Identification and Authentication

Cunsheng Ding
HKUST, Hong Kong, CHINA
cding@cs.ust.hk

User Identification & Authentication

- Is he/she a registered user at the system? (user identification)
- Can he/she prove who he/she is? (user authentication)

Username and Password

- **Username**: *identification purpose*
- **Password**: *authentication purpose*

- Remark: The mostly used approach to identification and authentication.

Problems with Passwords

- Password guessing by attacker
- Compromise of system's password file
- Theft, accidental disclosure, forced disclosure

Guessing a Password

- **Exhaustive search**: try all possible combinations of valid symbols, up to a certain length.
- **Intelligent search**: search through a restricted name space, e.g., names of friends and relatives, car registration number.

Defenses by **User**

- **Changing default password**: delivered system has a default password.
- **Password length**: thwart exhaustive search
- **Password format**: mix upper and lower case symbols and numerical characters.
- **Avoid obvious passwords**: birth date, etc.

The Dilemma

- Passwords of complex formats are hard to memorize. If you choose such one, you may have to write it down somewhere and hide it in your office. But this is also dangerous.
- Passwords of simple formats are easy to memorize, but do not offer good security.
- Question: What do you do in practice?

Password Security by System (1)

- **Password checkers**: The system checks passwords against some dictionary of 'weak' passwords.
- **Password generation**: some operating systems include password generators producing random but pronounceable passwords.

Password Security by System (2)

- **Password aging**: in many systems an expiring date can be set, to force a user to change his password.
- **Limit login attempts**: the system can monitor unsuccessful login attempts and react by locking the user account completely or at least for a certain period of time.

Password Security by **System** (3)

- **Inform user**: after a successful login, the system can display the time of the last login and the number of failed attempts since then, to warn the user about recent attempted attacks.

Spoofing Attacks

- **Purpose of attack**: password compromise
- **Who is the attacker**: anyone including a legitimate user.
- **How to attack**: an attacker runs a program that presents a fake login screen on some terminal or workstation. An unsuspecting user comes to this terminal and try to login. The ID and password are then stored. Execution is then handed over to the user, or login is aborted with a (fake) error message and the spoofing program terminates.

Defense against Spoofing Attacks

- **Inform**: Displaying the no. of failed logins may indicate to the user that such an attack happened.
- **Trusted path**: guarantee that user communicates with the operating system and not with a spoofing program. How?

Other Cases: Example

- **Information**: Web browsers cache information that make it possible for users to scroll back to pages they have recently visited.
- **Question**: Does this cause any security problem?

Example: Online Banking

- You enter your password on a web page, conduct your business, close the banking application, but **forget to** terminate the browser session.
- The next user can scroll back to the page with your password and login as you.

Password File

- **User Identity Verification**: the system compares the password entered by the user against a value stored in the password file.
- **Claim**: The password file must be protected. (rationale later)
- **How to protect the file?**

Protection of Password File

- **Cryptographic Protection**: encrypt the password file.
- **Question**: Does it solve the problem?
- **Answer**: If the encrypted file is obtained, one may carry out an "offline dictionary attack".
  - Please use the picture on Page 21 to explain the offline dictionary attacks!

Protection of Password File

- **Protection by operating system**: Access control to the file enforced by the operating system
- **Better protection**: A combination of cryptographic protection and access control, possibly with even further enhancements to slow down dictionary attacks.

Protection of Password File

- **Question**: How to encrypt a password file?
- **Answer**: One-way functions (follows)

- **Question**: How to protect a password file by access control enforced by an operating system?
- **Answer**: later

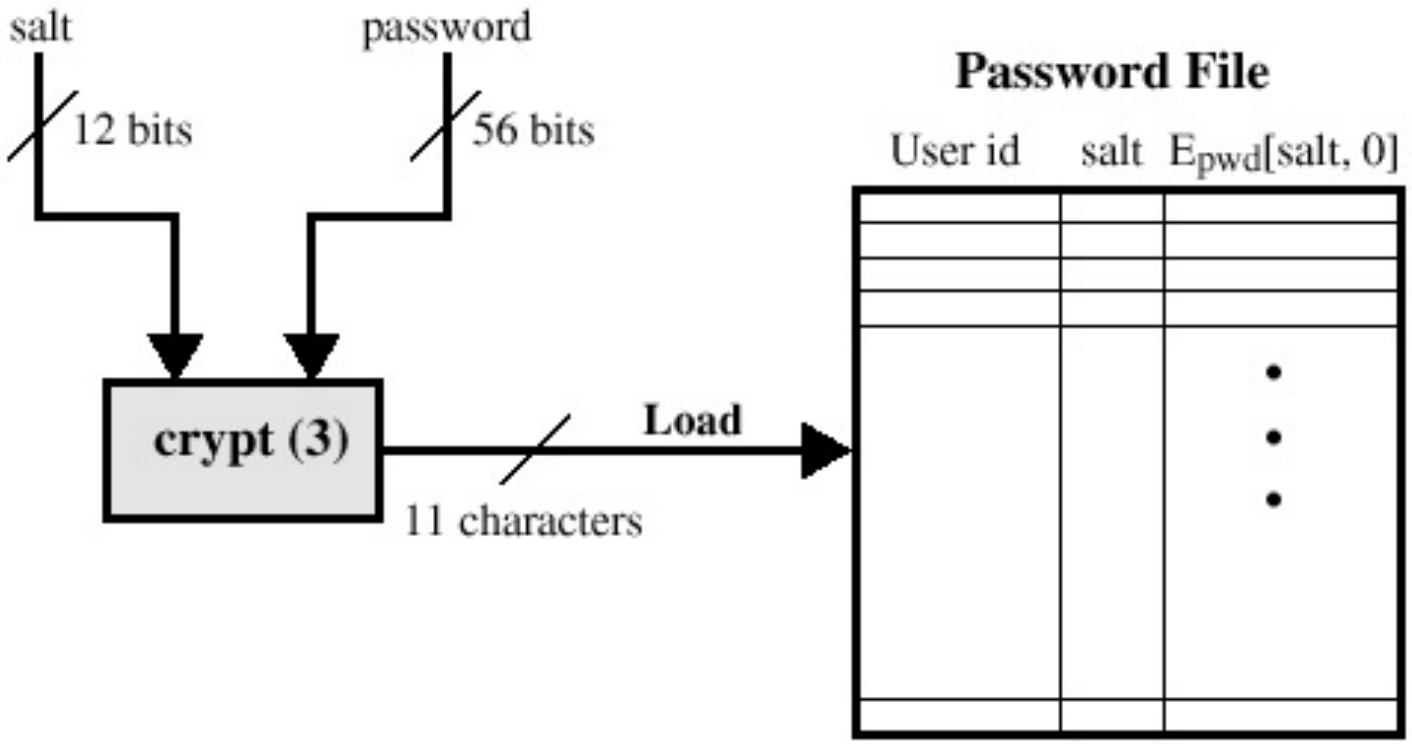
One-way Functions

- **Definition**: Given x, it is easy to compute $f(x)$; but it is hard to find x given $f(x)$.
- **Example**: $f(x) = a^x \bmod p$
- **Applications**: stored password protection, and other applications in cryptography.

One-way Function Crypt(3)

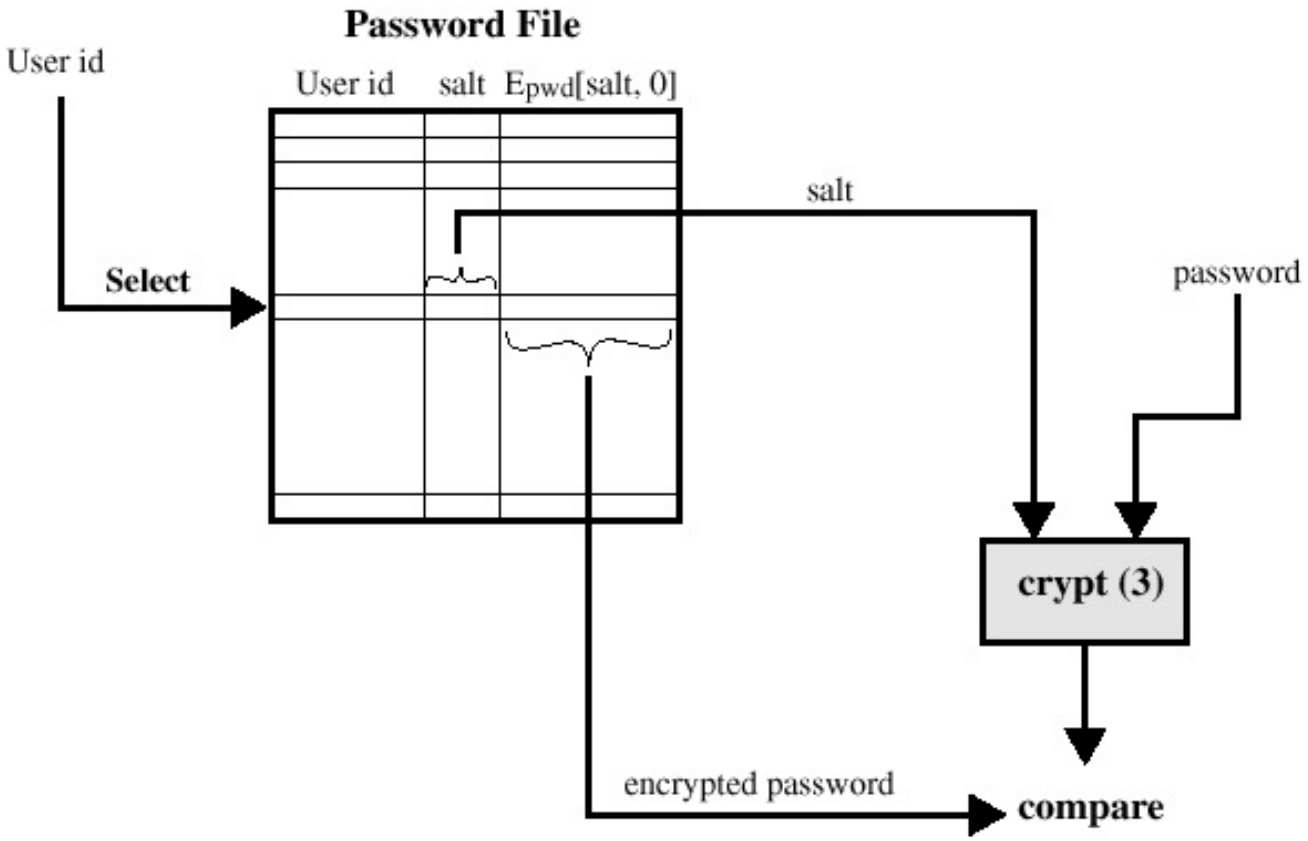
- It repeats a slightly modified DES algorithm 25 times, using the all-zero block as starting value and the password as key.
- It is used to "encrypt" the password file in Unix.

UNIX Password Scheme



Loading a new password

UNIX Password Scheme



Verifying a password file

"Salt"

- The salt serves the following purpose:
  - Prevents duplicate passwords from being visible in the passowrd file.

Protection of Password File

- **Question:** How to encrypt a password file?
- **Answer:** One-way functions

- **Question:** How to protect a password file by access control enforced by an operating system?
- **Answer:** follows

Password Protection by Access Control

- **Remark: Access control mechanisms** in an operating system restrict access to files and other resources to users holding the appropriate privileges.
- **Remark:** Only privileged users can have write access to the password file.
- **Remark: You cannot change other people's password if access control is successful!**

Password File in UNIX

- **Remark:** If the encrypted password file is compromised, an offline dictionary attack is possible.
- **Information:** Recent versions of Unix store encrypted passwords in a file that is not publicly accessible. Such files are called ***shadow password files.***
- E.g. in HP-UX, **/.secure/etc/passwd**

Password Security in Summary

- A combination of mechanisms can enhance protection. **Encryption** and **access control** are used to guard password files.

Alternative Approaches to User Identification & Authentication

- **Something you know**:
- **Example**: personal identification number (PIN) used with bank card.

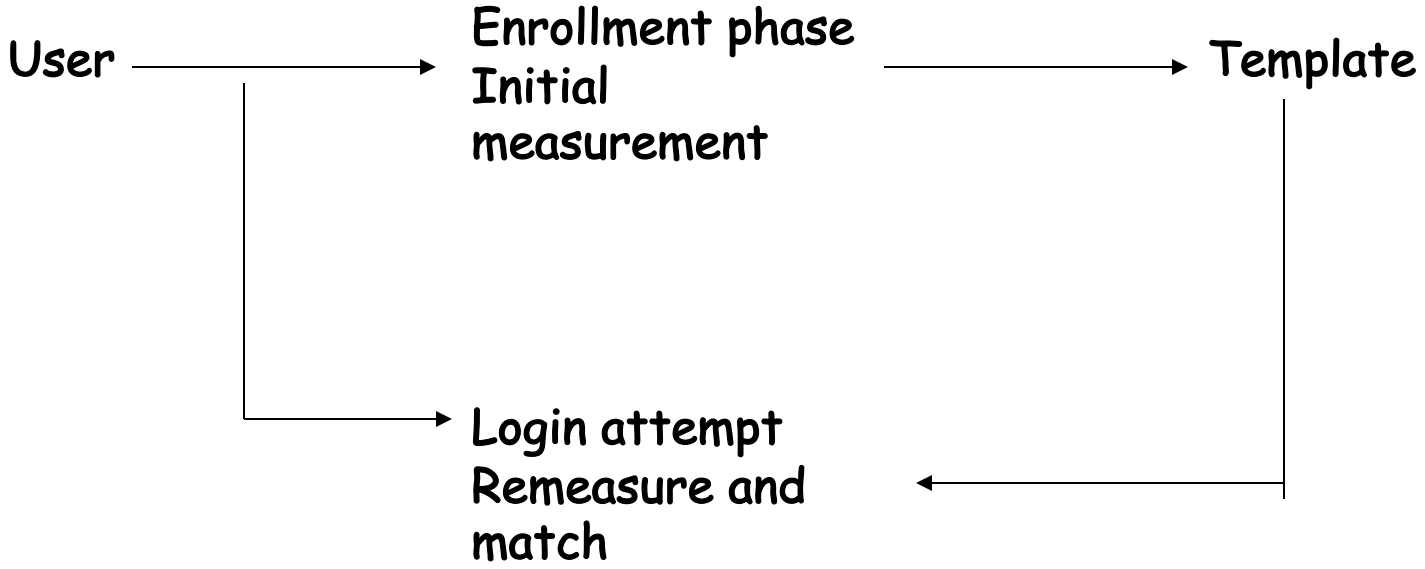
Alternative Approaches ctd.

- **Something you hold**:
- **Example**: a physical key opening a lock.
- **Example**: A card for accessing the office of the CS Department.
- **Remark**: A physical token can be lost or stolen.

Alternative Approaches ctd.

- **Who you are**: Example - biometrics
- Example: palm prints (掌心印)
- Example: fingerprints
- Example: face recognition
- Example: iris pattern (眼球虹膜形状)
- Example: retina pattern (视网膜形状)
- Example: voice recognition
- Example: signature recognition

Operation in Biometrics Authentication



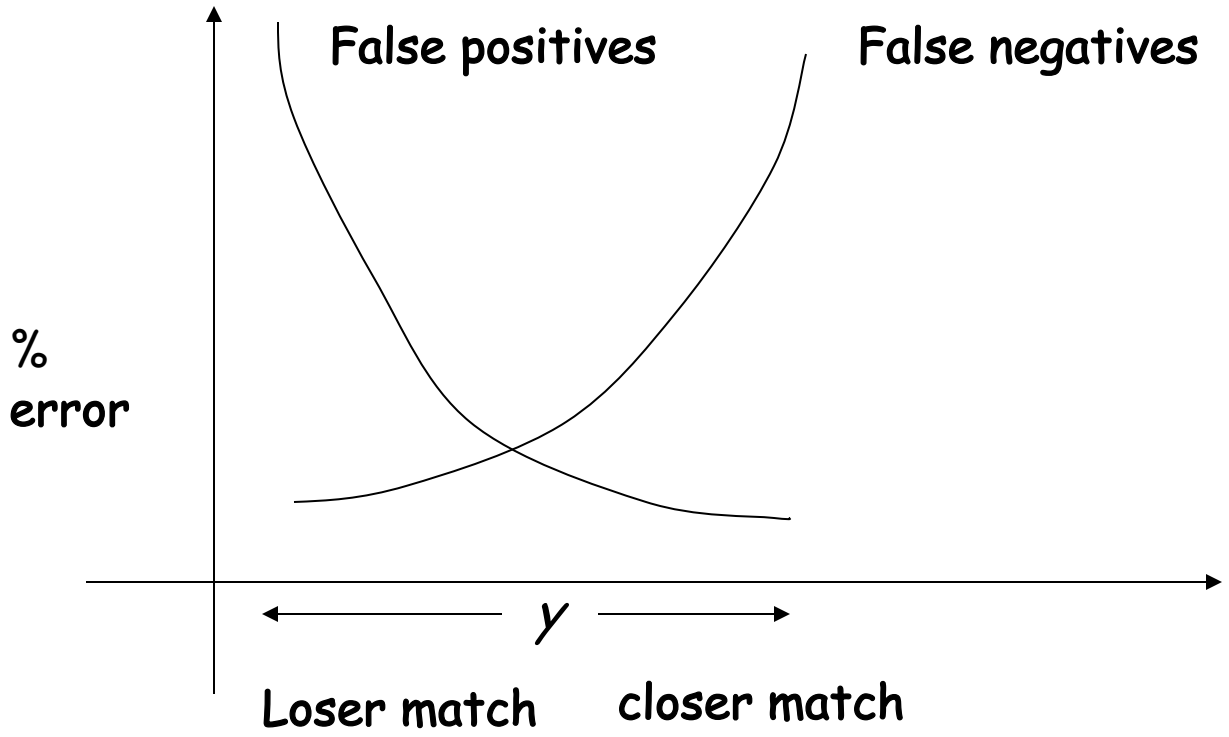
Problems with Biometrics Authentication

- **Initial measure of physical characters:** What measure? How to measure?
- **Authenticated:** how good is a match required before a user is considered authenticated?
  - Your face picture may vary from time to time due to weight change or your mode each day.
  - So the very nature of biometrics requires measuring some property which may vary from time to time, and an algorithm must decide how much variation to allow.

Error Rates in Biometrics Authentication

- Let y denote the set of parameters affecting the error rates.
- An authentication system can have two kinds of errors:
  - **false positives**: in which an unauthorized user is accepted;
  - **false negatives**: in which an authorized user is rejected.

The Graph for Error Rates *R(y)*



Design Issues

- A system designer must choose his parameters according to his security policy.
  - **Example:** a bank system might decide that too many false negatives would drive customers away, and opt to absorb the occasional loss in favor of keeping customers.
  - **Example:** a military installation would prefer to have parameters requiring a very strict match.

Other Issues

- Do we care if other people have these biometrics?
- Biometrics systems still must protect against standard attacks. E.g., a malicious user could impersonate the template server or hijack a connection after authentication.
- What systems will users accept?
  - Voice recognition, signature verification, and keyboard time are most likely to be accepted.

Comments on Biometrics

- Biometrics are powerful and useful, but they are not keys. They are useful in situations where there is a trusted path from the sender to the verifier.
- In those cases all you need is a unique identifier.
- They are not useful when you need the characteristics of a key:
  - secrecy, randomness, the ability to update and destroy.
- Biometrics are unique identifiers, but not secrets.

Problems for Discussion

- Passwords are entered by users and checked by computers. Thus there has to be some communications channel between user and computer. So far we have taken a very abstract view of this channel and assume that it exists and that it is adequately secure.
- When is this assumption justified? When is it unjustified?

Further Reading

- R. Morris and K. Thompson. Password security: A case history. Communications of the ACM, 22(11) pp. 594-597, Nov. 1997.
- L. Hadfield, D. Hatter, and D. bixler. Windows NT server 4 security handbook. Que Corporation, Indianapoils, IN, 1997.
- T. Sheldon. Windows NT security Handbook. McGraw-Hill, 1997.