CS 4910: Intro to Computer Security

Authentication II

Instructor: Xi Tan

Updates

- Project 1
  - due 2/24
- Assignment 2 released
  - due 3/05
- Select your research topic
  - due 3/19

What we already know

- Cryptography tools
- Authentication
  - Definition of entity authentication
  - Solutions
    - Password-based authentication

Authentication

- Message Authentication
  - Message Authentication Code (Keyed Hash) to confirm that the message came from the stated sender (its authenticity) and has not been changed in transit (its integrity).

- User/entity Authentication
  - Allow a user/computer to prove his/her/its identity to another entity (e.g., a system, a device).

Entity Authentication

- Identification mechanisms are often divided into 3 types based on how the identity evidence is gathered
  - User knows a secret
    - Password, PIN, answers to prearranged questions
  - User possesses a token
    - these are normally hardware tokens such as magnetic-striped cards or custom-designed devices for time-variant passwords
  - User has a physical attribute
    - characteristics inherent to the user such as biometrics, handwritten signatures, keystroke dynamics, facial and hand geometries, voice, etc.

Next

- Definition of entity authentication
- Solutions
  - Password-based authentication
  - Token-based authentication
  - Biometric-based authentication
  - Stronger forms of secure authentication

Remote Authentication

- Now assume we want to use passwords for remote authentication
  - will it work?
- Passwords observed on the network are trivially susceptible to replay
  - initially remote login and file transfer programs, such as telnet, communicated passwords in the clear
  - now encryption is used (ssh, scp, etc.)
- Authentication based on time-invariant passwords is therefore a weak form of authentication
  - this form of authentication is nevertheless the most common
- A natural way to improve security is to use one-time passwords

One-Time Passwords (OTP)

- In authentication based on one-time passwords each password is used only once
- Such authentication can be realized in the following ways:
  - the user and the system initially agree on a sequence of passwords
    - simple solution but requires maintenance of the shared list
  - the user updates her password with each instance of the authentication protocol
    - e.g., the user might send the new password encrypted under a key derived from the current password
    - this method crucially relies on the correct communication of the new password to the system
      - attack: fishing website/links

Entity Authentication

- An even stronger form of authentication is one where the user doesn't have to send the secret to the verifier
  - ideally you want to convince the verifier without leaking information about your secret
  - such solutions exist and often involve the verifier sending a random challenge to the claimant
  - the claimant uses the challenge and the secret to compute the response
  - anyone who monitors the channel, cannot deduce information about the secret

Challenge-Response Techniques

Challenge-Response Techniques

- The goal of challenge-response techniques is to
  - use a single secret for authentication
  - provide evidence of the secret without leaking information about it
  - proving possession of a secret without leaking information about it is called a zero-knowledge proof of knowledge

- Challenge-response protocols can be built
  - from simple cryptographic primitives (e.g, MACs and signature schemes)
  - from scratch (Schnorr, Okamoto, and Guillou-Quisquater schemes)

Challenge-Response Techniques

- The basic form of such protocols is normally as follows:
  - suppose Alice is authenticating to Bob
  - Alice has a secret *s* and Bob has a verification value *v*
  - Bob sends to Alice a challenge *c* (chosen or computed anew)
  - Alice computes a response $r = f(s, c)$ and sends it to Bob
  - Bob verifies *r* using *c* and *v*
- Building a secure challenge-response protocol is non-trivial
  - must be secure against active adversaries
    - parallel session attack
    - man-in-the-middle attack

Authentication

- Definition of entity authentication
- Solutions
  - Password-based authentication
  - Token-based authentication
  - Biometric-based authentication
  - Stronger forms of secure authentication

Token-based Authentication

- Authentication based on what you possess can be done using different types of tokens
  - Types of authentication tokens
    - memory cards
    - smart tokens (hardware authentication tokens)
      - smart cards
        - electronic identity card (eID)
      - USB dongles
      - one-time password (OTP) device

Figure from: https://www.flickr.com/photos/lindseywb/3690885606/in/photostream/

Token-based Authentication

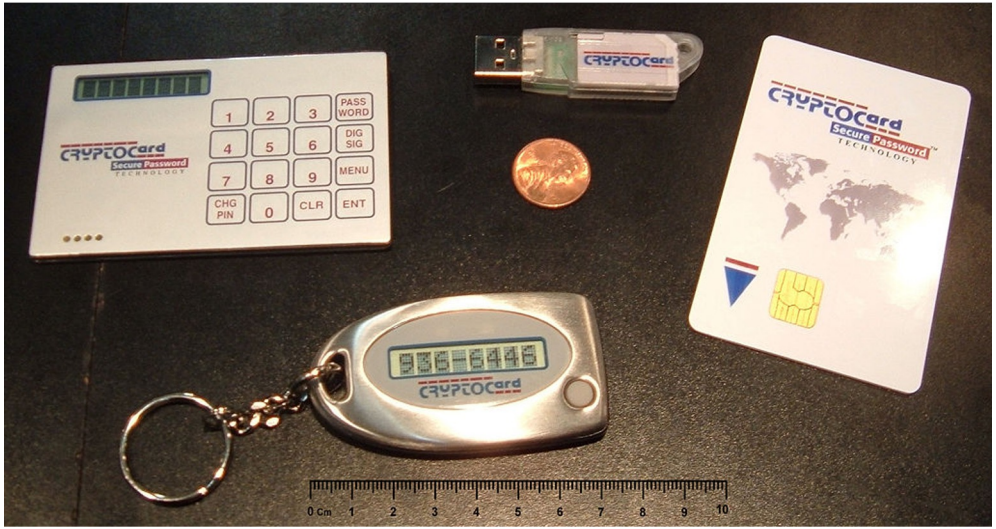
- memory cards
  - can store but do not process data
  - a card reader can retrieve information stored on the card
  - e.g., gift card, hotel keys
  - memory cards provide a limited level of security (i.e., card contents can be read by any reader and copied to another card)
  - memory cards are often combined with a password or PIN
  - using memory cards with computers requires special readers

Token-based Authentication

- smart tokens (hardware authentication tokens)
  - such tokens have a built-in microprocessor, programmable read-only memory and random-access memory (RAM)
  - they can engage in different types of authentication protocols including challenge-response
  - such tokens can also be used to generate dynamic passwords
    - each minute the device generates a new password
    - the device and the verifier must be synchronized
  - tamper-resistance of such tokens must be addressed
    - it’s been shown in the past that key material can be recovered with relatively inexpensive equipment

Token-based Authentication

- smart tokens examples

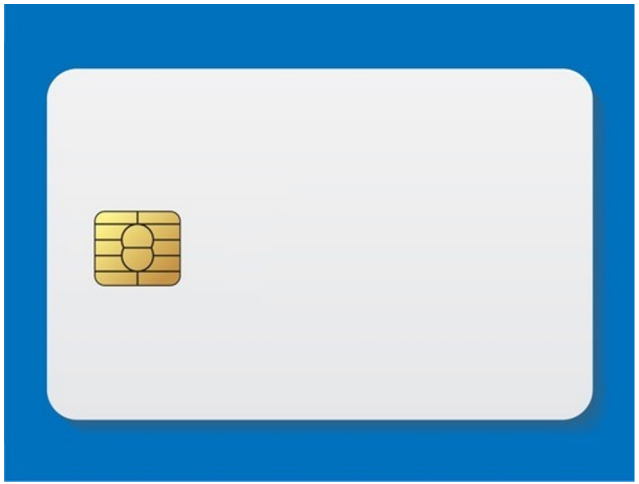


Left: Calculator-style card allows secure entry of a Personal Identification Number (PIN). **Top:** USB token. **Right:** Smart card format
Bottom: Key chain fob with onetime passcode displayed.

https://commons.wikimedia.org/wiki/File:SecurityTokens.CryptoCard.agr.jpg

Token-based Authentication

- smart tokens examples
  - smart cards
    - Most important category of smart token
      - Looks like a credit card
      - Has an electronic interface
      - May use any of the smart token protocols
    - Contain:
      - An entire microprocessor
      - Processor
      - Memory
    - What is in here?
      - I/O ports

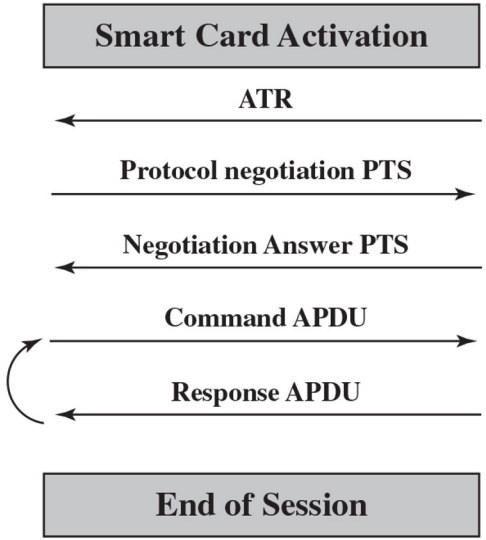


Token-based Authentication

- smart tokens examples
  - smart card (cont.)
    - reader exchanger
  - Examples: electronic identity cards (eID)
    - National ID
      - verified by the national government
      - stronger proof of identity?
    - UCCS student eID

Smart card

Card reader



APDU = Application protocol data unit
ATR = Answer to reset
PTS = Protocol type selection

Token-based Authentication

- smart tokens examples
  - USB dongle
    - USB tokens can also be used for authentication
    - they can store static data as well as code
      - recent dongles also include non-volatile memory
    - no additional hardware such a special-purpose reader is necessary
    - USB dongles are commonly used for copy protection of copyrighted material
    - dongle products often don’t provide enough security to be used in rigid security requirement environments

Token-based Authentication

- smart tokens examples
  - One-time password (OTP) device:
    - Has a secret key to generate an OTP
    - User enters the OTP and the system validates the value entered
    - Uses a block cipher/hash function to combine secret key and time or nonce value to create OTP
    - Has a tamper-resistant module for secure storage of the secret key

Token-based Authentication

- smart tokens examples
  - One-time password (OTP) device (cont.):
    - Uses HMAC with a hash function
    - Used in many hardware tokens and by many mobile authenticator apps
    - Password is computed from the current Unix format time value
  - Systems using time based OTP need to allow for clock drift between token and verifying system
  - Systems using nonce need to allow for failed authentication attempts

Token-based Authentication

- Smart tokens (hardware authentication tokens)
  - Disadvantage: any other person can see the code
    - Alternative: use of a communications link
  - Single-factor vs. multifactor:
    - Single-factor: provides authentication service with just one factor
    - Multifactor: provides authentication service after a local authentication step

Token-based Authentication

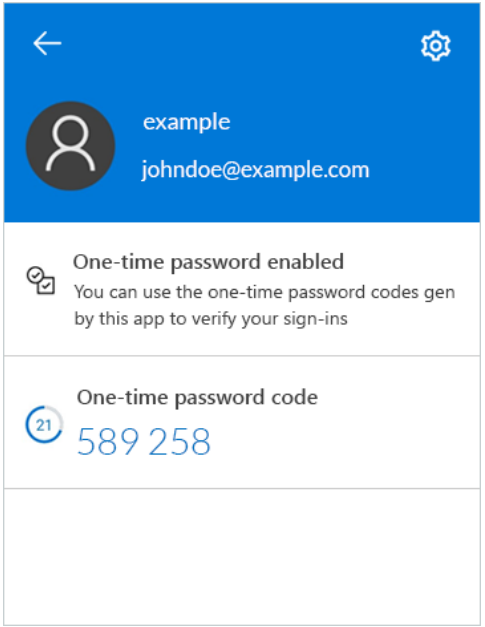
- Smart tokens (hardware authentication tokens)
  - FIDO2 (Fast IDentity Online 2)
    - Includes “WebAuthn” standard and “Client to Authenticator Protocol 2 (CTAP2)”
    - Uses a user agent as an intermediary between authenticator and authenticating service
    - Prominent members: Google and Microsoft

Authentication Using a Mobile Phone (1 of 2)

- Authentication code via message:
  - One of the simplest authentication approaches
  - Used for banking, government service access, etc.
  - No need to have any additional app on the phone
  - Disadvantage:
    - Requires mobile coverage to receive SMS
    - When mobile phone is lost or stolen, user will lose access or an attacker might gain access
    - Attackers might use a SIM swap attack
    - Attacker might also intercept messages using either a fake mobile tower, or by attacking S S 7 signaling protocol

Authentication Using a Mobile Phone (2 of 2)

- Mobile authentication apps:
  - Implements a one-time password generator
  - Implements the “Time-based one-time password (TOTP)” algorithm
  - Does not require a network connection
  - Can be used with multiple accounts
  - More secure than authentication code
  - Disadvantages:
    - Phone might be lost or stolen
    - Attacker might compromise by installing malware
    - Attacker might convince the user to reveal secret code

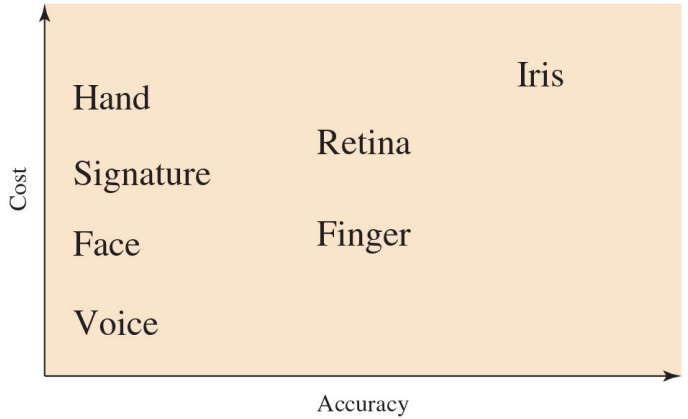


Authentication

- Definition of entity authentication
- Solutions
  - Password-based authentication
  - Token-based authentication
  - Biometric-based authentication
  - Stronger forms of secure authentication

Biometric Authentication

- Biometric authentication systems authenticate an individual based her physical characteristic
- Types of biometric used in authentication
  - face
  - palm geometry
  - fingerprint
  - iris
  - signature
  - voice
- Most common uses of biometric authentication is for specific applications rather than computer authentication



Cost versus accuracy of various biometric characteristics in user authentication schemes

Biometric Authentication

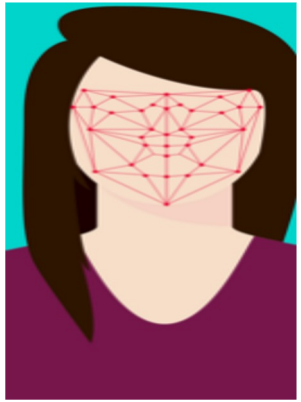
- Like other authentication mechanisms, biometric authentication includes an enrollment phase during which a biometric is captured
  - the initial reading is often called a template
  - at authentication time, a new biometric reading is performed and is compared to the stored template
- Unlike other authentication mechanisms, biometric matching is approximate
  - each reading can be influenced by a variety of factors
    - e.g., light conditions, facial expressions, hair style, glasses, etc. for face recognition
  - some types of biometrics can match more accurately than others
    - e.g., iris vs. face or palm

Biometric Authentication

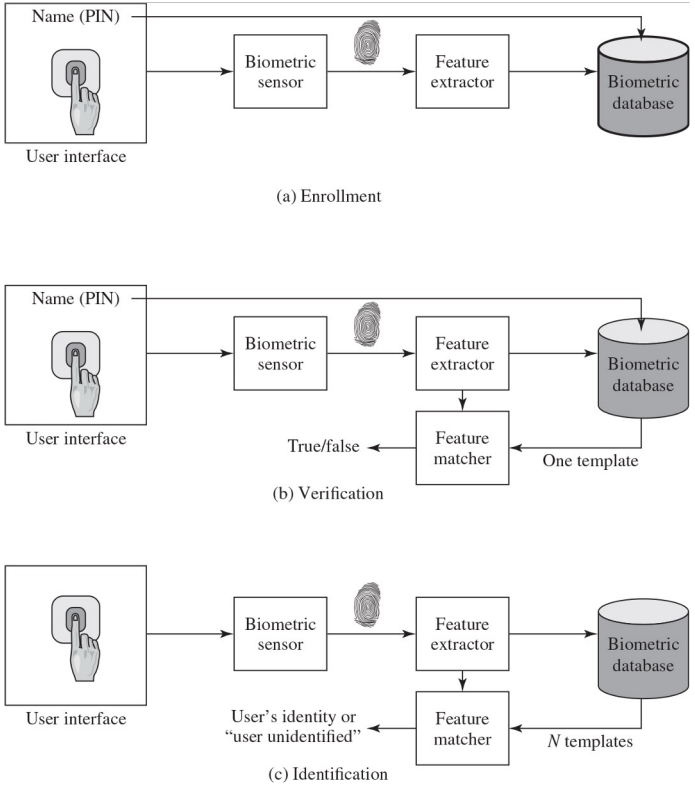
- Biometric matching can be used to perform
  - verification
    - user's biometric scan is used to match her own template only
  - identification
    - user's biometric scan is used to match a database of templates
- Identification might not always be possible
- Biometric systems attempt to minimize
  - false reject rate: authentic biometric is rejected
  - false accept rate: imposter biometric is accepted
- Depending on the environment, minimizing one of them might be more important than minimizing both

A Generic Biometric System

Feature Extraction for Template



(10)



Why use PIN?

Biometric Authentication

- New types of biometrics are being explored
  - brain waves, heart beats, etc.
- Many forms of traditional biometrics can be stolen
- Static biometrics can be replayed

Biometric Authentication

- Current research direction: biometric key generation
  - the idea: a biometric can be used to generate a cryptographic key
  - the key can be reproduced using another biometric close enough to the original
    - no need to remember any information such as a password
  - the key can be used for authentication or encryption
  - key generation algorithm produces a helper data that can later aid in recovering the same key from a noisy version of the biometric
  - security requirements are strict
    - the helper data must leak minimal information about the biometric
    - compromise of the key must not lead to recovery of the biometric

Summary

- Entity authentication is an important topic with the main application in access control
- Various techniques exist ranging from time-invariant passwords to provably secure identification schemes
- Despite the weak security password-base authentication provides, it is the most widely used authentication mechanism
  - ease of use, user familiarity, no infrastructure requirements
- Next time
  - access control mechanisms