CSC12001

Data Security in Information Systems

C02 - User Authentication

Dr. Phạm Thị Bạch Huệ
MSc. Lương Vĩ Minh

Information System Department – Faculty of Information Technology
Univeristy of Science, VNU-HCM

What is authentication?

- Positive verification of identity (man or machine)
- Verification of a person's claimed identity
- Who are you? Prove it.

Basics

- There exists two reasons for authenticating users:
  - The user identity is a parameter in access control decisions
  - The user identity is recorded when logging security-relevant events in the audit trail
- It is not always necessary or desirable to base access control on user identities, while there is a much stronger case for using identities in the audit logs

Basics

- When a user connects to a computer system has to enter
  - *Username* – this step is called *identification*
  - *Password* – this step is called *authentication*
- Authentication: the process of verifying a claimed identity

How to authenticate?

- 4 categories:
  - What you know
  - What you have
  - Who you are
  - Where you are

Authentication Process

- It consists of several steps:
  - Obtaining the authentication information from an entity
  - Analyzing the data
  - Determining if the authentication information is associated with that entity

What you know

- Password
- Passphrase
- PIN

Passwords

- Sequence of characters
  - Examples: 10 digits, a string of letters, *etc*.
  - Generated randomly, by user, by computer with user input
- Sequence of words
  - Examples: pass-phrases

  Note: A *pass-phrase* is a sequence of characters that it is too long to be a password and it is thus turned into a shorter virtual password by the password system
- Algorithms
  - Examples: challenge-response, one-time passwords

Passwords-based Authentication

- A *password* is information associated with an entity that confirms its identity.
- How can passwords be protected?
- A solution: *one-way hashing*

A user's password is encrypted and then stored. The stored password is never decrypted.

It should be difficult for an attacker to revert the stored password to the plaintext password.

A user A may try to guess the password of another user, B, and thus *impersonate* B.

Storage

- Store as cleartext
  - If password file compromised, *all* passwords are revealed
- Encipher file
  - Need to have encryption, decryption keys in memory
  - Reduces to previous problem
- Store one-way hash of password
  - If file read, attacker must still guess passwords or invert the hash

Salting

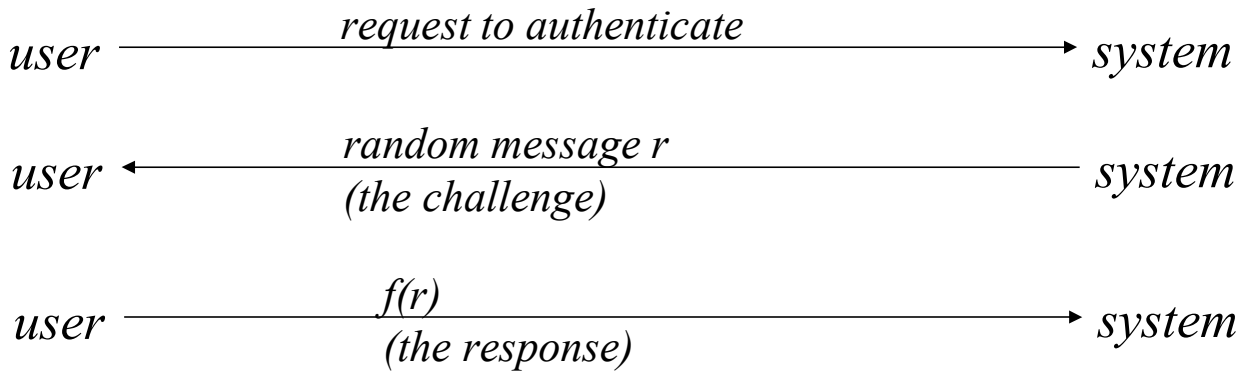
- Goal: slow dictionary attacks aimed at finding *any* user's password (as opposed to a *particular* user's password)
- Method: perturb hash function so that:
  - Parameter controls *which* hash function is used
  - Parameter differs for each password
  - To determine if the string *s* is the password for any of a set of *n* users, the attacker has to perform *n* complementations, each of which generates a different complement

Challenge-Response

- Passwords have the fundamental problems that they are *reusable*
- If an attacker sees a password, she can later *replay* the password
- An alternative is to authenticate in such a way that the transmitted password changes each time
- Let a user *u* wishing to authenticate himself to a system *S*.
  - Let *u* and *S* have an agreed-on secret function *f*.
  - A *challenge-response* authentication system is one in which *S* sends a random message *m* (the challenge) to *u*
  - Then *u* replies with the transformation $r = f(m)$ (the response).
  - *S* then validates *r* by computing it separately.

Challenge-Response

- The user and system share a secret function *f* (in practice, *f* can be a known function with unknown parameters, such as a cryptographic key)



Challenge-Response
Pass Algorithms

- Challenge-response with the function *f* itself a secret
  - Example:
    - Challenge is a random string of characters such as “a**b**c**d**e**f**g”, “a**g**e**k**s**i**d**o**”
    - Response is some function of that string such as “bdf”, “gkio”
    - The algorithm is every other letter beginning with the second
  - Usually used in conjunction with fixed, reusable password

One-Time Passwords

- Password that can be used exactly *once*
  - After use, it is immediately invalidated
- Problems
  - Synchronization of user and system
  - Generation of good random passwords
  - Password distribution problem

Approaches: Password Selection

- Random selection
  - Any password from *A* equally likely to be selected
  - Such passwords are difficult to remember for users, especially when they have multiple randomly-selected passwords
- Pronounceable passwords
- User selection of passwords

Pronounceable Passwords

- Generate phonemes randomly
  - Phoneme is unit of sound, eg. *cv*, *vc*, *cvc*, *vcv* where
    - *c* is a consonant
    - *v* is a vowel
  - Examples: helgoret, juttelon are pronounceable; przbqxdfl, zxrptglfn are not pronounceable
- Problem: the number of pronounceable passwords of length *n* is considerably lower than the number of random passwords of length *n*

User Selection

- Problem: people pick easy to guess passwords
  - Based on account names, user names, computer names, place names
  - Dictionary words
  - Too short, digits only, letters only
  - License plates, acronyms, social security numbers
  - Personal characteristics or foibles (pet names, nicknames, job characteristics, *etc*.)

Selecting Good Passwords

- Good passwords can be constructed in several ways
  - A password containing at least one digit, one letter, one punctuation symbol, and one control character is usually a strong password
- “LIMm*2^Ap”
  - Letters chosen from the names of members of 2 families

PIN

- Consider the case of a 4-digit PIN
- Suppose that the number of possible passwords (PINs) is $N=10^4$ (assuming that the digits 0-9 are allowed in each of the 4 positions of the PIN
- Assume that an attacker can make $G=10{,}000$ per second in an offline attack
- How long would it take to guess a PIN with absolute certainty?

What you have

- Digital authentication
  - physical devices to aid authentication
- Common examples:
  - eToken
  - smart cards
  - RFID

eToken

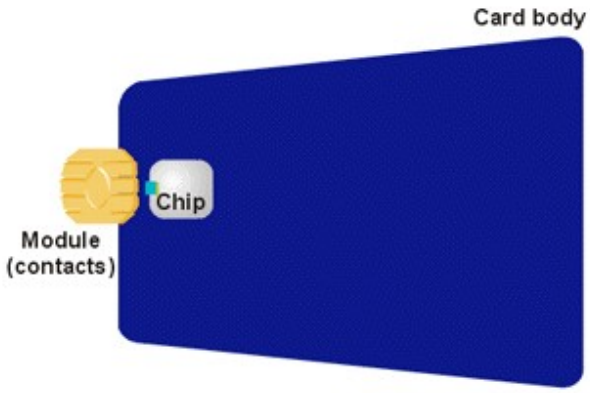
- Can be implemented on a USB key fob or a smart card
- Data physically protected on the device itself
- On the client side, the token is accessed via password
- Successful client-side authentication with the password invokes the token to generate a stored or generated **passcode**, which is sent to the server-side for authentication.

eToken

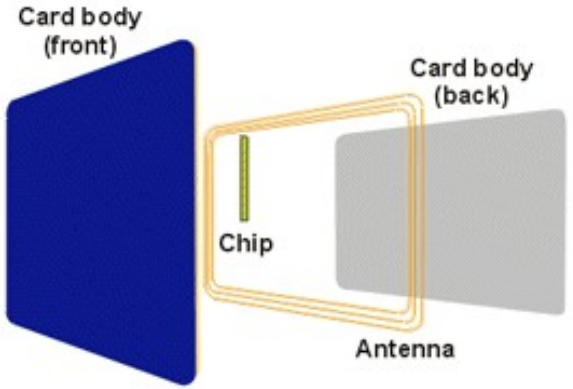
- May store credentials such as passwords, digital signatures and certificates, and private keys
- Can offer on-board authentication and digital signing

Smart cards

- Size of a credit card
- Usually an embedded microprocessor with computational and storage capabilities
- Contact vs. contactless
- Memory vs. microprocessor



Source: Gemplus - All About Smart Cards



Source: Gemplus - All About Smart Cards

RFID

- RFID - Radio Frequency IDentification
- Integrated circuit(s) with an antenna that can respond to an RF signal with identity information
- No power supply necessary—IC uses the RF signal to power itself
- Susceptible to replay attacks and theft
- Examples:
  - Smart Tag, EZPass
  - Garage parking permits

Who you are

- Biometric authentication
  - Use of a biometric reading to confirm that a person is who he/she claims to be
- Biometric reading
  - A recording of some physical or behavioral attribute of a person

Physical Biometrics

- Fingerprint
- Iris
- Hand Geometry
- Finger Geometry
- Face Geometry
- Ear Shape
- Retina
- Smell
- Thermal Face
- Hand Vein
- Nail Bed
- DNA
- Palm Print

Behavioral Biometrics

- Signature
- Voice
- Keystroke
- Gait

Fingerprints

- Vast amount of data available on fingerprint pattern matching
- Data originally from forensics
- Over 100 years of data to draw on
  - Thus far all prints obtained have been unique

Fingerprint Basics

- Global features
  - Features that can be seen with the naked eye
  - Basic ridge patterns
- Local features
  - Minutia points
  - Tiny unique characteristics of fingerprint ridges used for positive identification

Basic Ridge Patterns

- **Loop**
  - 65% of all fingerprints

- **Arch**
  - Plain and tented arch

- **Whorl**
  - 30% of all fingerprints
  - One complete circle

Local Features

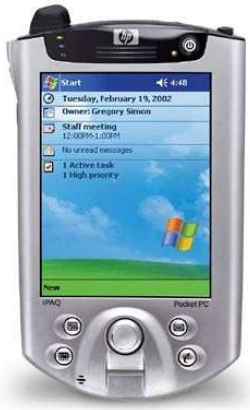
- Also known as *minutia points*
- Used for positive identification
- Two or more individuals may have the same global features, but different minutia
- Minutia points do not have to be inside the pattern area

Types of Minutia

- Ridge ending
- Ridge bifurcation
- Ridge divergence
- Dot or island – ridge so short it appears t
- Enclosure – ridge separates and then reunites around an area of ridge-less skin
- Short ridge – bigger than a dot

Fingerprint Scanners

Digital Persona U.are.U Pro



HP IPAQ

IBM Thinkpad T42

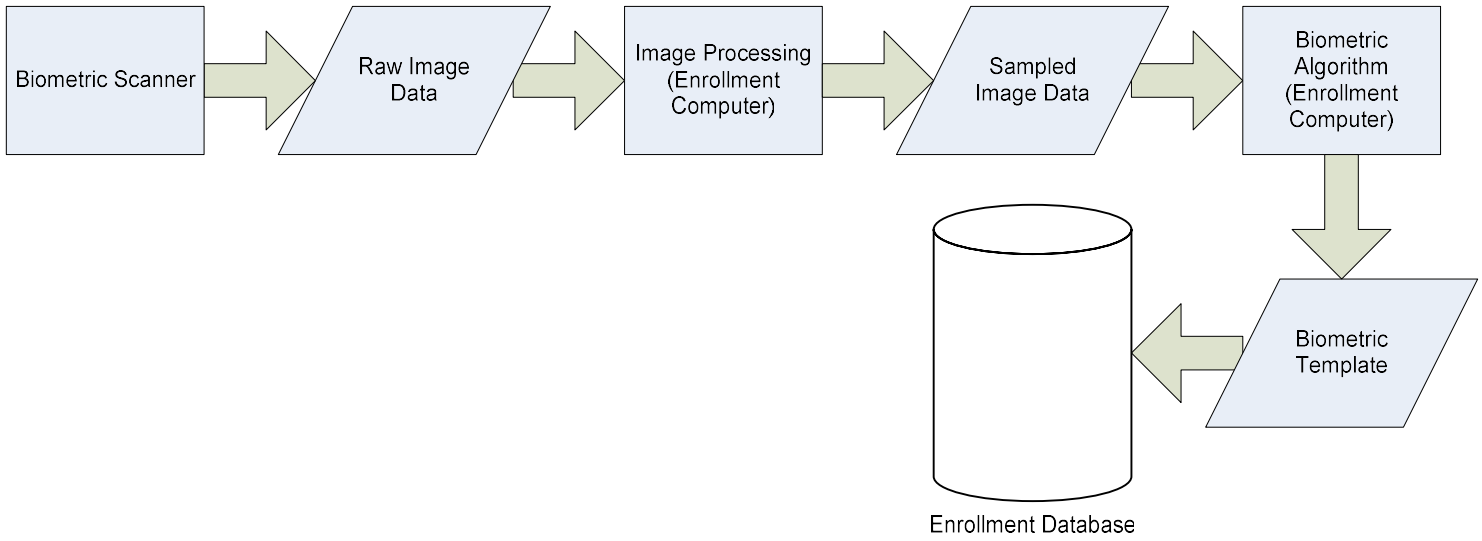
Where you are

- If you know where user is, validate identity by seeing if person is where the user is
  - Requires special-purpose hardware to locate user
    - GPS (global positioning system) device gives location signature of entity
    - Host uses LSS (location signature sensor) to get signature for entity

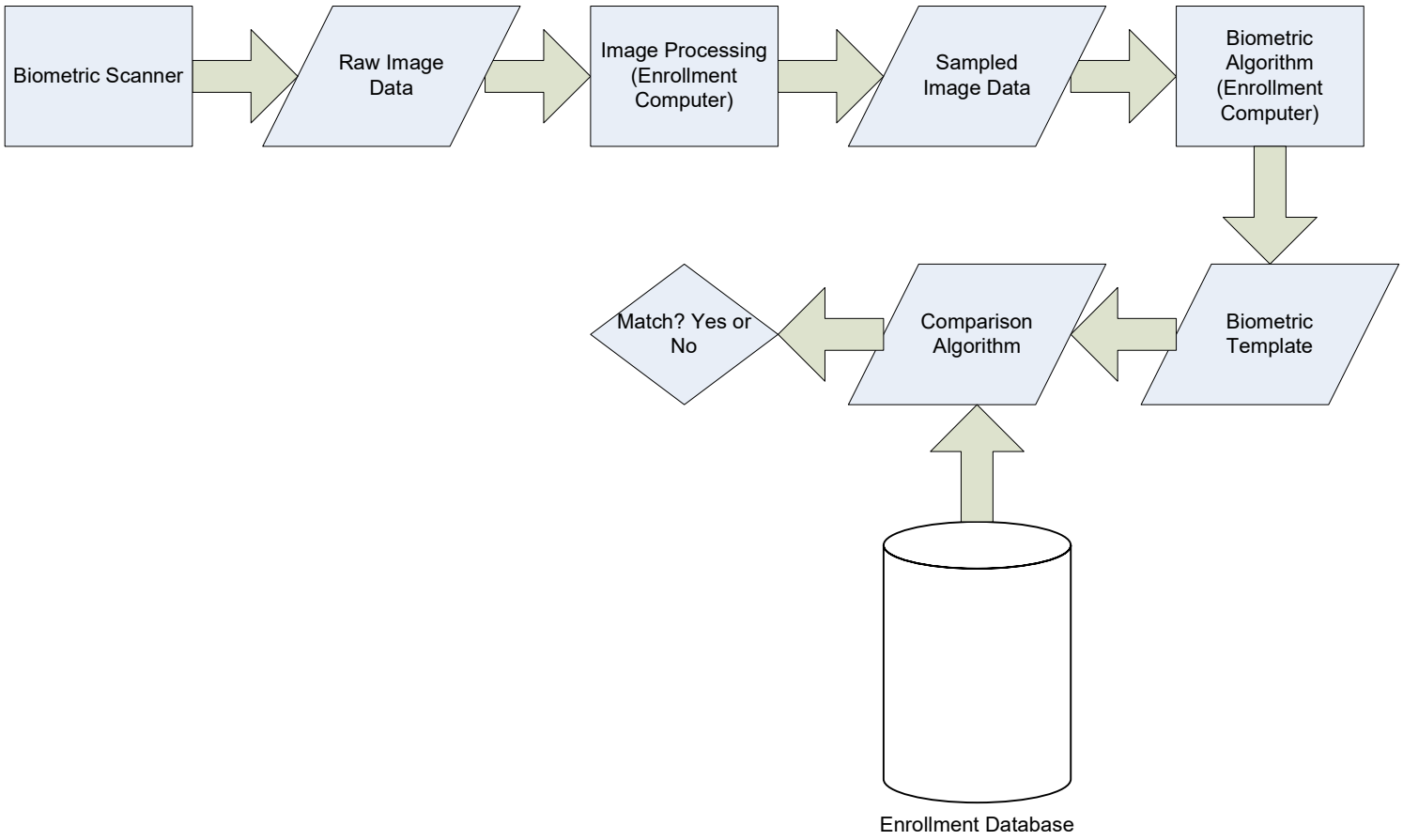
Review: Four Categories

- What you know
  - Password
  - PIN
- What you have
  - e-Token
  - RFID
- Who you are
  - Biometrics
- Where you are
  - Location

Example - Enrollment



Example - Verification



Motivation

- Real-world considerations:
  - What you know and what you have
    - Can be stolen or forgotten
    - Susceptible to replay attacks
  - Who you are
    - Unique biometrics that hinder replay attacks and imposters
    - Privacy issues arise

Multiple Methods

- Example: “where you are” also requires entity to have LSS and GPS, so also “what you have”
- Can assign different methods to different tasks
  - As users perform more and more sensitive tasks, must authenticate in more and more ways (presumably, more stringently) File describes authentication required
    - Also includes controls on access (time of day, *etc*.), resources, and requests to change passwords
  - Pluggable Authentication Modules

Bibliography

- Authentication
  - L. O'Gorman, "Comparing Passwords, Tokens, and Biometrics for User Authentication," Proc. IEEE, Vol. 91, No. 12, Dec. 2003, pp. 2019-2040.
- Class slides of Elisa Bertino, Paul Bui.

Q&A

Dr. Phạm Thị Bạch Huệ - ptbhue@fit.hcmus.edu.vn

MSc. Lương Vĩ Minh - lvminh@fit.hcmus.edu.vn