CS 4910: Intro to Computer Security

Authentication

Instructor: Xi Tan

Updates

- Assignment 1 due 2/12
- Project 1 due 2/24

What we already know

- Cryptography tools
  - Symmetric and public-key encryption
  - Message integrity
    - MAC
    - Hash functions
  - Digital certificates and random numbers

- Next: intro to authentication

Outline

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

- Authentication is a broad term and is normally referred to mechanisms of ensuring that
  - entities are who they claim to be
  - data has not been manipulated by unauthorized parties
- Entity authentication or identification refers to the means of verifying user identity
  - if such verification is successful, the user is granted appropriate privileges
- The need for user authentication in early computer systems arose once it became possible to support multi-user environments

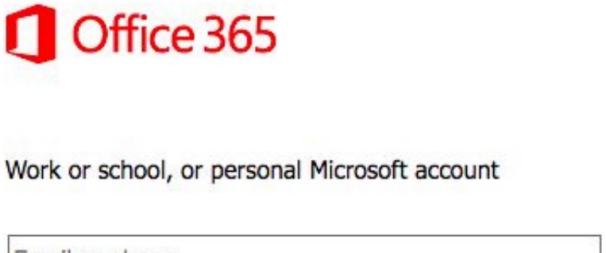
Entity Authentication

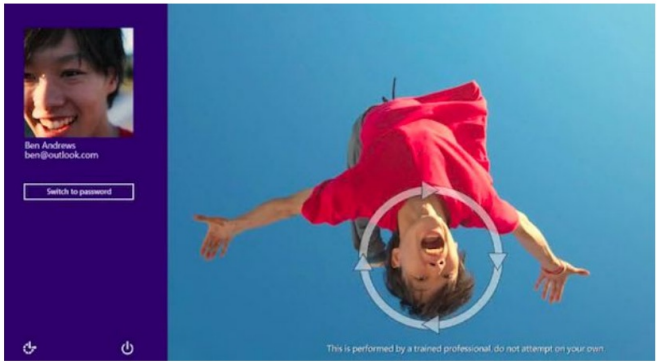
- During an authentication protocol:
  - one party, the verifier, gathers evidence that the identity of another party, the claimant, is as claimed
- Goals of authentication protocols:
  - honest parties should be able to successfully finish the protocol with their identity accepted as authentic
  - it should be difficult for dishonest parties to impersonate an identity of another user
  - impersonation must remain difficult even after observing a large number of successful authentications by other parties
- User registration is required prior to an authentication protocol

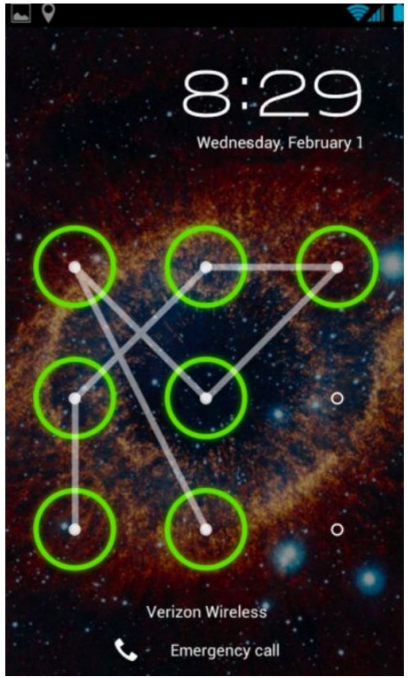
Entity Authentication

- 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.

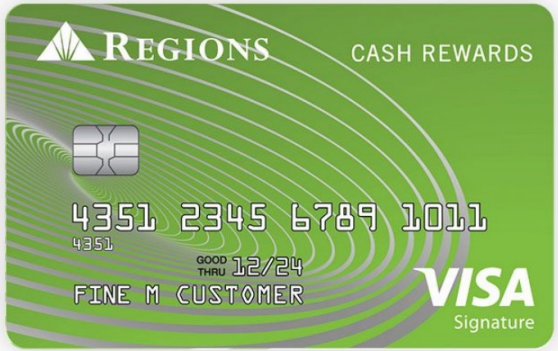
User Knows a Secret



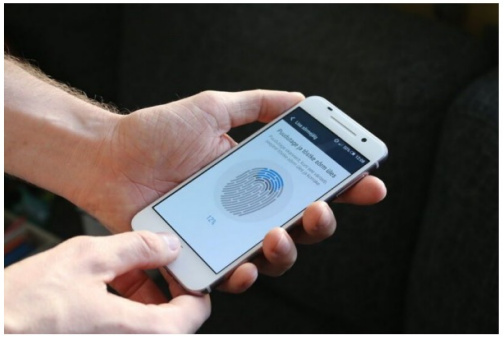


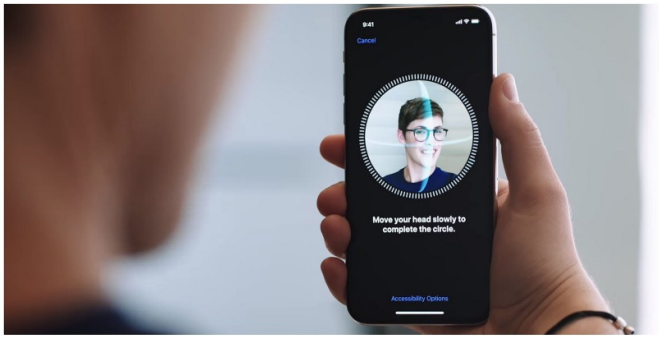


User Possesses a Token

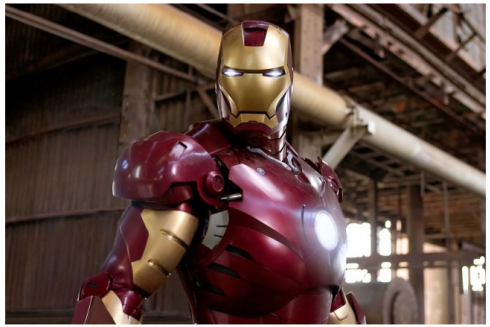


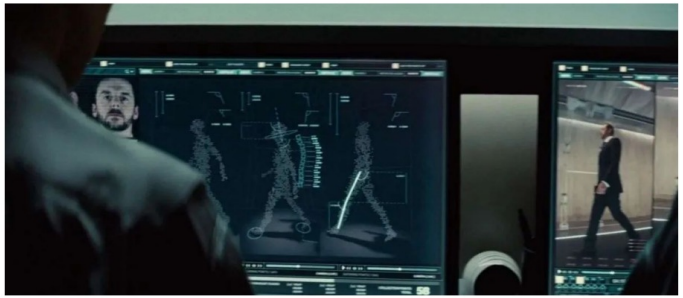
User Has a Physical Attribute (biometrics)

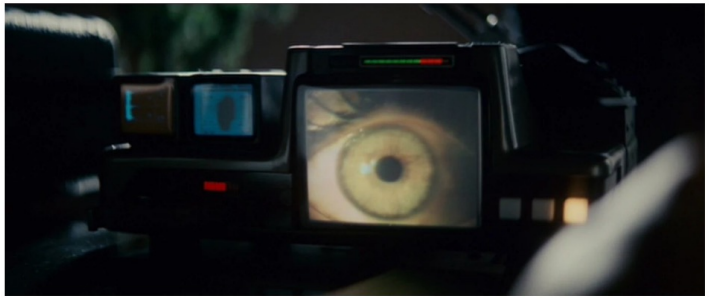




User Has a Physical Attribute (biometrics)







Entity Authentication

- Often, different types can be combined together
  - e.g., PIN-based authentication is often used with a physical device (user ID, credit card)
  - biometric-based authentication is often used in combination with a password or a physical token
- Many identification mechanisms used in practice are not secure
  - calling cards
  - credit card purchases
  - passwords
- Ideally, we want solutions against which replay attacks don’t work

Password-based Authentication

- A password is a string of (normally 8 or more) characters associated with a certain user
  - it serves the purpose of a shared secret between the user and the system
- During the identification protocol:
  - a user sends (userid, password) pair
    - userid identifies the user
    - password provides the necessary evidence that the user possesses the secret
  - the system compares that information with its has stored
  - if the check succeeds, access is granted

Password-based Authentication

- Storage of passwords
  - the most straightforward way of storing passwords is in clear text
    - there is a problem with such approach
  - to mitigate it, most systems apply a one-way hash function to a password and store the hash
    - the password itself cannot be recovered, but there are other concerns

Password-based Authentication

- Attacks on passwords
  - replay of passwords: an attacker reuses a captured password
    - an attacker can capture a password by seeing a user type it, using a keylogger program or obtaining it in transit
  - rainbow table attacks: an attacker generates a table of hash values
    - pre-compute tables of hash values
    - a mammoth table of hash values
    - can be countered by using a sufficiently large salt value and a sufficiently large hash length

Password-based Authentication

- Attacks on passwords (cont.)
  - exhaustive search: an attacker attempts to guess a user password by trying all possible strings
    - this can be done on the verifier itself or by obtaining a copy of the password file and performing the attack off-line
    - often the attack is infeasible if the password space is large enough
      - but it is still possible to exhaust all short passwords
  - dictionary attack: an attacker tries to guess a password using words from a dictionary and variations thereof
    - can have a high probability of success
    - dictionary attacks become increasingly sophisticated

Password-based Authentication

- Is there a way to decrease the vulnerability of the system to such attacks?
- Additional measures are normally employed, some of which are:
  - salting passwords
    - this technique makes guessing attacks less effective
    - a password is augmented with a random string, called salt, prior to hashing
    - the salt is stored in cleartext in the password file

$$uid_1, salt_1, h(salt_1||pwd_1)$$

$$uid_2, salt_2, h(salt_2||pwd_2)$$

- how does it improve security?

Password-based Authentication

- Measures for improving security of passwords (cont.)
  - slowing down password verification
    - the hash function for password verification is made more computationally extensive
    - this can be done, e.g., by iterating the computation n times
    - what is its drawback?
  - limiting the number of unsuccessful password guesses
    - a user account is locked after the number of successive unsuccessful authentication attempts exceeds the threshold
  - employing password rules
    - additional rules on password choices are imposed
    - this often strengthens password choices but limits the search space

Password-based Authentication

- Measures for improving security of passwords (cont.)
  - preventing direct access to password file
    - the file/database with hashed passwords is kept inaccessible by ordinary users

- Another technique that aims at improving security of passwords is called password aging (enforce the regular changing of passwords)

- It is always a challenge to find a balance between memorability of passwords and their resistance to dictionary attacks
  - do users make acceptable password choices?
  - can we help them with choosing strong passwords?

Password-based Authentication

- Password strength has been studied since 1990s
  - a significant portion of used passwords is guessable
    - passwords of short length can be cracked using brute force search
    - account-related or dictionary-derived passwords are common
  - password crackers today are increasingly complex

- How can we help users to select stronger passwords?
  - systems are much better at helping users than before
  - a variety of tools exist

Password-based Authentication

- User-chosen secrets
- Suppose passwords could be up to 9 characters long
- This would produce 10^18 possible passwords
- 320,000 years to try them all at 10 million a second!

- Unfortunately, not all passwords are equally likely to be used
- Users have the tendency to choose easy to remember but weak password

Password-based Authentication

The Security of Modern Password Expiration: An Algorithmic Framework and Empirical Analysis

Yinqian Zhang
University of North Carolina at Chapel Hill
Chapel Hill, NC
yinqian@cs.unc.edu

Fabian Monrose
University of North Carolina at Chapel Hill
Chapel Hill, NC
fabian@cs.unc.edu

Michael K. Reiter
University of North Carolina at Chapel Hill
Chapel Hill, NC
reiter@cs.unc.edu

Password-based Authentication

- Tools for choosing stronger passwords
  - computer-generated passwords
    - selecting less predictable passwords which users can remember can be done by using computer-generated pronounceable passwords
    - e.g.: heloberi, hoparmah, ulensoev, atonitim
  - password checking
    - a proactive password checker rates password strength at the time of password selection
  - Reactive password checking
    - System periodically runs its own password cracker to find guessable passwords
  - other types of passwords
    - techniques for using images and graphical interfaces for authentication have been developed

Password-based Authentication

- Tools for choosing stronger passwords (cont.)
  - image-based passwords and graphical interfaces
    - displaying a sequence of images
    - drawing patterns on a grid
    - choosing points using an image
    - their unpredictability is often not as great as desired

- Unpredictability and usability of passwords is hard to achieve simultaneously
  - passwords can provide only a weak form of security

Best Password Practices

- NIST's Special Publication 800-63 provides authentication guidelines for organizations including password-based authentication
  - the latest version is dated by June 2017

- In general, you want to
  - use strong passwords
  - not reuse passwords across different services
  - not share your passwords with anyone else

- Password managers are of great help in dealing with password explosion
  - e.g., 1password

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

Next

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