds in some cases, and 100 hours in the best case — on computers of the 70s.

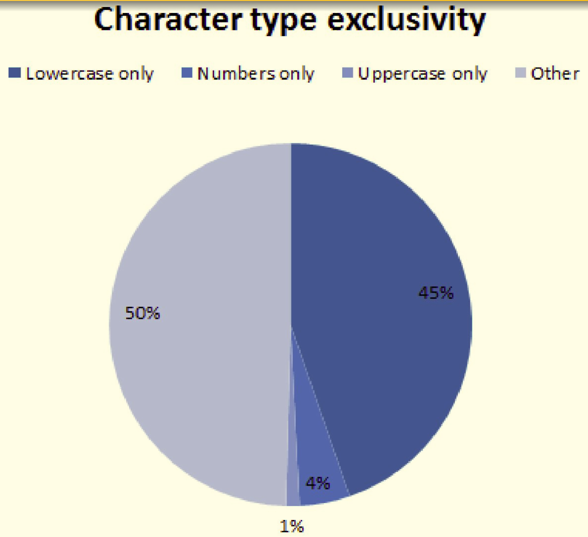
Password weaknesses [www.troyhunt.com]

Use of weak passwords is largely unchanged

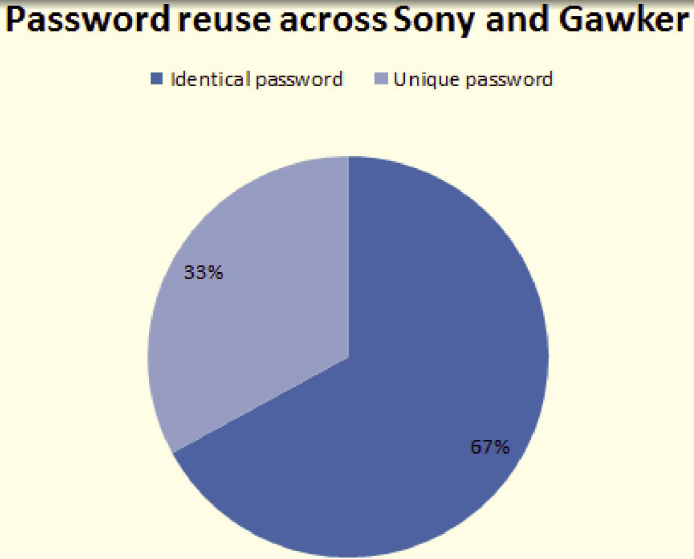
- There are almost no passwords of length < 4



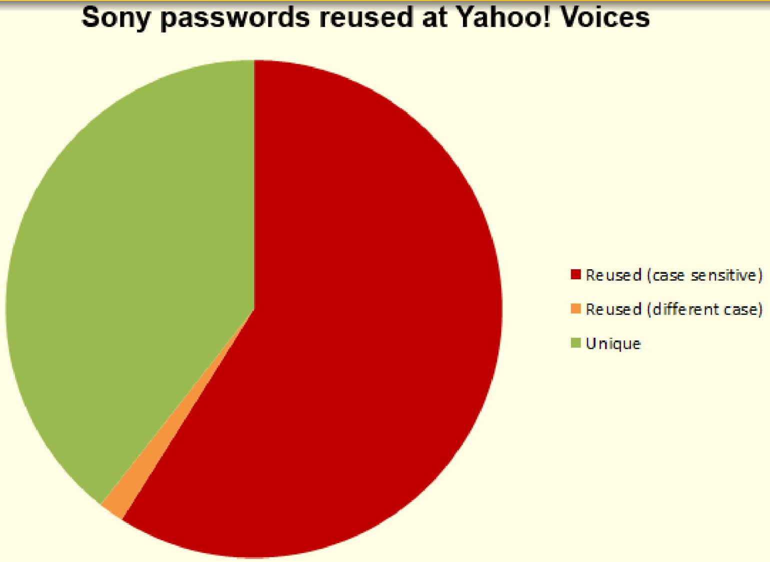
Password Weakness [www.troyhunt.com]



Password Weakness [www.troyhunt.com]

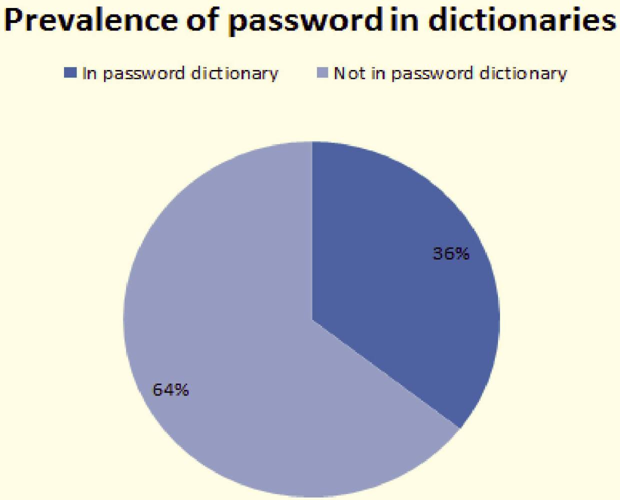


Password Weakness [www.troyhunt.com]



Password Weakness [www.troyhunt.com]

- Easy-to-remember passwords rely on patterns or algorithms
  - that can be used to generate a candidate list
  - Dictionary can also be built from passwords stolen from other sites



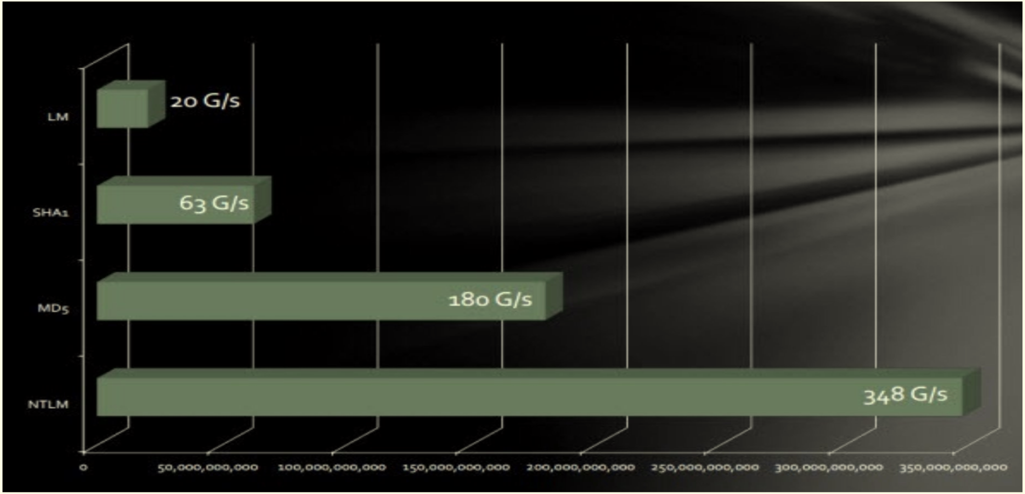
Attacks on Passwords

Categories of Attacks on Passwords

- Offline attacks: attacker has access to hashed passwords
  - Can make an unbounded number of attempts at guessing the password
    - guess, hash, compare with the hashed password
  - Brute-force attack
    - Guess password, hash, compare
  - Dictionary attack
    - Use an intelligent algorithm to enumerate passwords
    - In early days, this meant English dictionary or phone books
- Online attacks: no access to hashed passwords, so each attack attempt requires entering the password at the password dialog
  - Systems limit number of attempts, so online attacks need to succeed within a few attempts.

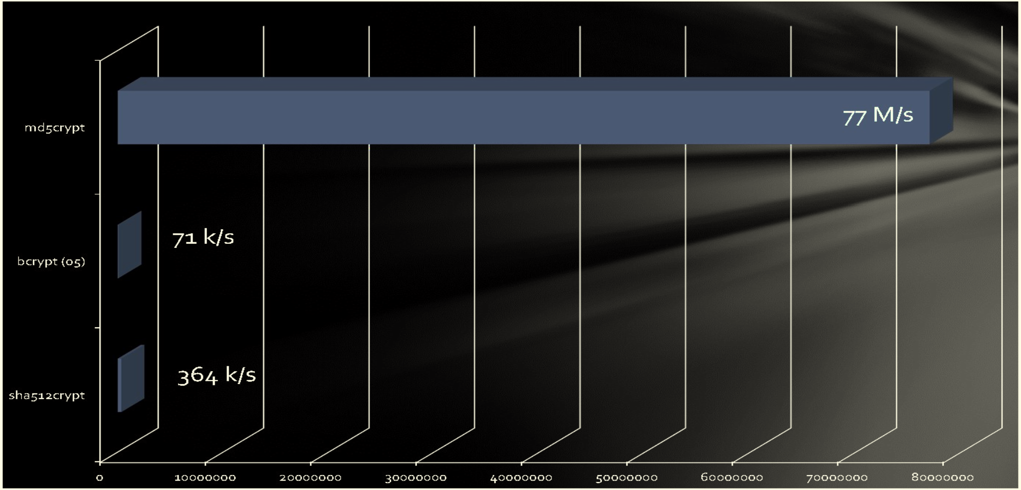
Password weaknesses [Gosney 12]

- Brute-force, dictionary attacks greatly speeded by GPUs



Password weaknesses [Gosney 12]

- Even GPUs are not too fast for some hash algorithms



Defending against Offline attacks

- Slow down offline attacks
  - Make hash algorithm slower
  - Make attacker repeat work for every user (“salt”)
    - Each user assigned a random salt value (which is stored in the password file)
    - Original proposal: $\mathrm{DES}^{25}_{\mathrm{Password}||\mathrm{salt}}(0)$
    - Eliminates attacks that hash once, compare against passwords of all users
- Protect password file
  - /etc/passwd is world-readable, so easy to steal
  - Modern UNIX versions separate password hashes (and salt) into an /etc/shadow that is readable only by root

Online Attacks

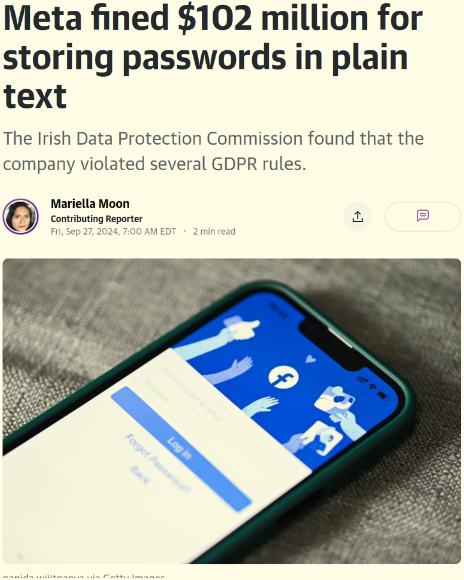
- Guessing is typically unsuccessful except for the most easily guessed passwords.
  - Delays: remove login prompt after 3 failed attempts.
    - Increase delay (e.g., double) after additional failures.

    Lock outs: prevent user from logging in after *N* failures. CAPTCHAs: make user solve CAPTCHA after *N* failures.
- Password stealing is the most viable approach for succeeding in online attacks.
  - Phishing (fake password dialogs)
  - “Password dumps” — passwords stolen through cyber attacks and revealed afterwards
  - Network sniffers.
  - Keyloggers and other malware.
  - Password reset.

RockYou2024: Unpacking the Largest Password Leak in History

Jasdev Dhaliwal | JUL 08, 2024 | 3 MIN READ

This Fourth of July brought fireworks in the form of a digital security breach, one that has been recorded as the most significant password leak in history. Dubbed RockYou2024, this colossal data dump was unveiled by a user named "ObamaCare" on a prominent hacking forum, revealing a staggering 9.9 billion unique passwords in plain text.



Meta fined $102 million for storing passwords in plain text

The Irish Data Protection Commission found that the company violated several GDPR rules.

Mariella Moon
Contributing Reporter
Fri, Sep 27, 2024, 7:00 AM EDT · 2 min read

panida wiiitpanya via Getty Images

Password Theft and Trusted Path

- How to make sure that your password is not stolen when it is used
  - Key challenge today due to spyware, spoofing, phishing, etc.
- Trusted path: a secure way for a user to communicate with the subsystem performing user authentication
  - Ctrl-Alt-Del on Windows
    - Provided that the OS is not infected ...
    - And the BIOS is not infected ...
    - And the hardware is not malicious ...

Phishing and Trusted Path

- Phishing attacks typically involve tricking a user into revealing their passwords
    - Attacker sets up a web site that looks like attack target, e.g., a bank web site
    - Attacker steals the password when the user tries to log into the fake web site

Phishing Defenses

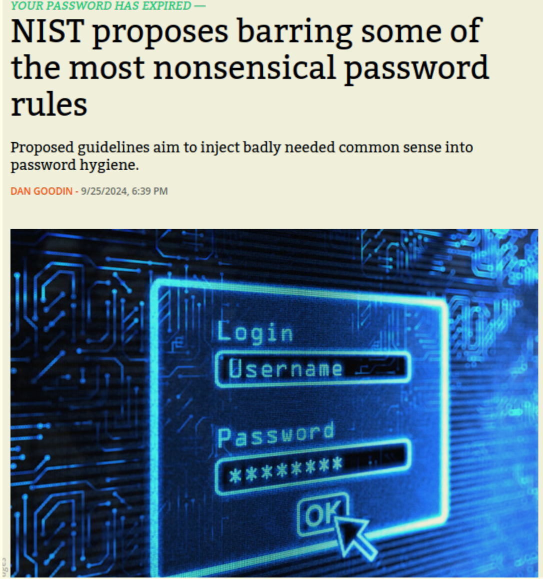
- Two-stage login with personalized prompts
  - Security skins, site-keys (personalized images)
    - Requires user vigilance
    - Phisher may say “system failure, so we can’t retrieve your image at this time”
    - Small “key space” for possible images
  - Security questions
    - Pain to use
    - Small key space
    - Answers easily guessed, especially by family/friends

Phishing Defenses

- SSL provides strong defense (completes trusted path)
  - Password managers are not fooled by typo squatters!
  - What can still go wrong?
    - Self-signed certificates — But today's browsers provide stronger warning (or silently suppress) sites that change a CA-provided certificate into a self-signed one.
    - Social engineering ("our SSL servers are down today")
    - Compromise of Certification Authorities
- Two-factor authentication

Password weaknesses: Non-solutions

- CAPTCHAs to defeat online attacks
  - Increasingly, becoming too hard for humans!
- Security questions
  - Often, answers are available on social media
- Password rules
  - A nightmare for users
  - Questionable increase in password strength
- Alternative password schemes
  - Face or picture recognition



YOUR PASSWORD HAS EXPIRED —

NIST proposes barring some of the most nonsensical password rules

Proposed guidelines aim to inject badly needed common sense into password hygiene.

DAN GOODIN - 9/25/2024, 6:39 PM

Summary of Password attacks

- Offline
  - Brute-force and dictionary attacks greatly speeded up by GPUs
  - Dictionary attacks speed up the search, especially if they are based on passwords revealed in data breaches
- Online and offline:
  - Use of weak passwords
  - Keyloggers (and formerly, network sniffers)
  - Social engineering (phishing)
  - Password reset mechanisms

Authentication Over Networks

Approach 1: Server-side authentication of plaintext passwords

- Don’t trust client computer; server performs this task
- Used by rsh/rlogin/rexec, telnet, ftp, etc.
- Bad option unless you (a) physically secure the network, and (b) trust all clients on the network
  - Otherwise, easy password compromise by network sniffers

Approach 2: Host-based authentication

- Trust client host to perform user authentication
- Used in NFS, also rsh/rlogin/rexec with `/etc/hosts.equiv`
- Not a great option today, as users often have admin privileges on client machines
  - With so much user control (and high risk of mismanagement), it is bad practice to trust these machines

Approach 3: Transmit only encrypted passwords

- Encrypt user password using a client host specific secret
  - Server uses client secret to decrypt and verify user password
  - Unfortunately, encrypted password is as good as an unencrypted one!
    - A rogue client can sniff and reuse this encrypted password to log into the server, without ever needing to decrypt it
- Need solutions against such replay attacks
  - Challenge-response protocols
  - One-time passwords (theft no longer a problem)

One-time passwords (Early solution to network sniffing)

- Start with a password P to generate a sequence of one-time passwords $O_1, \ldots, O_N$
  - Requirements: O_k should not provide any info about $O_{k+1}, O_{k+2}, \ldots, O_N$
- Solution: $O_k = H^{N-k}(P)$, where H is a secure one-way hash function
- Protocol:
  - System $\rightarrow$ User: i
  - User $\rightarrow$ System: $H^{N-i}(P)$
  - Even if user doesn't respond, use $i + 1$ as next challenge
- Note: system need not store P, just the previous OTP
  - check that $H(\text{current OTP}) = \text{prev OTP}$

Other OTPs: SecureID

- A hand-held device sold by RSA
  - Widely deployed in enterprises
  - Well-publicized hack on this system in early 2011 led to attacks on high-profile businesses
- Uses a device-specific secret to generate authentication token every minute or so
  - E.g., AES_{K_S}(Time)
  - Tamper-resistant device, so one cannot steal K_S
  - Server must know device-specific secret
- Combined with a PIN or password
  - Perhaps the first widely-deployed two-factor authentication

Challenge-response protocols: SSH

- Password based authentication
  - $S \rightarrow C : KU_S$
  - $C \rightarrow S : E_{KU_S}(K_{SES} = random()), E_{K_{SES}}(password)$
  - All subsequent communication encrypted using K_{SES}
  - Weakness: integrity of KU_S not assured. SSH asks user to confirm the key the first time a server is accessed, and saves the key for use in future accesses to same server
- Public key based authentication: replace password sending step with the following challenge-response protocol:
  - $C \rightarrow S : KU_{USER}$
  - $S \rightarrow C$: Verify presence in `.ssh/authorized_keys` in user's home directory, send challenge = $E_{KU_{USER}}(random)$
  - $C \rightarrow S$: decrypt and send the result

Challenge-response protocol: Websites

- Web sites use password authentication over https
  - $S \rightarrow C$: Public key certificate $E_{KR_{CA}}(KU_S)$
  - $C \rightarrow S$: $E_{KU_S}(K_{SES} = random())$
  - All subsequent communication encrypted using K_{SES}
- Similar to SSH password authentication
- Most protocols (e.g., ftp) can be made secure by simply carrying their traffic over https or ssh tunnels.

Password weaknesses: Solutions

- Master password
  - Generate random passwords, encrypt them using master password
  - A password manager helps, but even the low-tech approach of noting them down in an encrypted file is a great improvement.
- Public keys, e.g., SSH or PGP
  - Need tools to help, e.g., USB security keys, laptops (ssh), ...
- Two-factor authentication
  - Tokens, cards, biometrics, ...
  - Pass keys
- One-time passwords or PINs
  - Useful if a channel trusted communication channel is available, e.g., SMS or email.

Password Management Challenges

- Easy-to-remember passwords may be easy to guess
  - Dictionary attacks
- Password management
  - Dealing with multiple passwords
  - Writing passwords down (should I?)
  - Password selection rules
  - Password expiry rules

Using Master Passwords

- A master password is used to encrypt all other passwords
  - Focus on creating/remembering one strong password
    - low tech approach: all other passwords written down in a file that is manually encrypted with the master password
    - more usable approaches rely on "password managers"
    - built into common applications like ssh and browsers

Benefits of Password managers

- Allows strong passwords unique to each website
  - Generate a random password for each site
- Reduces theft due to practices such as writing them down
- Computers are not easily phished
  - Avoids password being revealed to sites that
    - look similar
    - have URLs that are misspelled or have typos
    - use http instead of https

Issues with password managers

- Bad idea on shared devices
- Stolen (or temporarily lost) devices with passwords
- False sense of security if master password can be stolen

Summary of User Authentication Approaches

- Something you know
  - A secret: text, visual, or other types of passwords
  - Issues: difficulty of guessing, ease of remembering
- Something you have
  - key, magnetic card, RFID chip, smart card, cell phone, ...
  - Issue: possibility of losing
  - Combine with a secret to minimize damage due to loss
- Something you are
  - Fingerprint, photo, voice, handwriting, ...
  - Issues: accuracy of recognition, possibility of stealing
  - Works best in a supervised setting

Biometrics

- Authenticate by recognizing some aspect of human physiology, anatomy, skill or trait
  - Physiological (fingerprint, iris, retina, face, hand geometry, DNA)
  - Behavioral (keystroke, voice/speech, ...)
- Benefits:
  - convenience
  - protection against poor choice of passwords
  - more difficult to steal, particularly in controlled (supervised) setting
- Drawbacks
  - Need for special equipment
  - Not 100% reliable (false positives and negatives)
  - User acceptance

Biometrics: Terminology, Issues

- False match or acceptance rate (FMR/FAR)
  - “fraud rate”
- False non-match/rejection rate (FNMR/FRR)
  - “insult rate”
- trade-off between the two: equal error rate
- verification (pair-wise comparison) Vs
- identification (one-to-many comparison)
  - even very small error rates get magnified for the latter, and hence become unacceptable.

Biometrics: Terminology, Issues

- Issues
  - User acceptance
  - Privacy and discrimination
  - Can’t be canceled/changed if stolen
  - Danger of physical harm to owner

Handwritten signatures

- Routinely used in transactions and contracts for centuries
- Recognition may be manual, machine-assisted or completely mechanical
- Different approaches may be warranted based on application
  - legal Vs check-out counter Vs check-clearing for small checks
- Signature tablets
  - record signature dynamics as well as the resulting image

Fingerprints

- most commonly used biometric
- Issues:
  - even low error rates can compound when doing a one-to-many match
  - manipulation: lift prints artificially and deposit where there are needed.
  - ++ mature
  - ++ as always, deterrent effect can be higher than actual effect

Iris recognition

- Benefits
  - unique for each person
  - does not wear out or is exposed to external environment
  - easy to make out from a picture.
  - many times the number of degrees of freedom as fingerprint
  - minimally influenced by genetics
  - stable through lifetime
- Gabor filters – a signal processing technique to transform an image of the iris into a 256-byte code. Two codes computed from same iris will match in 90% of the bits
  - Compare with fingerprints, where detection, classification and orientation of minutiae is hard.

Iris recognition

- Can achieve very high accuracy in controlled settings, but real-world performance not as good
- Other issues:
  - Requires camera-to-eye distance of approx. 2ft or less (intrusive)
  - Can potentially be copied

Voice Recognition

- text-dependent recognition (challenge-response)
- noise can be a problem (may need microphone held close to mouth)
- one-to-many comparisons are not very accurate
- affected by stress, cold, alcohol or other drugs, ...

Other

- Keystroke dynamics
- Hand geometry
- Retina
- DNA

Problems with Biometrics

- age of reference data (e.g., fingerprint)
- age of data (when was that fingerprint left? yesterday when the bank robbery took place, or last week when there was a legitimate visit to the bank?)
- recordings
- collusions (voluntarily provide bad writing samples or photos)
- birthday problem
- combining biometrics does not necessarily help: it may reduce false accepts, but at the cost of increased false rejects (or vice-versa)
- may not work for all users (“goats”)
- objections based on social and religious concerns

Visual Passwords

- Leverage highly evolved visual perception
  - Pictures seem so much easier to remember than the details in an arbitrary text password
- Several schemes
  - Passpoints: select points on an image
  - Select images from an array
    - Passfaces: leverage human capacity to recall faces
    - Random art

Issues with Graphical Passwords

- Many of the basic attack techniques continue to work
  - Dictionary attacks, guessing, social engineering, ...
  - Easy-to-remember passwords may also be easily guessable
- And there are several new ones
  - Shoulder-surfing
  - Deceptively low entropy
    - Studies show that users tend to have favorites, e.g., pretty faces from one's own race (for passfaces)
  - Memorability has not been conclusively demonstrated

Summary of User Authentication

- Purpose: bind physical-world entities with cyber-world entities
- Means: Present “credentials”
  - Secret
    - passwords
  - Possession
    - Key-card
    - Biometrics
- Attacks: theft, guessing attacks, ...
- Defenses
  - Multi-factor authentication
  - Password managers