Identity and Authentication

- Access rights granted on the basis of identity of the entity performing access (principal)
- Authentication mechanisms used to establish that a principal is who he/she claims to be
  - Alternatively, one may be interested in proving that they have certain rights
- Covers
  - User authentication
    - Main focus in the next few pages
    - Primary problem within 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 hashes of passwords
  - Hash: one way function that is infeasible to revert
  - Originally used DES, subsequently shifted to MD5
    - MD5 now considered weak for this purpose, use SHA-512 or bcrypt
  - Use of salt to thwart offline dictionary attacks
    - Salt = different random value for each user, used in hashing; stored together with hashed password

Issues in Password-based Authentication

- 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: $DES^{25}_{Password}(0)$
      - More recently, use hashes (MD5crypt, SHA-512crypt)
    - Apply same process to user-supplied password, compare with stored value (in /etc/passwd)

Categories of Attacks on Passwords

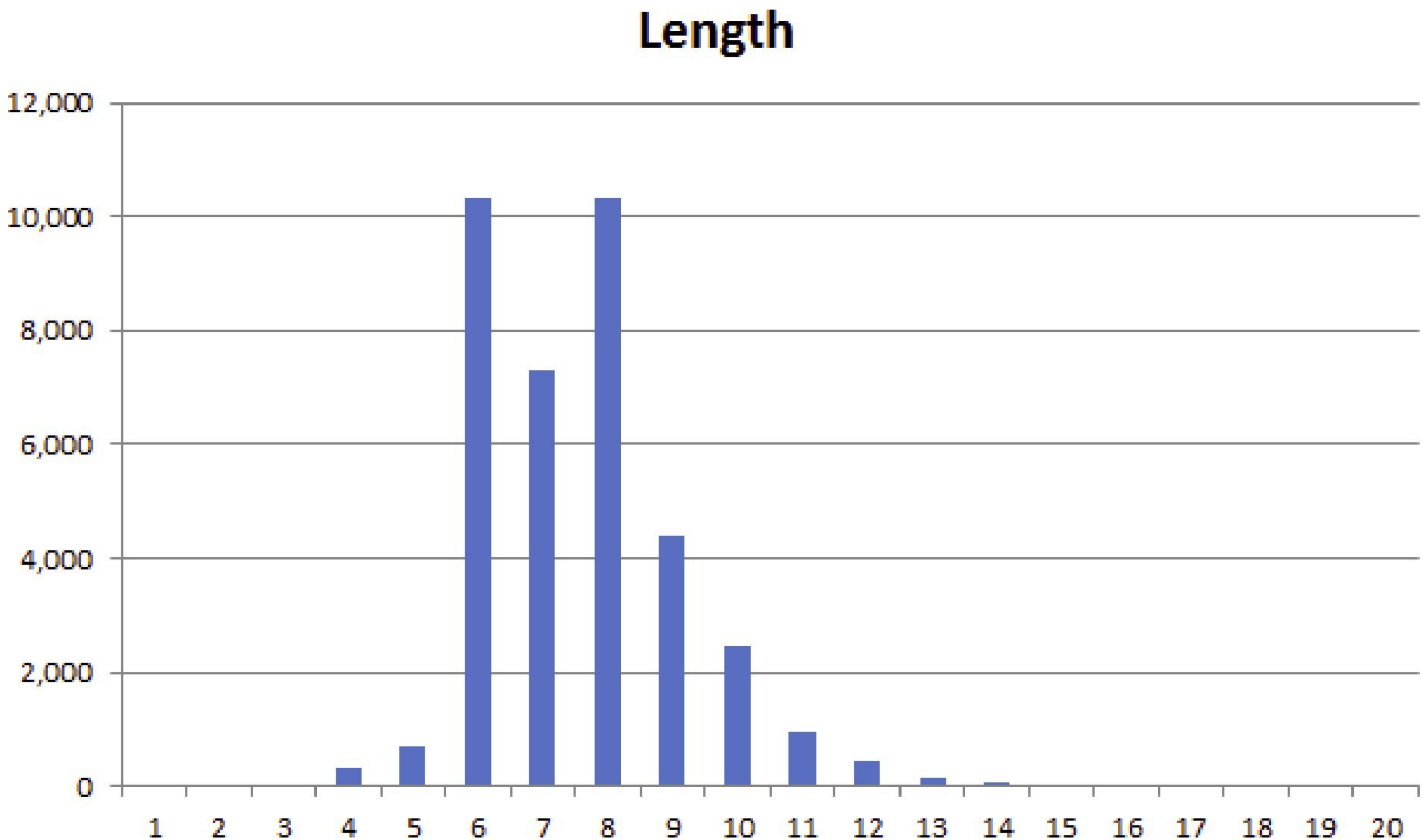
- Offline attacks: attacker has access to hashed passwords
- Online attacks: no access to hashed passwords
- Online attacks need to succeed with very few attempts, while these attacks rely on the ability to cycle through millions or billions of guesses:
  - Brute-force attack
    - Guess password, hash, compare
  - Dictionary attack
    - Use an intelligent algorithm to enumerate passwords
    - In early days, this meant English dictionary or phone books

Password weaknesses [Morris, Thompson 79]

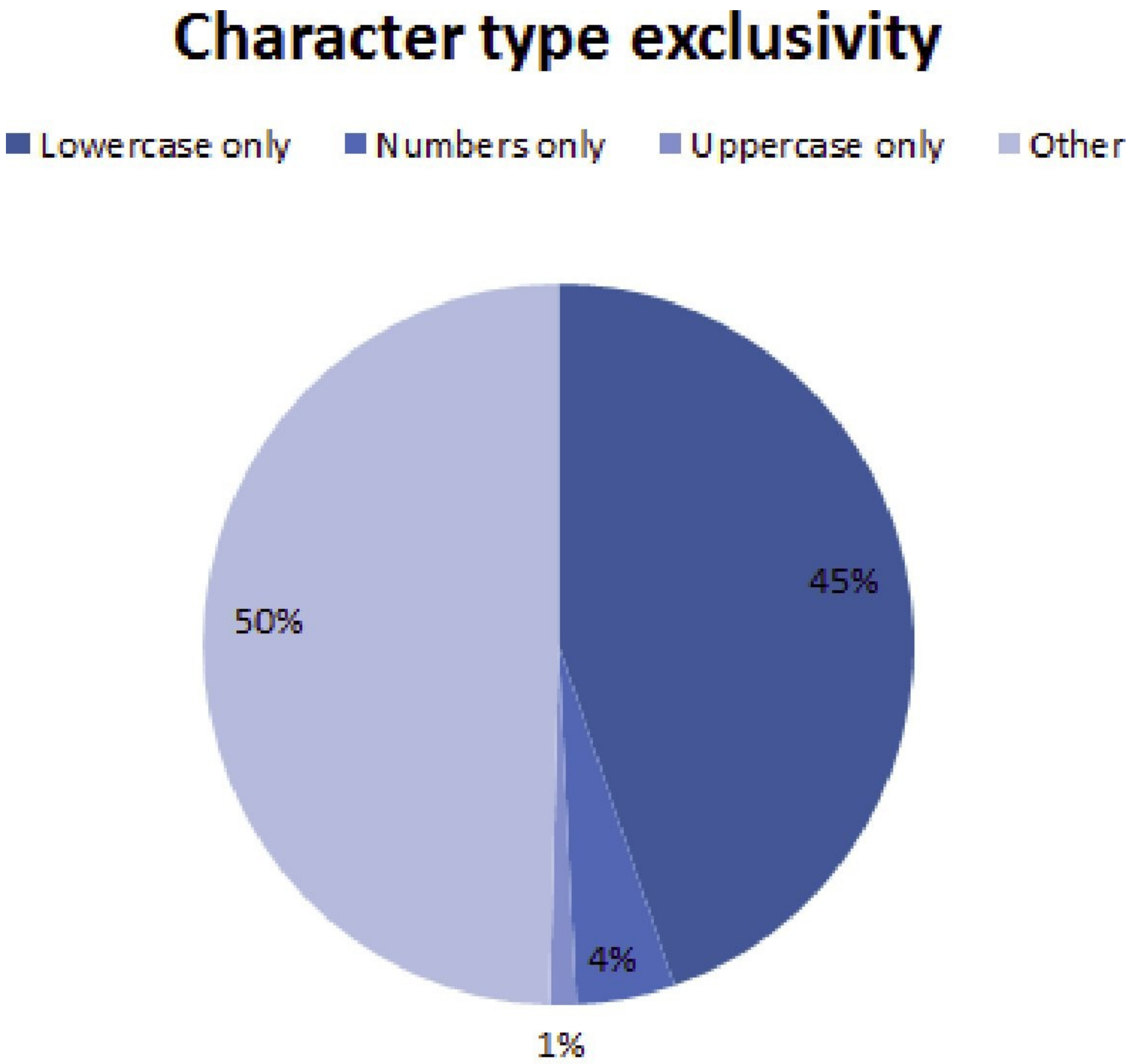
- In a collection of 3,289 passwords:
  - 15 were a single ASCII character
  - 72 were strings of two ASCII characters
  - 464 were strings of three ASCII characters
  - 477 were strings of four alphamerics
  - 706 were five letters, all upper-case or all lower-case
  - 605 were six letters, all lower-case
  - 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]

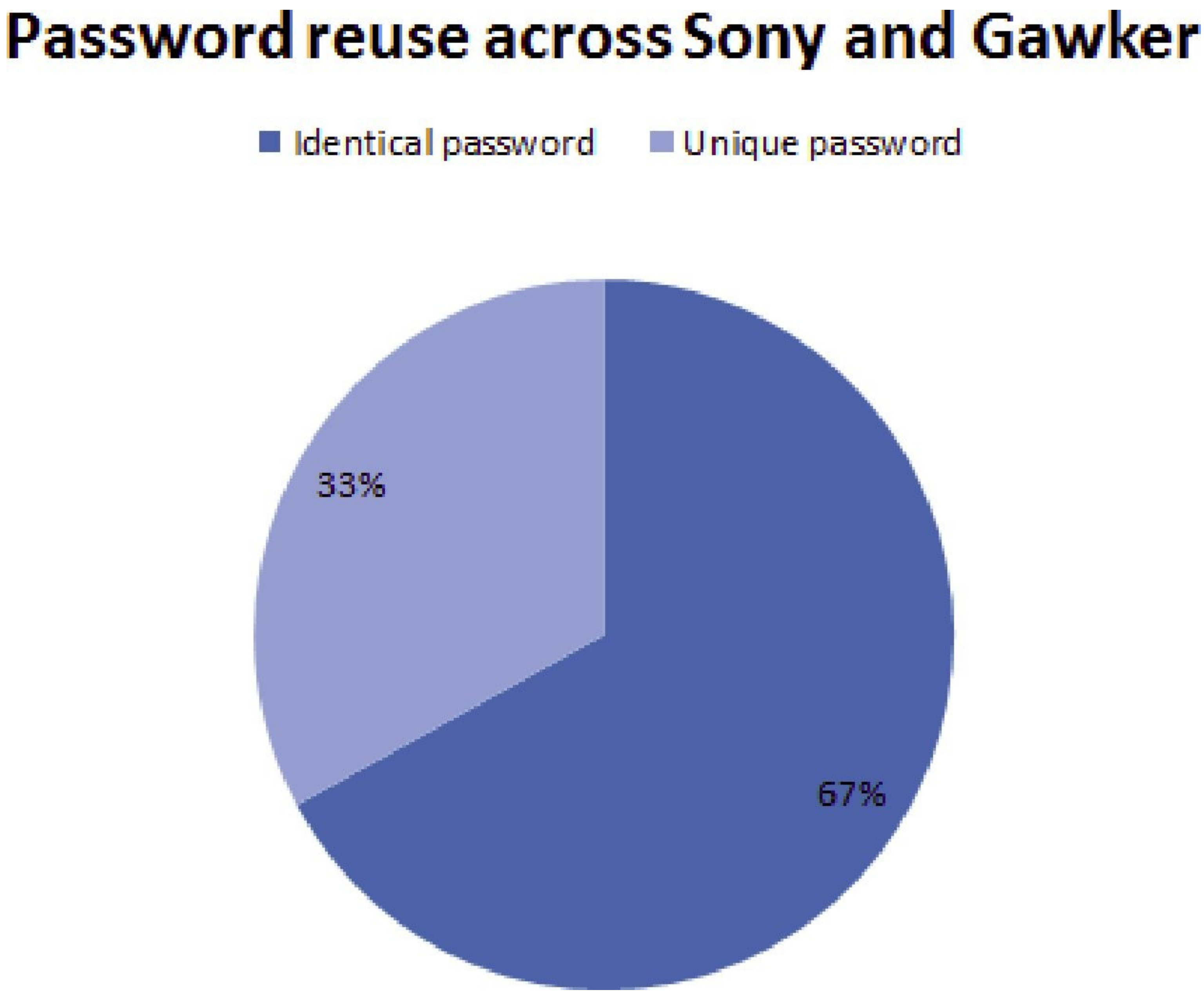
- Use of weak passwords is largely unchanged
  - OK, there are almost no passwords of length < 4



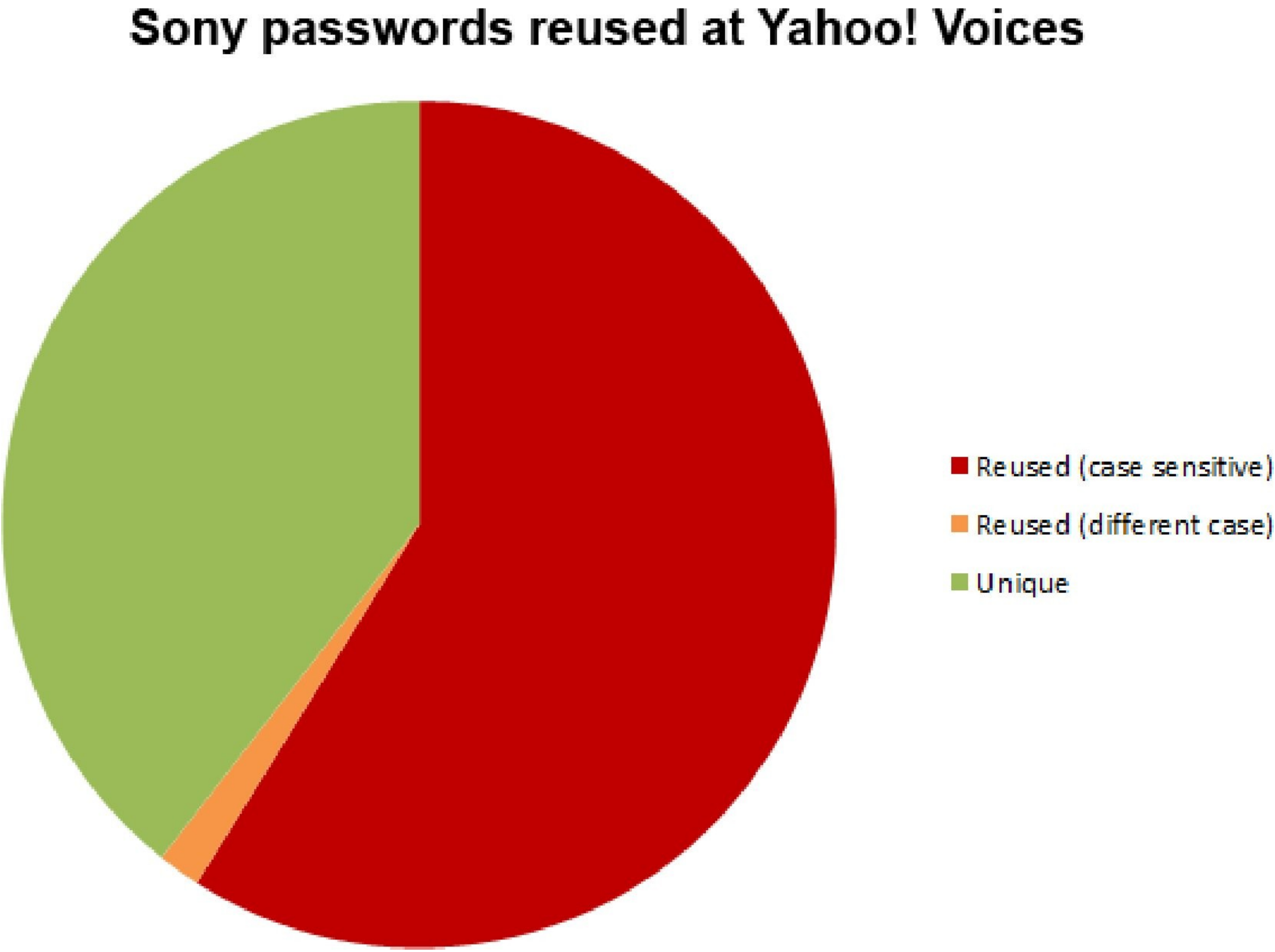
Password weaknesses [www.troyhunt.com]



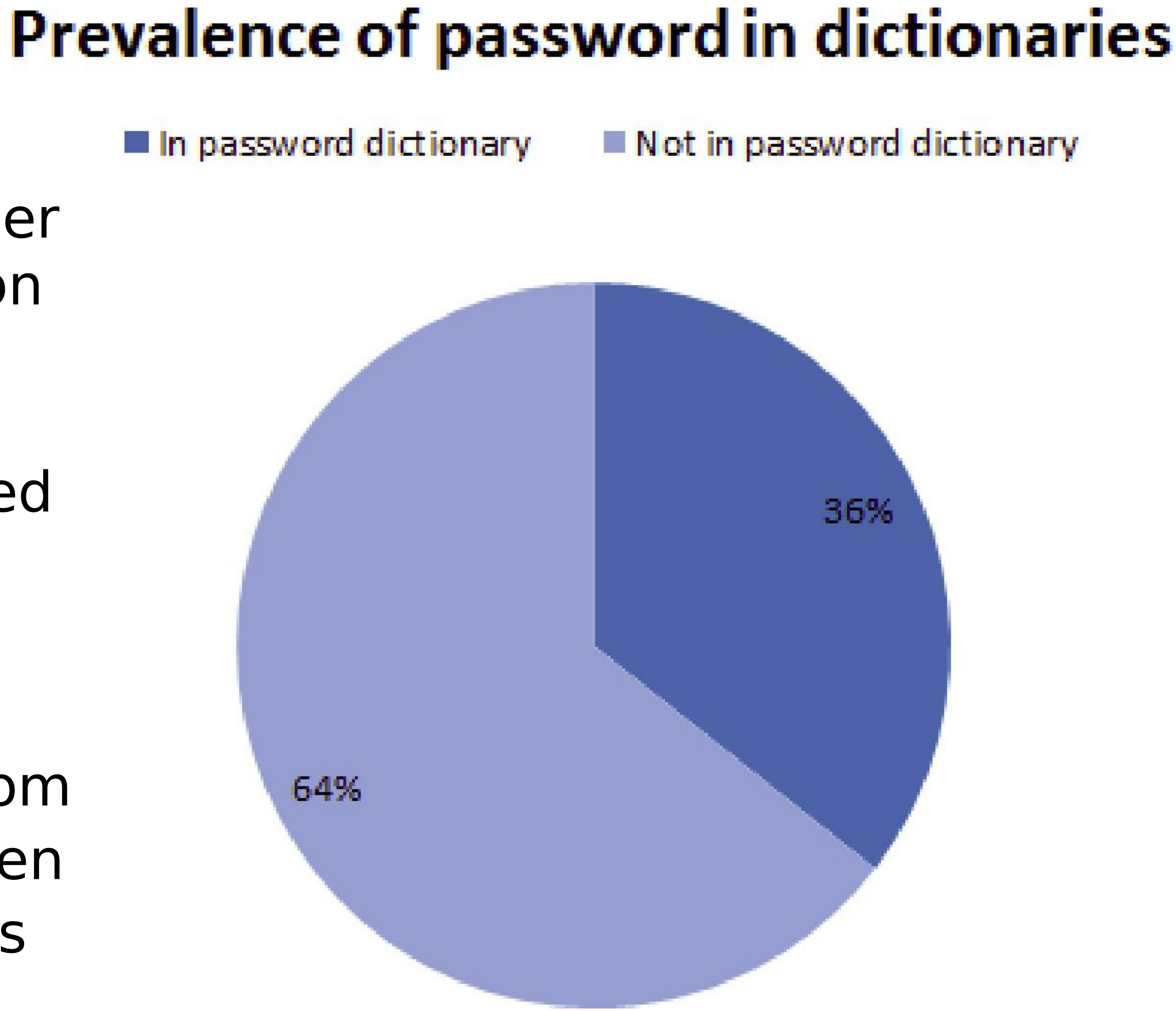
Password weaknesses [www.troyhunt.com]



Password weaknesses [www.troyhunt.com]



Password weaknesses [www.troyhunt.com]

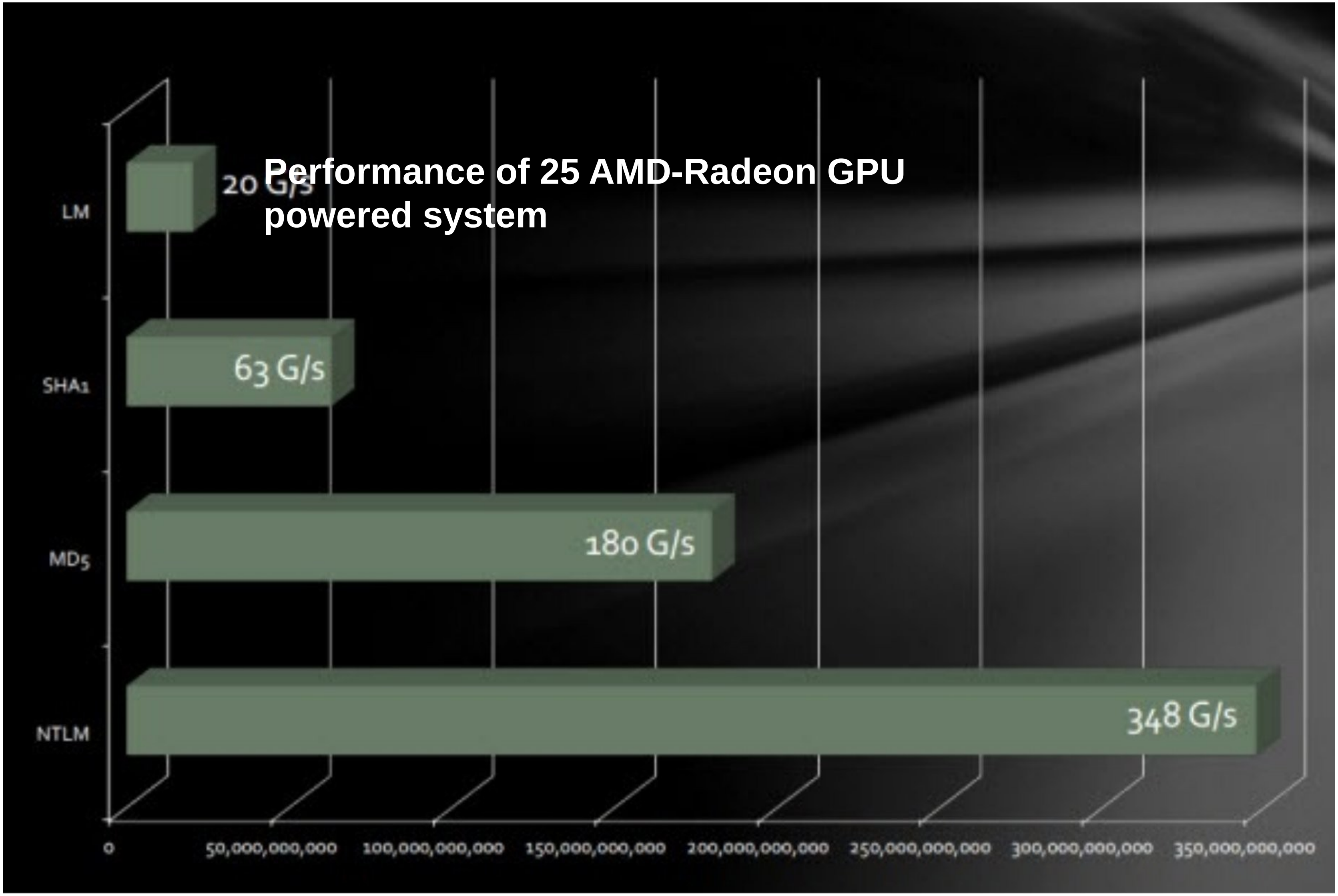


Easy-to-remember passwords rely on patterns or algorithms

- that can be used to generate a candidate list
- Dictionary can also be built from passwords stolen from other sites

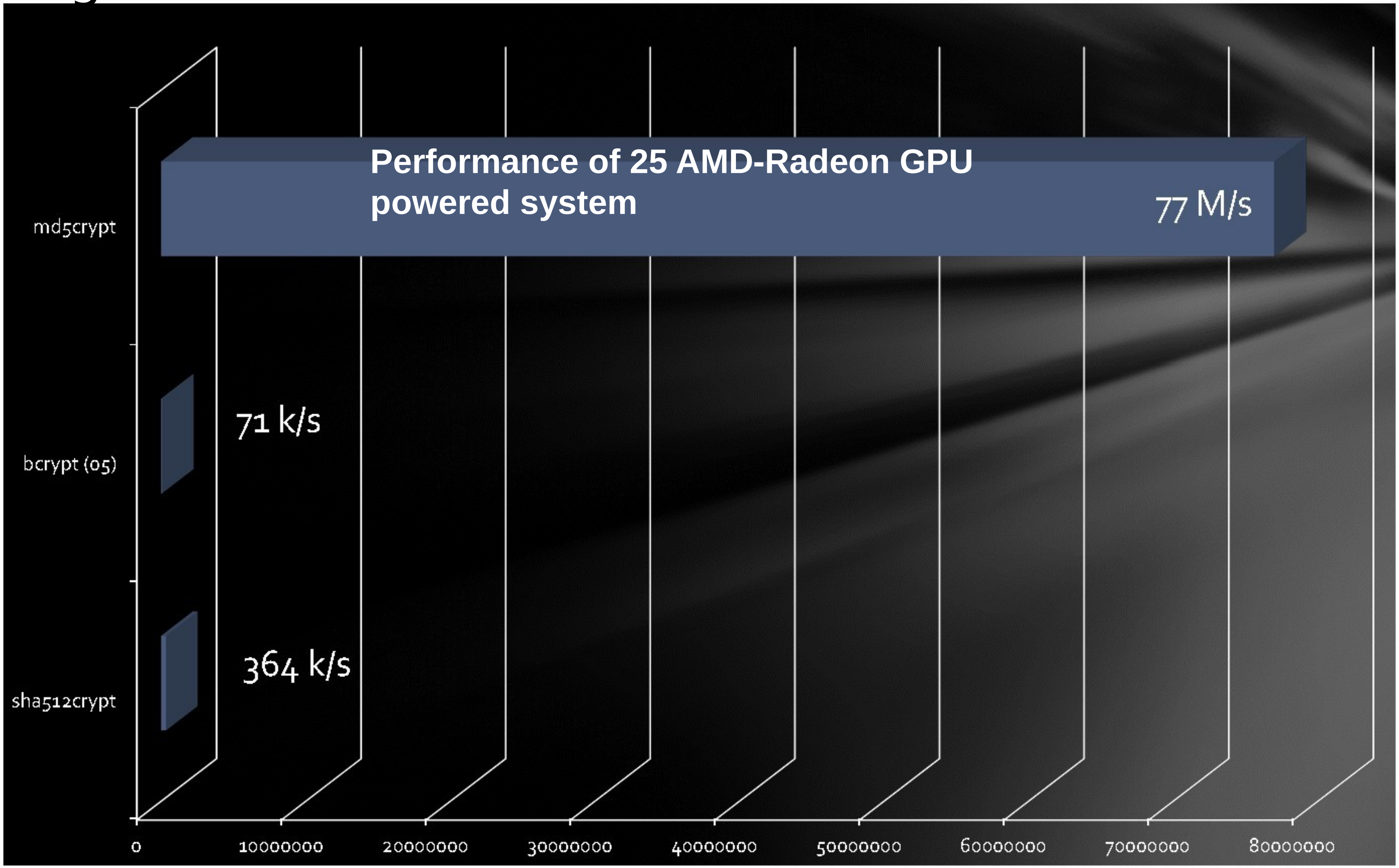
Password weaknesses [Gosney 12]

- Brute-force, dictionary attacks greatly speeded by GPUs



Password weaknesses [Gosney 12]

- Even GPUs are not too fast for some hash algorithms



Password Defense Measures

- Slowing 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: $DES^{25}_{Password||salt}(0)$
    - Eliminates attacks that hash once, compare against passwords of all users
- Protecting password file
  - /etc/passwd is world-readable, so easy to steal
  - Modern UNIX versions separate password hashes into a /etc/shadow file that is read-protected

Password Defense Measures

- Online attacks
  - Guessing
    - Delays: remove login prompt after 3 failed attempts
      - Increase delay (e.g., double) after additional failures
    - Lock outs: prevent user from login after N failures
    - CAPTCHAs: make user solve CAPTCHA after N failures
  - Stealing
    - Network sniffers (solutions discussed later)
    - Phishing (fake password dialogs)
    - Keyloggers and other malware
    - Password reset

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)
  - people lulled into accepting self-signed certificates
    - But today’s browsers provide stronger warning (or silenty suppress) sites that change a CA-provided certificate into a self-signed one
  - social engineering (“our SSL servers are down today”)
  - DNS redirects!
  - Compromise of Certification Authorities
    - Once thought unlikely, but is increasingly being used against high-value targets

Summary of Password weaknesses

- Brute-force attacks greatly speeded up by GPUs
- Use of weak passwords
- Dictionary attacks
- Keyloggers (and formerly, network sniffers)
- Social engineering (phishing)
- Password reset mechanisms

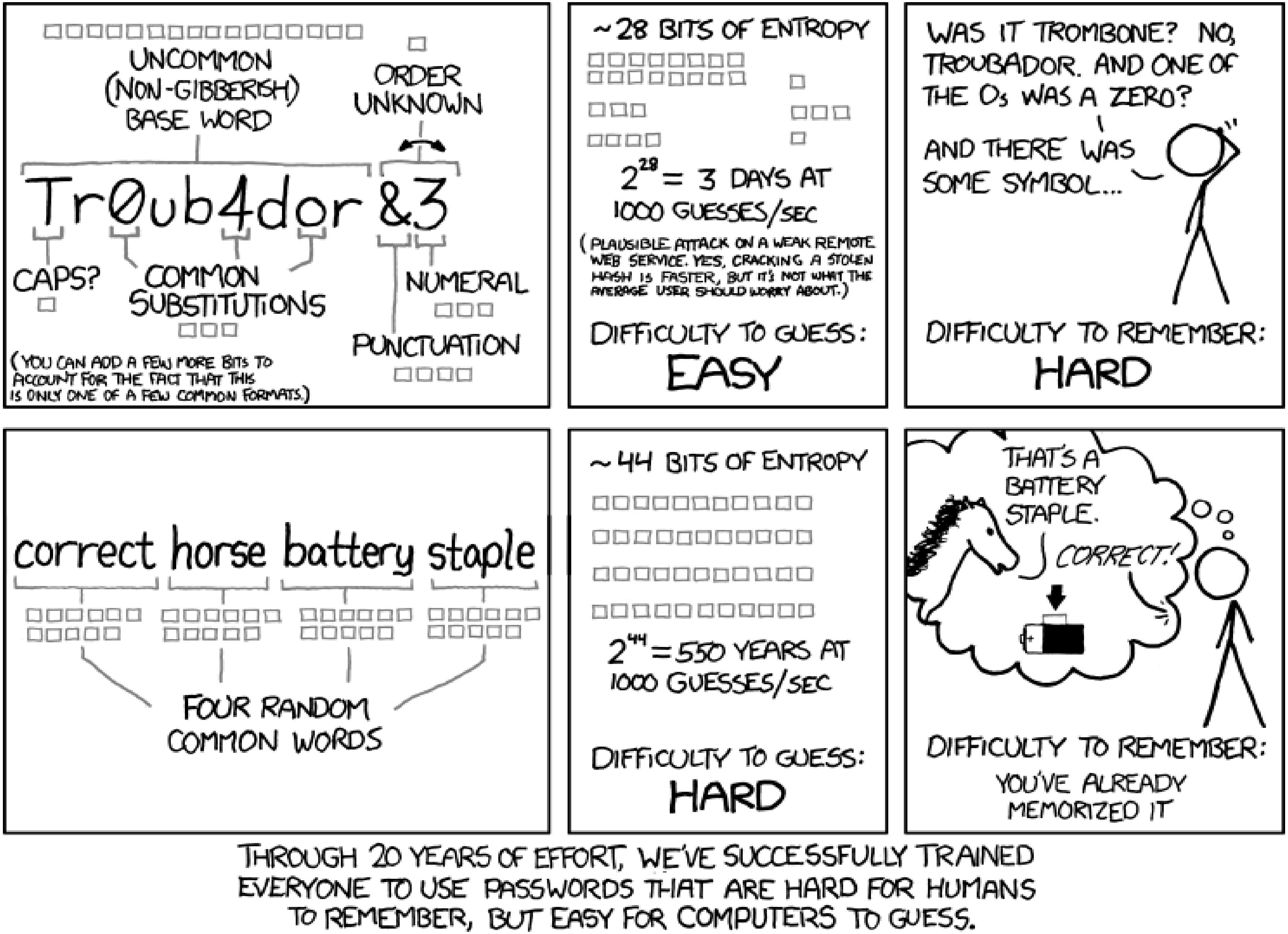
More password problems

- 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

Password weaknesses: Non-solutions

- CAPTCHAs to defeat guessing 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
    - Users often add easily guessed prefix or suffix to a simple password, e.g., “0-” or “#1”
- Alternative password schemes
  - Face or picture recognition

Taken from:
http://imgs.xkcd.com/comics/password_strength.png



Improving basic password schemes

- Using master password
- One-time passwords
- Two-factor authentication
  - Small device or smartcard, combined with a password
- Biometrics (?)
- Visual passwords (??)

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
      - ssh
      - Browsers

Password managers on browsers

- Benefits
  - 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
    - Avoid password being revealed to sites that
      - look similar
      - have URLs that are misspelled or have typos
      - use http instead of https
  - Immune to keyloggers and malware snooping on cut/paste buffers
    - But key loggers can capture your master password
- Drawbacks
  - Bad idea on shared devices
  - False sense of security if master password can be stolen

Authentication across the network

- Trust client to authenticate (avoid network transmission of password)
  - Host-based authentication
    - Used in NFS, also rsh/rlogin/rexec with hosts.equiv
  - Not a great option today, as users often have admin privileges on client machines
- 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 trust all clients on the network
    - Otherwise, easy password compromise by network sniffers

Authentication across the network

- Trust client to encrypt user-supplied password
  - The encryption part is performed by the client, while the checking part is done by the server
  - Only encrypted password transmitted over network
    - But it is as good as unencrypted password!
      - A rogue client can sniff and reuse this encrypted password to log into the server, without ever needing to decrypt it
- Solutions against such *replay attacks*
    - One-time passwords (theft no longer a problem!)
    - Challenge-response protocols (esp. using public keys)

One-time passwords

- 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 -> User: i
  - User -> 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(current OTP) = prev OTP

Challenge-response protocols

- SSH
  - Password based authentication
    - S --> C: KU_S
    - C --> S: $E_{KU_S}(K_{SES} = \text{random}())$, $E_{K_{SES}}(\text{password})$
    - All subsequent communication encrypted using K_{SES}
    - Problems: 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
    - C --> S: KU_{USER}
    - S --> C: Verify presence in ~user/.ssh/authorized_keys, send challenge = $E_{KU_{USER}}(\text{random})$
    - C --> S: decrypt and send the result

Challenge-response protocols

- Web sites use password authentication over https
  - S → C: Public key certificate $E_{KR_{CA}}(KU_s)$
  - C → S: $E_{KU_S}(K_{SES} = random())$
  - All subsequent communication encrypted using K_{SES}
- Similar to SSH password authentication
- Protocols such as telnet can be made secure by simply carrying their traffic over https
- Challenges
  - Certificates cost $$, so there are self-signed certs
    - Users get used to certificate violations, ignore warnings
  - Recent browsers make it difficult to ignore warnings
    - Some violations silently disallowed, e.g., changes to certificates of certain servers

Two-factor authentication: 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

User Authentication Approaches

- Something you know
  - A secret key (password)
  - 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.
- Issues
  - User acceptability
  - 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 of biometric product sales (over 75%)
- Issues:
  - even low error rates can compound when doing a one-to-many match
  - birthday attack issues
  - crime scene issues: dirty, smudged, smeared prints
  - 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.
- 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
- Hand-drawn pictures
- Retina
- DNA

Problems with Biometrics

- age of reference data (eg 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
    - Concrete nouns

Issues with Graphical Passwords

- Many of the basic attack techniques continue to work
  - Dictionary attacks, guessing, social engineering, …
- Shoulder-surfing
- Entropy
  - User studies have revealed that users tend to favor some images over others, e.g., pretty faces of people from one's own race
- Memorability has not been conclusively demonstrated

Password weaknesses: Solutions

- Password managers, master passwords
  - Often thwarted by lawyers and administrators
- Public keys, e.g., SSH or PGP
- Two-factor authentication
  - Tokens, cards, biometrics, …
- One-time passwords or PINs
  - Especially useful if a channel trusted by both sender and receiver is always available, e.g., SMS

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
  - One-time passwords