Authentication Systems

Ahmet Burak Can
Hacettepe University
abc@hacettepe.edu.tr

Entity Authentication

- Entity authentication (identification): the process whereby one party is assured of the identity of a second party involved in a protocol.
  - Entities can be people, processes, etc.

- Authentication can be done in many ways ☺

  https://youtu.be/ll6Ci-fkFtA

Entity Authentication

- Non-cryptographic
  - Address-based (E-mail, IP, etc.)
  - Passwords
  - Biometrics
- Cryptographic
  - Symmetric key
  - Public key

Requirements of Authentication Protocols

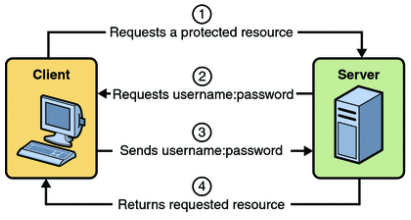
- Requirements of identification protocols
  - for honest prover A and verifier B, A is able to convince B
  - no other party can convince B
  - in particular, B cannot convince C that it is A
- Authentication can be based on
  - What you know? (password schemes)
  - What you have? (keys, smart cards, etc.)
  - What you are? (fingerprints, retinal scans, etc.)
- Kinds of attackers
  - passive and replay
  - active, man in the middle
  - the verifier

Properties of Authentication Protocols

- Reciprocity of identification (one-way or mutual)
- Computational efficiency (encryption, signing)
- Communication efficiency (communication rounds, messages)
- Involvement of a third party
- Nature of trust in the third party
- Storage of secrets

Authentication Using Fixed Passwords

- Client authenticates to a server using a password.
  - Passwords must be kept in encrypted password files or as digests



Initial Password Distribution

- Initial off-line authentication
- Passwords can be chosen on site by users
- An initial password can be issued by the system administrator.
- Pre-expired passwords
  - Must be changed at the first login

Attacks on Passwords

- Attacks:
  - Careless users writing down passwords
  - Stealing password files
  - Eavesdropping
  - On-line password guessing
  - Off-line guessing attacks
    - Dictionary attacks
    - Exhaustive search

Eavesdropping

- Watching the screen
- Watching the keyboard
- Login Trojan horses
  - Different appearance
  - Interrupt command for login
- Keyboard sniffers
  - Good system administration
- Network sniffers
  - Cryptographic protection
  - One-time passwords

On-line Password Guessing

- Careless choices (first names, initials, etc.); poor initial passwords
- Defenses: After wrong guesses,
  - Lock the account
    - Not desirable, can be used for DoS
  - Slow down
  - Alert users about unsuccessful login attempts
  - Don't allow short or guessable passwords

Off-line Password Guessing

- Stealing & using password files
- Passwords should not be stored in clear. Typically, they're hashed and stored.
- Attacks:
  - Exhaustive search
  - Dictionary attacks
- Defenses:
  - Don't allow short/guessable passwords
  - Don't make password files readable
  - Salting: Mix a random number to each hash

Unix crypt Algorithm

- Used to store Unix passwords
- UNIX password information stored is in /etc/passwd :
  - Iterated DES encryption of 0 (64 bits), using the first 8 characters of the password as key
  - 12 bit random salt taken from the system clock time at the password creation
- Strengthen passwords by "salting".
  - Why use the salt?: To alter the expansion function E of DES, to defend against attacks on DES using off-the-shelf hardware that can crack DES

One-Time Passwords

- Some systems use a different password for each login operations.
  - A used password expires and then a new password is created for the next login.
  - Example:
    SMS messages sent to cell phone during online banking logins
- Generally, one-time passwords are created using crptograhic algorithms
  - Sometimes a secure device is used for creating one-time passwords

Lamport's One-Time Password

- Stronger authentication that password-based
- One-time setup:
  - A selects a value w, a hash function H(), and an integer t, computes $w_0 = H^t(w)$ and sends w_0 to B
  - B stores w_0
- Protocol: to identify to B for the i^{th} time, $1 \leq i \leq t$
  - A sends to B: A, i, $w_i = H^{t-i}(w)$
  - B checks $i = i_A$, $H(w_i) = w_{i-1}$
  - if both holds, $i_A = i_A + 1$

Challenge-Response Protocols

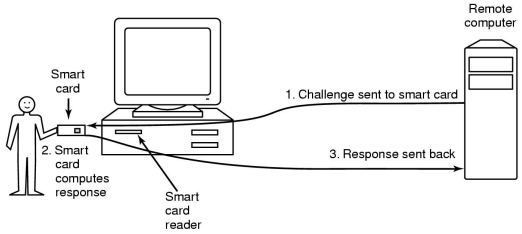
- Goal: one entity authenticates to other entity by proving the knowledge of a secret, not by revealing the secret
- Time-variant parameters used to prevent replay attacks, provide uniqueness and timeliness: nonce (number used only once)
- Three types of challenges:
  - Random numbers
  - Sequences
  - Timestamp

Authentication Tokens

- Keys (physical)
- ATM, credit cards, smart cards
- USB Tokens

Smart Cards

- Smart cards: On-card processor for cryptographic authentication.
  - PIN-protected cards: Memory protected by PIN
  - Challenge-response cards: Performs challenge-response authentication through SC reader
    - New technology: Tokens working through USB ports.



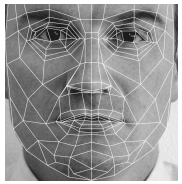
USB Tokens

- Challenge Response Tokens
  - May use public key certificates

- Cryptographic calculator
  - Generally used for one-time password protocols
  - Current time encrypted
  - Displayed to user
  - Entered to terminal

Biometrics

- Authentication by inherent physical characteristics
- E.g., fingerprint readers, retina/iris scanners, face recognition, voice recognition



Problems with Biometrics

- Expensive
  - Generally the recognition devices are expensive or hard to deploy
- Not fault tolerant
  - Face, voice recognition is still not stable enough
- Not possible to change in case of theft
  - If stealed, it is not possible to change a user's biometric info unlike passwords, tokens.
  - Such as steal of fingerprints on the surface of scanner devices and replication of the fingerprint using latex material
- Can be replayed in remote authentication
  - If biometric info is stealed, it can be used by attackers