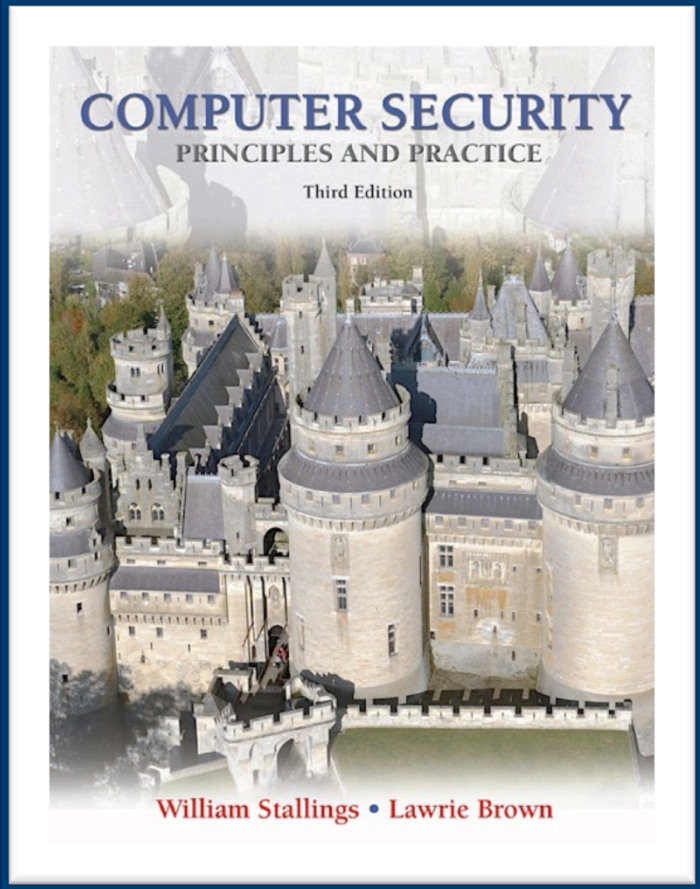
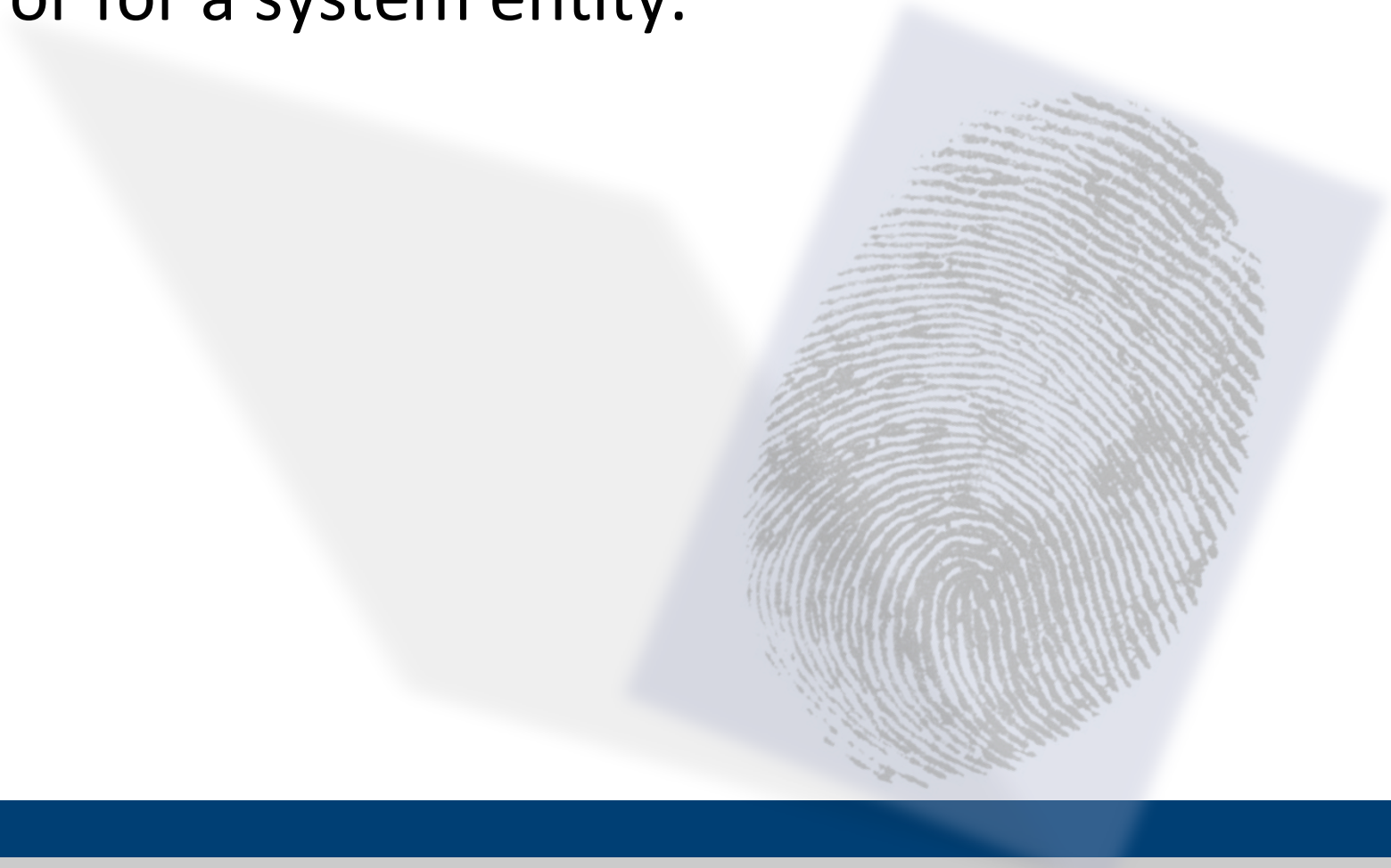
Lecture 3
User Authentication

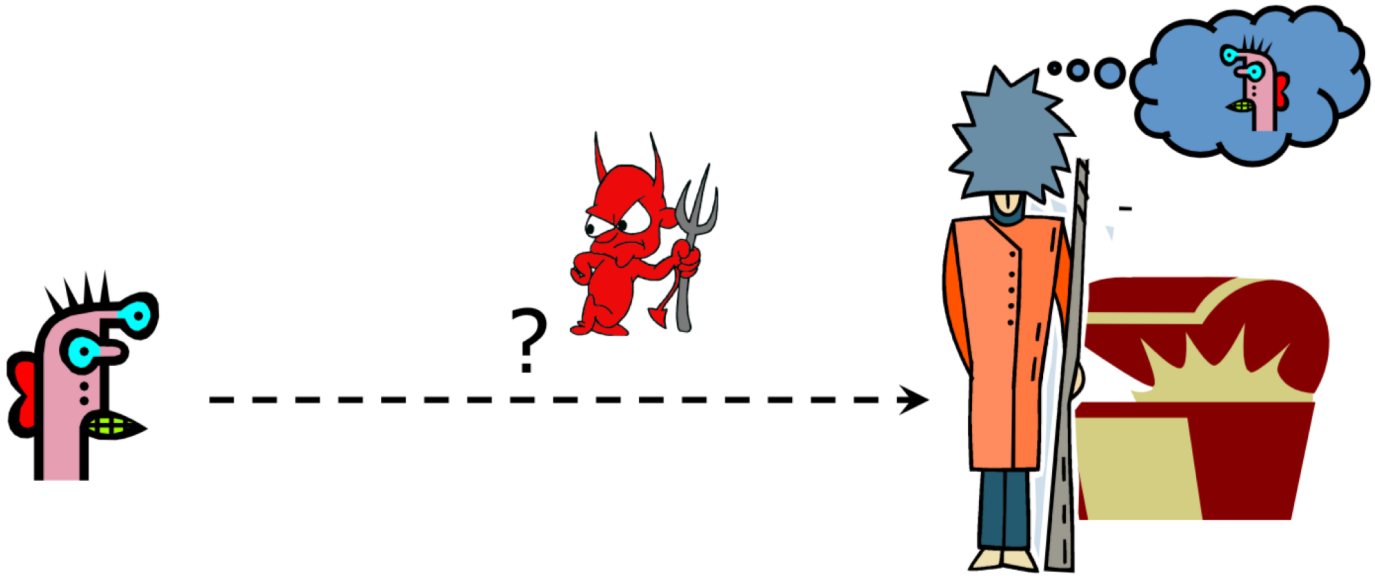


RFC 2828: user authentication

"The process of verifying an identity claimed by or for a system entity."



Basic Problem



- How do you prove to someone you are who you claim to be?
- Man-in-the-middle attack, etc.
- Any system with access control must solve this problem.

Authentication Process

- Fundamental building block and primary line of defense
- Basis for access control and user accountability
- **Identification step**
  - presenting an identifier to the security system
- **Verification step**
  - presenting or generating authentication information that corroborates the binding between the entity and the identifier

User Authentication

the four means of authenticating user identity are based on:

something the individual knows	something the individual possesses (token)	something the individual is (static biometrics)	something the individual does (dynamic biometrics)
• password, PIN, answers to prearranged questions	• smartcard, electronic keycard, physical key	• fingerprint, retina, face	• voice pattern, handwriting, typing rhythm

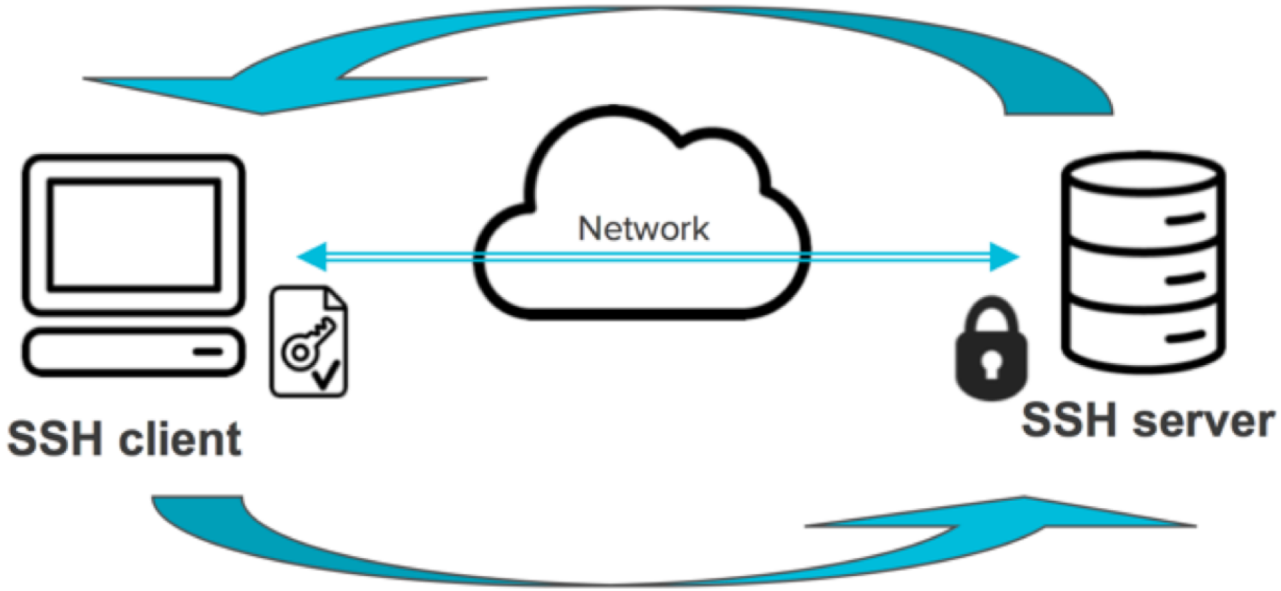
- https://youtu.be/7kNvZuZp6YI?t=15s

Password Authentication

- A widely used line of defense against intruders
  - user provides name/login and password （identification step）
  - system compares password with the one stored for that specified login (verification step)

Storing User Password

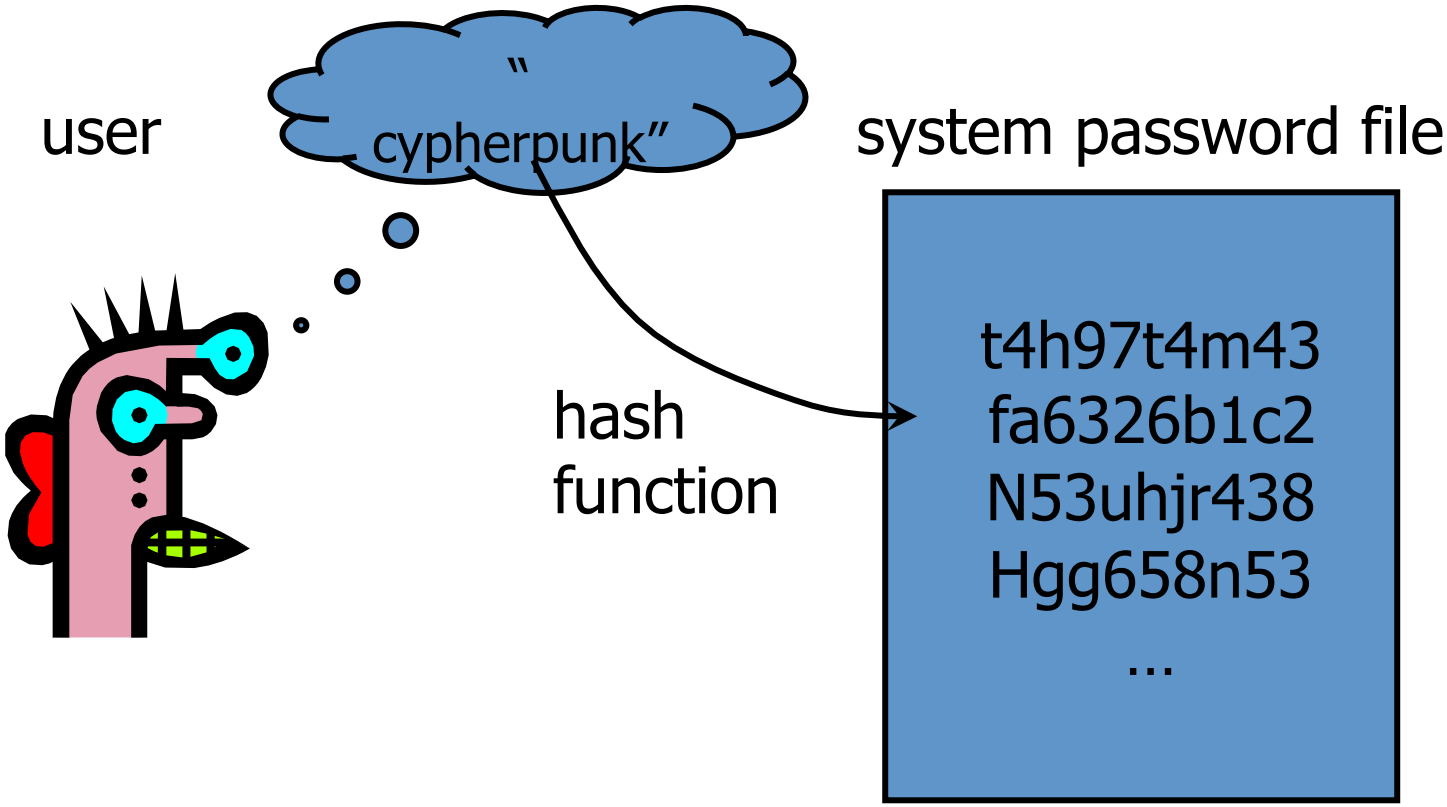
- **Where to store:**
  - Store passwords on individual machine
  - Store all passwords at authentication storage node: authentication done at server



Storing User Password

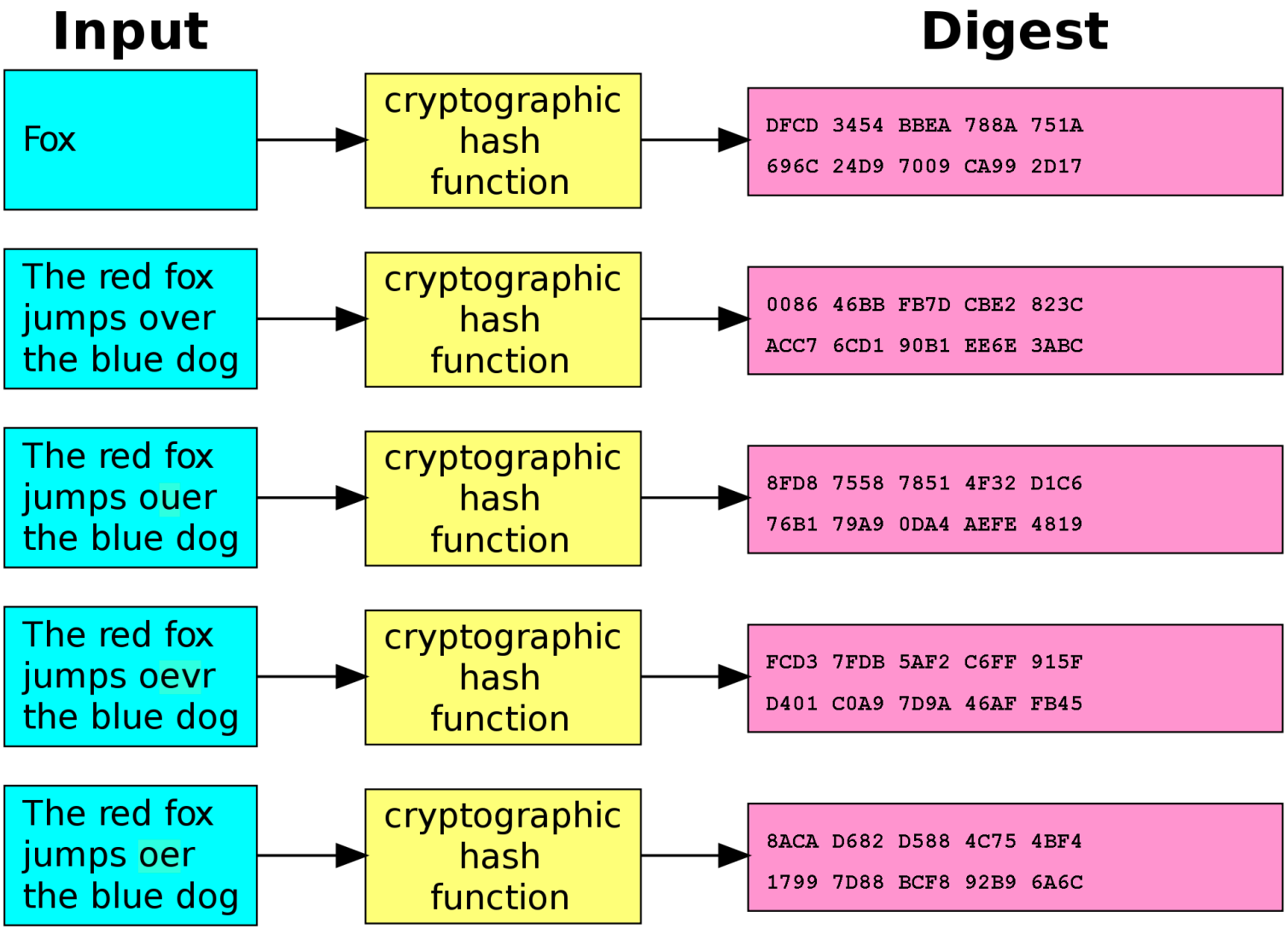
- **In what format:**
  - Store passwords in plaintext
  - Store password hashes
  - Store encrypted passwords
  - Store encrypted password hashes

UNIX-Style Passwords



Hashed passwords are originally stored in a publicly accessible file /etc/passwd

Hash Functions

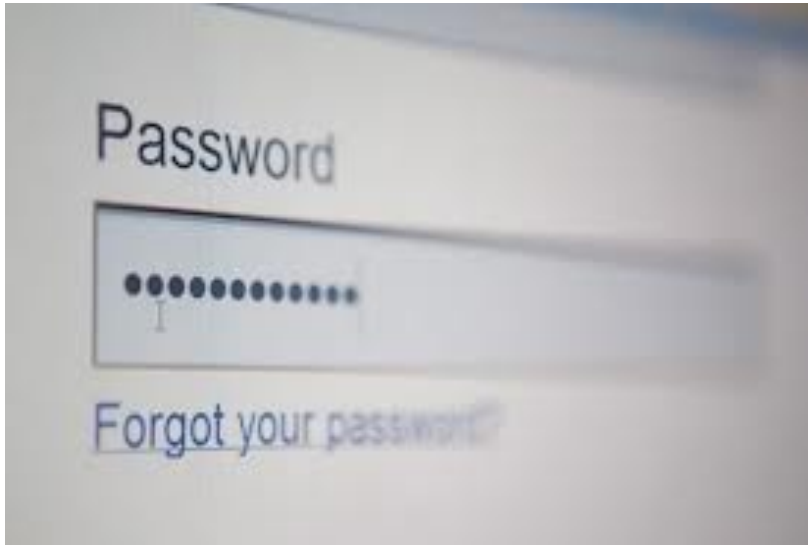


The most used 20 passwords

Tips for creating strong passwords

TIP #1 - LENGTH

- Make your passwords long
- 8 or more characters is a good length
- Almost 3% of the passwords were 3 characters or fewer in length.



- **Why pwds with 3 characters are weak?**

- **Why pwds with 3 characters are weak?**
- Ans: X^3
  - Where x stands for the number of all possible characters

TIP #2 – COMPLEXITY-1

- Include letters, punctuation, symbols, and numbers.
- Use the entire keyboard, not just the letters and characters you use or see most often.
- The greater the variety of characters in your password, the better.

- **If only using numbers as pwds, how many possible pwds are there? Assume each pwd has the length of 8.**

- **If only using numbers as pwds, how many possible pwds are there? Assume each pwd has the length of 8.**
- Ans: 10^8

- **If only using numbers as pwds, how many possible pwds are there? Assume each pwd has the length of 8.**
- Ans: 10^8
- **If using numbers and lebers (both upper and lower cases), how many possible pwds are there? Assume each pwd has the length of 8.**

- **If only using numbers as pwds, how many possible pwds are there? Assume each pwd has the length of 8.**
- Ans: 10^8
- **If using numbers and lebers (both upper and lower cases), how many possible pwds are there? Assume each pwd has the length of 8.**
- Ans: 62^8

TIP #3 - VARIETY

- Don't use the same password for everything!

- Cybercriminals steal passwords on websites that have very little security, and then they use that same password and user name in more secure environments, such as banking websites.

TIP #4 - VARIATION

- To keep strong passwords effective, change them often.

- Remind yourself to change your passwords every three months or so.

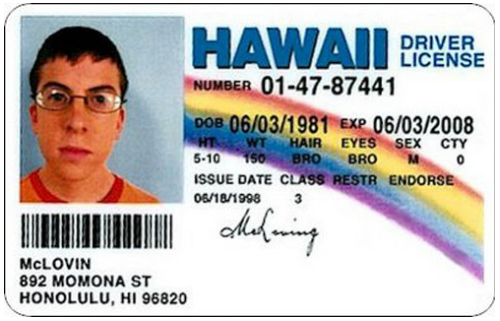
TIP #5 – THINGS TO AVOID

- **Dictionary words in any language.**
  - e.g. iCloud hack
  - https://youtu.be/opRMrEfAIiI?t=42s
- **Words spelled backwards, common misspellings and abbreviations.**
- **Top Worst Passwords of 2012**
  - password
  - 123456
  - 12345678
  - abc123
  - qwerty

TIP #5 – THINGS TO AVOID

- Sequences or repeated characters. Examples: 12345678, 222222, abcdefg, or adjacent letters on your keyboard (qwerty).

- Personal information. Your name, birthday, driver's license, student number, passport number, or similar information.



Exercises-1

- Assume that passwords are selected from four-character combinations of 26 alphabetic characters. Assume that an adversary is able to attempt passwords at a rate of one per second. What is the maximum time to discover the correct password?

Exercises-1

- **Assume that passwords are selected from four-character combinations of 26 alphabetic characters. Assume that an adversary is able to attempt passwords at a rate of one per second. What is the maximum time to discover the correct password?**
- Ans: 26^4

Exercises-2

- **A phonetic password generator picks two segments randomly for each six-letter password. The form of each segment is CVC (consonant, vowel, consonant), where V=<a, e, i, o, u> and C=complementary{V}.**
  - what is the total password population?

Exercises-2

- **A phonetic password generator picks two segments randomly for each six-letter password. The form of each segment is CVC (consonant, vowel, consonant), where V=<a, e, i, o, u> and C=complementary{V}.**
  - what is the total password population?
  - Ans: T=(21*5*21)^2

Exercises-2

- **A phonetic password generator picks two segments randomly for each six-letter password. The form of each segment is CVC (consonant, vowel, consonant), where V=<a, e, i, o, u> and C=complementary{V}.**
  - what is the total password population?
  - Ans: T=(21*5*21)^2
  - what is the probability of an adversary guessing a password correctly?

Exercises-2

- **A phonetic password generator picks two segments randomly for each six-letter password. The form of each segment is CVC (consonant, vowel, consonant), where V=<a, e, i, o, u> and C=complementary{V}.**
  - what is the total password population?
  - Ans: T=(21*5*21)^2
  - what is the probability of an adversary guessing a password correctly?
  - P=1/T

Exercises-3

- **Explain the suitability of the following passwords**
  - YK334

Exercises-3

- **Explain the suitability of the following passwords**
  - YK334
  - mfmitm (for “my favorate movie is tender mercies”)

Exercises-3

- **Explain the suitability of the following passwords**
  - YK334
  - mfmitm (for “my favorate movie is tender mercies”)
  - Natalie1

Exercises-3

- **Explain the suitability of the following passwords**
  - YK334
  - mfmitm (for “my favorate movie is tender mercies”)
  - Natalie1
  - Washington

Exercises-3

- **Explain the suitability of the following passwords**
  - YK334
  - mfmitm (for “my favorate movie is tender mercies”)
  - Natalie1
  - Washington
  - tv9stove

Exercises-3

- **Explain the suitability of the following passwords**
  - YK334
  - mfmitm (for “my favorate movie is tender mercies”)
  - Natalie1
  - Washington
  - tv9stove
  - 12345678

Exercises-3

- **Explain the suitability of the following passwords**
  - YK334
  - mfmitm (for “my favorate movie is tender mercies”)
  - Natalie1
  - Washington
  - tv9stove
  - 12345678
  - olleh

Attacks to PWD Authentications

- Offline dictionary attack
  - Dictionary: a set of pwds that are commonly chosen
  - Dictionary attack is possible because many passwords come from a small dictionary
  - Attacker can compute H(password) for every password in the dictionary (**rainbow table**) and see if the result is in the password file
    - **Password file is sometimes available to the attacker**

Countermeasures

- Password selection strategy

Countermeasures

- Password selection strategy
- Rapid reissuance of passwords

Countermeasures

- Password selection strategy
- Rapid reissuance of passwords
- **Controls to prevent unauthorized access to password file**

Password File Access Control

can block offline guessing attacks by denying access to encrypted passwords

make available only to privileged users

shadow password file

- a separate file from the user IDs where the hashed passwords are kept

vulnerabilities

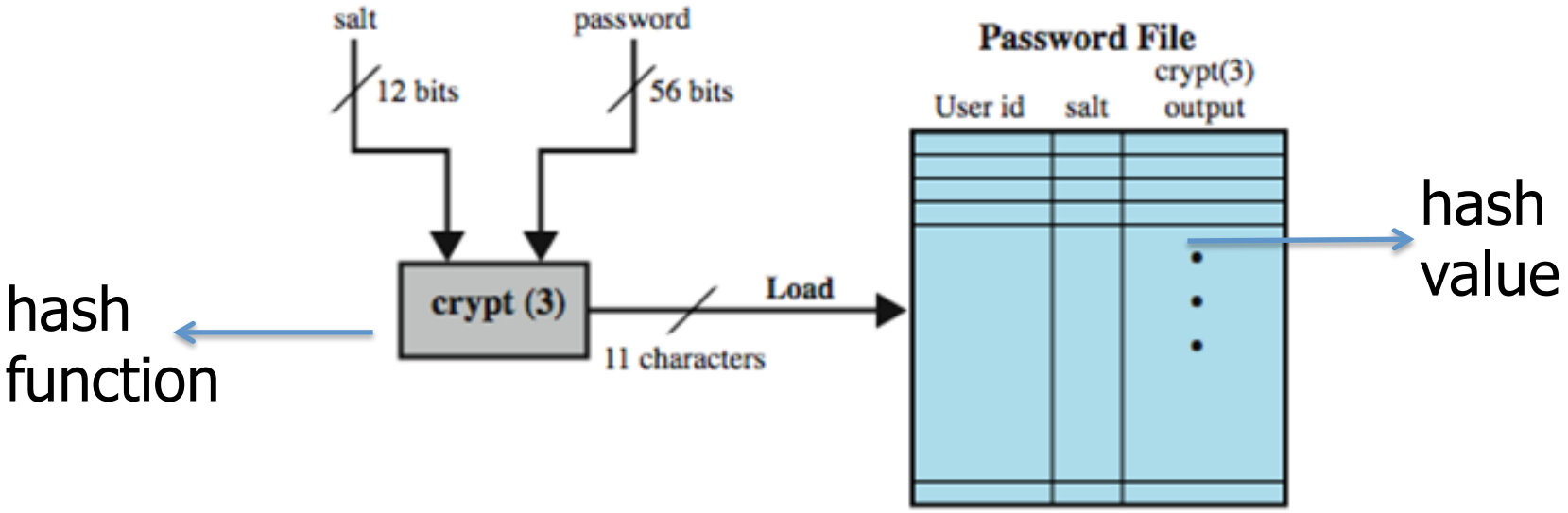
- weakness in the OS that allows access to the file
- accident with permissions making it readable
- users with same password on other systems
- access from backup media
- sniff passwords in network traffic

Countermeasures

- Password selection strategy
- Rapid reissuance of passwords
- Controls to prevent unauthorized access to password file
- **Salt**

Functionality of Salt—add a pwd

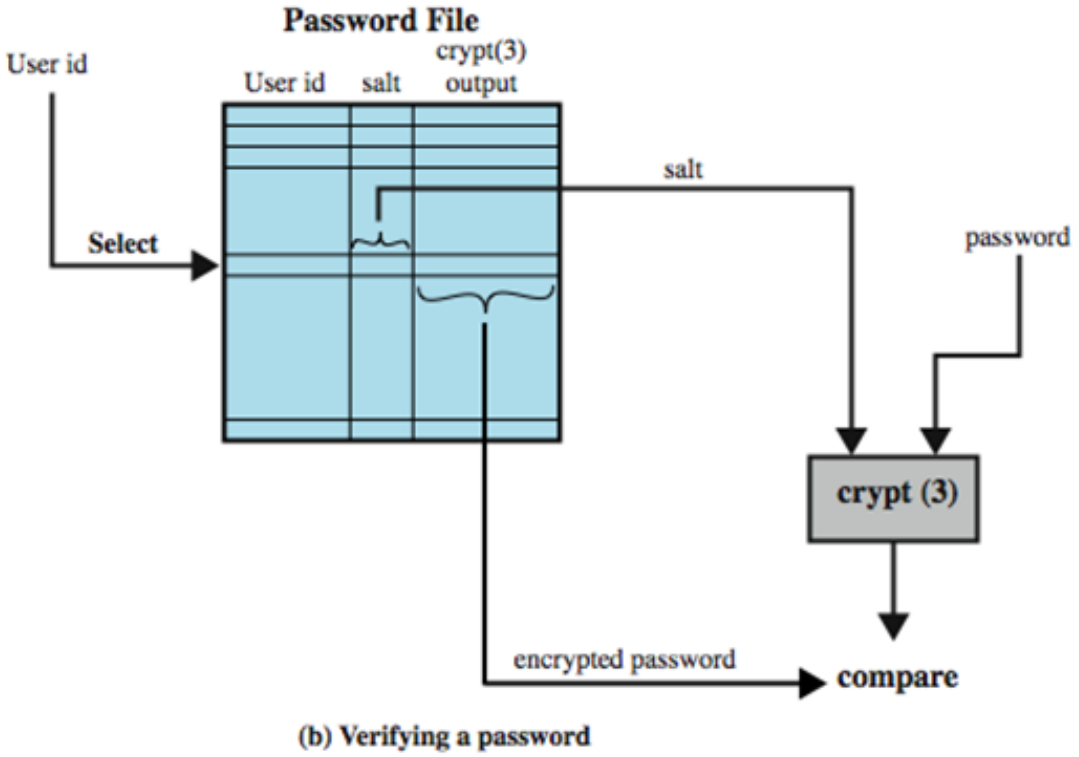
- A password is combined with a fixed-length **salt value**
- The hash value of the salted pwd is calculated and stored in the pwd file



(a) Loading a new password

Functionality of Salt—verify a pwd

- Pwd verification
  - Locate the salt value according to user ID; calculate h(pwd||salt); compare it to the stored item in pwd file



(b) Verifying a password

Why Can Salt Relieve Dictionary Attack?

- To crack a pwd…
  - Without a salt, an attacker could compute hash value of each item in the dictionary, and check if this hash appears anywhere in the pwd file
  - With a salt…