User Authentication

Introduction to Computer Security

Week 8

Topics in User Authentication

- Electronic user authentication principles
- Password-based authentication
- Token-based authentication
- Biometric authentication
- Remote user authentication
- Security issues for user authentication

Learning objectives

- Discuss the four general means of authenticating a user's identity
- Explain the mechanism by which hashed passwords used for user authentication
- Understand the use of the Bloom filters in password management
- Present an overview of token-based user authentication
- Discuss the issues involved and the approaches for remote user authentication

User Authentication

- User authentication is basis of access control and user accountability
- The process of verifying an identity claimed by or for a system entity
- Two steps
  - identification: specify identifier
  - verification: bind entity (person) and identifier
- Distinct from message authentication

  (which is concerned with the integrity of messages)

Means of User Authentication

- Four means of authenticating user's identity

Means of User Authentication

- Four means of authenticating user's identity
- Based on something you
  - know: e.g., password, PIN

Means of User Authentication

- Four means of authenticating user's identity
- Based on something you
  - know: e.g., password, PIN
  - possess: e.g., token, smartcard, password device

Means of User Authentication

- Four means of authenticating user's identity
- Based on something you
  - know: e.g., password, PIN
  - possess: e.g., token, smartcard, password device
  - are (static biometrics): e.g. fingerprint, retina

Means of User Authentication

- Four means of authenticating user's identity
- Based on something you
  - know: e.g., password, PIN
  - possess: e.g., token, smartcard, password device
  - are (static biometrics): e.g. fingerprint, retina
  - do (dynamic biometrics): e.g. voice, signature

Means of User Authentication

- Four means of authenticating user's identity
- Based on something you
  - know: e.g., password, PIN
  - possess: e.g., token, smartcard, password device
  - are (static biometrics): e.g. fingerprint, retina
  - do (dynamic biometrics): e.g. voice, signature
- Can be used alone or combined

Password authentication

- Widely used user authentication method
  - user provides name and password
  - system compares password with that saved for specified login
- Authenticates ID of user logging and
  - that the user is authorized to access system
  - determines the user's privileges
  - is used in discretionary access control

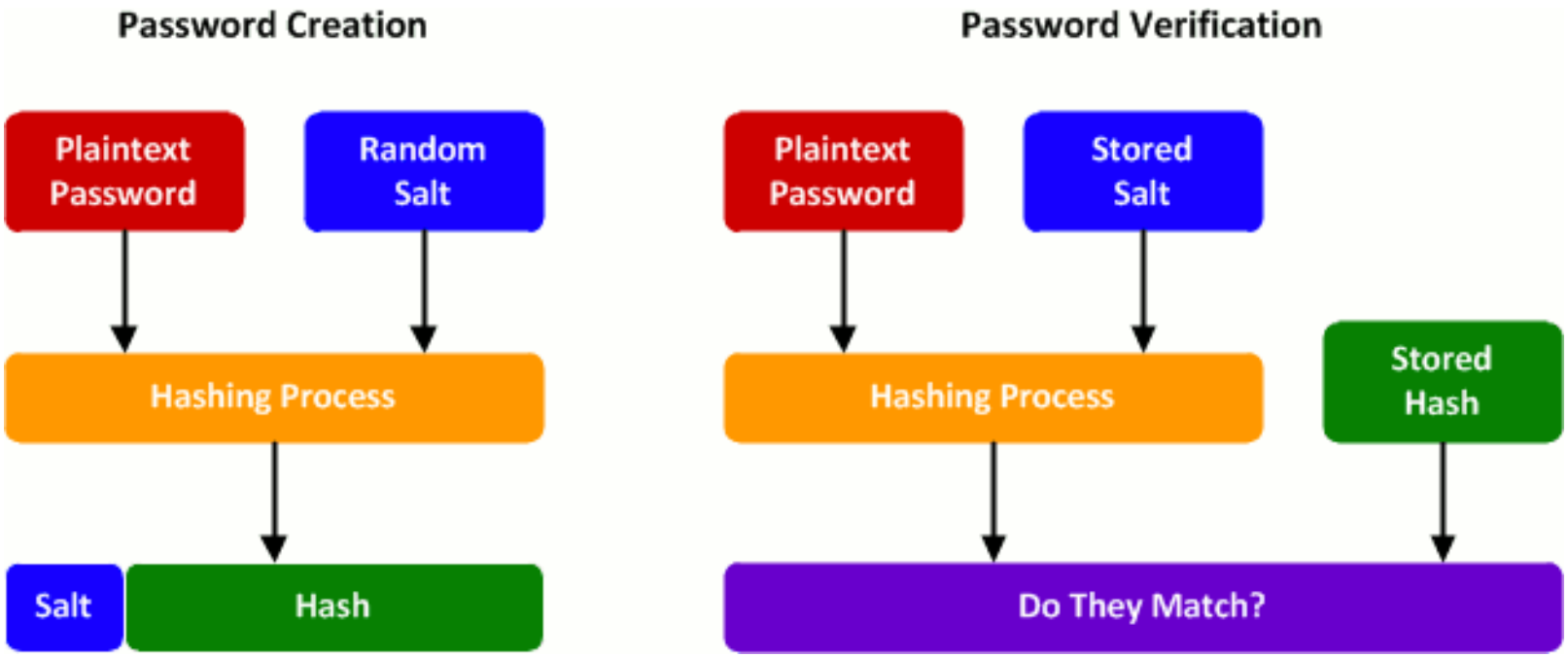
Password vulnerabilities

- offline dictionary attack (with e.g. `etc/passwd` available)
- password guessing against single user (w/ previous knowledge about the user, such that password may be related to username)
- popular password attack (try '123' or '123456', try common English names, etc.)
- exploiting user mistakes (e.g., users writing password down, multiple sites with the same password)
- workstation hijacking (attacker waits until a logged-in workstation is unattended)
- Electronic monitoring (network eavesdropping)

Countermeasures for password vulnerabilities

- prevent unauthorized access to `etc/passwd` file
- account lockout mechanisms (e.g., after three unsuccessful attempts)
- against password guessing: enforce password selection strategies
  - enforce computer generated password (often hard for users to remember)
  - reactive/proactive password checking (prevent passwords that do not satisfy certain criteria or can be easily guessed, and disallow them)
- workstation hijacking: automatic workstation logout
- against eavesdropping: encrypted network links

Hashed Password Scheme



Each entry in the UNIX `passwd` file stores three fields:
User Id, **Salt**, **Hashcode** (hashed password w Salt)

The Use of Salt

- Same password in different entries look different in hashcode (avoid duplicant password look the same)
- Increases the difficulty of offline dictionary attacks
- Nearly impossible to tell if a person used the same password on multiple systems

UNIX Implementation

- Original scheme
  - 8 character password form 56-bit key
  - 12-bit salt used to modify DES encryption into a one-way hash function
  - output translated to 11 character sequence
- Now regarded as woefully insecure
- Sometimes still used for compatibility

Improved implementations

- Have other, stronger, hash/salt variants
- Many systems now use MD5
  - with 48-bit salt
  - password length is unlimited
  - is hashed with 1000 times inner loop
  - produces 128-bit hash
- OpenBSD uses Blowfish block cipher based and hash algorithm called Bcrypt
  - uses 128-bit salt to create 192-bit hash value

Password Cracking

- Dictionary attacks
  - try each word then obvious variants in large dictionary against hash in password file
- Rainbow table attacks
  - a large dictionary of all possible passwords
  - for each password: precompute tables of hash values for all salts, resulting in a mammoth table of hash values: e.g. 1.4GB table cracks 99.9% of alphanumeric Windows passwords in 13.8 secs
  - not feasible if larger salt values used (using too much space for caching)

Password choices concerns

- Users may pick short passwords
  According to a study at Perdue University: 3% of all passwords are three characters or shorter
- Users may pick guessable passwords
  - same as account name: 2.7%
  - common English names: 4%
  - system dictionary words: 7.4%
- Crackers can use lists of likely passwords
- e.g. one study of 14000 encrypted passwords guessed nearly 1/4 of them

Password File Access Control

- Can block offline guessing attacks by denying access to encrypted passwords
  - make available only to privileged users
  - often using a separate shadow password for super users (`su`) only
- Still have vulnerabilities
  - exploit O/S bug
  - accident with permissions making it readable (human mistake)
  - users with same password on other systems
  - access from unprotected backup media
  - sniff passwords in unprotected network traffic

Proactive Password Checking Mechanisms

- Rule enforcement plus user advice, e.g.
  - length of at least 8 chars
  - must contain upper/lower/numeric/punctuation
- Password cracker
  - list of bad passwords
  - time and space issues (system resource consumption)
- Markov Model
  - generates guessable passwords
  - hence reject any password it might generate
- Bloom Filter
  - use to build table based on dictionary using hashes
  - check desired password against this table

Token-based authentication

- what a user possesses to authenticate, e.g.
  - memory card (barcode, magnetic stripe, RFID)
  - smartcard
  - password devices
  - USB key

Memory Card

- store but do not process data
- barcode, magnetic stripe card (e.g. bank card)
- can be of more complex forms (smart card, passive RFID card)
- sometimes used alone for physical access
  (e.g., as hotel room keycards, or company cards for restricted area access)
- some with password/PIN (e.g., ATMs)
- Drawbacks
  - need special reader
  - loss of token
  - user dissatisfaction (easy for ATM, not OK for computer access)

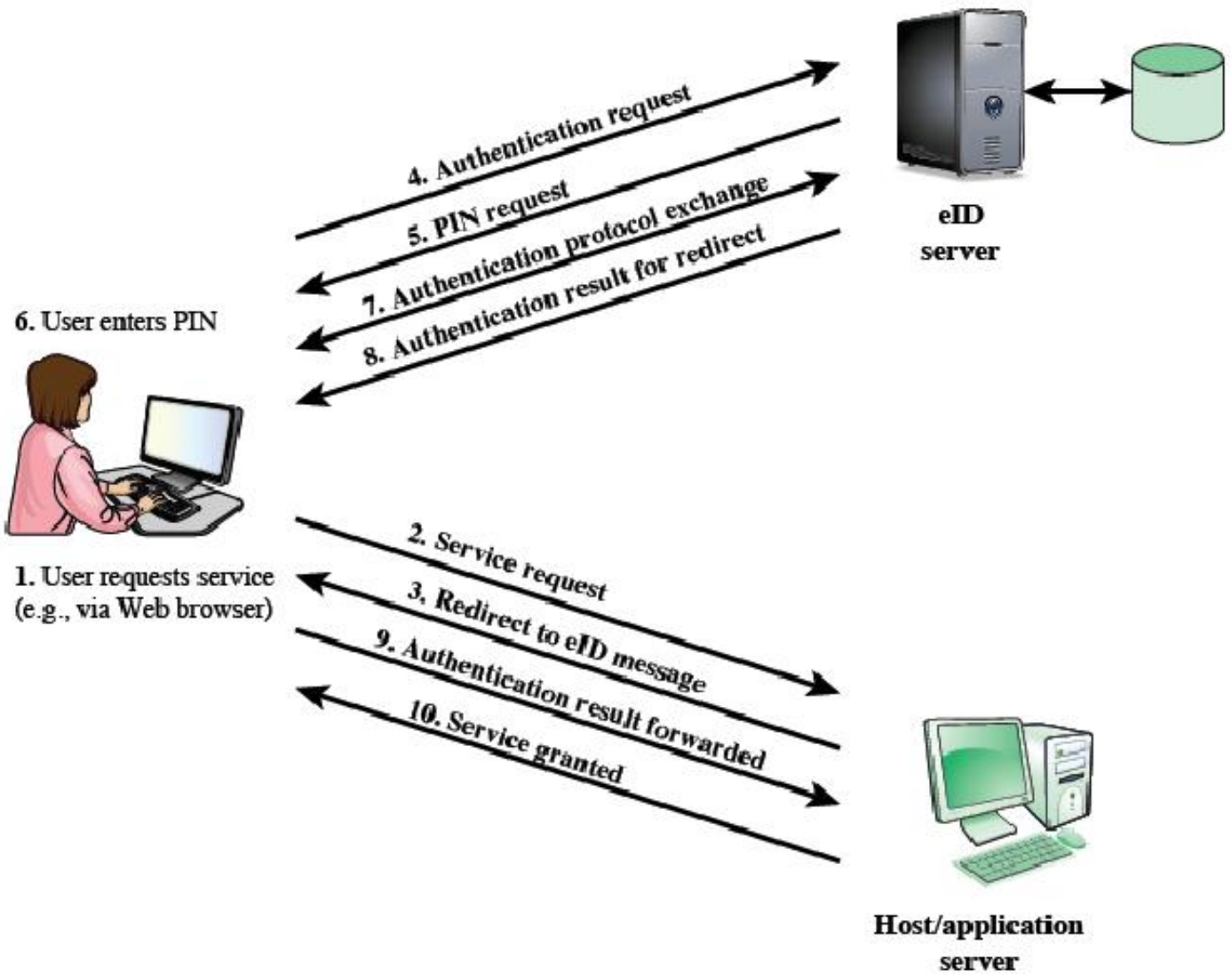
Smartcard

- credit-card like
- has own processor, memory, I/O ports (ROM, EEPROM, RAM memory)
  - ROM holds read-only data, such as serial number and user ID
  - EEPROM (Electrically erasable programmable ROM) holds program/protocol data
  - RAM holds temporary data generated by program execution
- executes a protocol to authenticate with reader/computer
  - static: similar to memory cards
  - dynamic: passwords created every minute; entered manually by user or electronically
  - challenge-response (e.g., computer creates a random number; smart card provides its hash, such as in password generators for Internet banking transfer)

Electronic identification cards

- An important application of smart cards
- A national e-identity (eID)
- Serves the same purpose as other national ID cards (e.g., a driver's licence)
  - provide stronger proof of identity
  - A German card contains: Personal data, Document number, Card access number (six digit random number), Machine readable zone (MRZ): the password
  - Uses: ePass (government use), eID (general use), eSign (can have private key and certificate)

Electronic identification cards

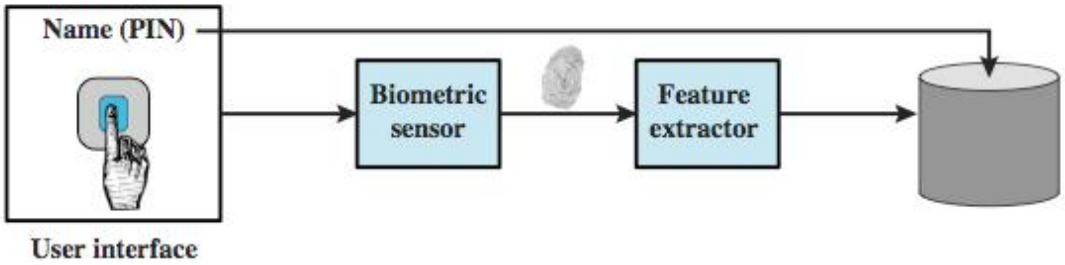


Biometric Authentication

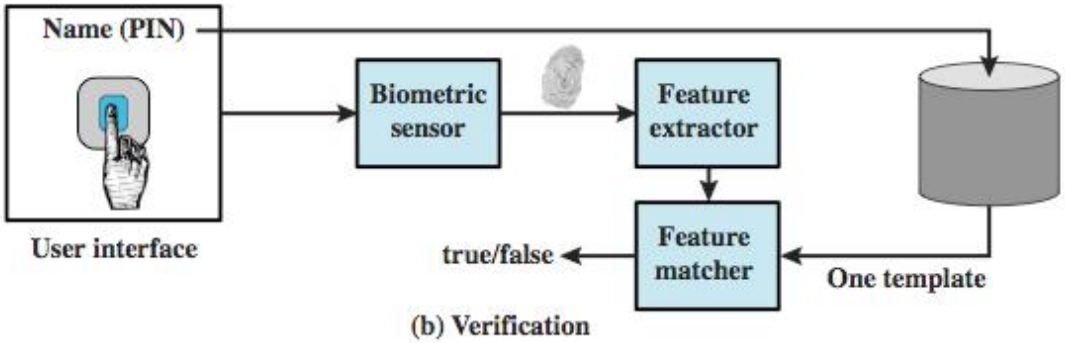
Authenticate user based on one of their physical characteristics:

- facial
- fingerprint
- hand geometry
- retina pattern
- iris
- signature
- voice

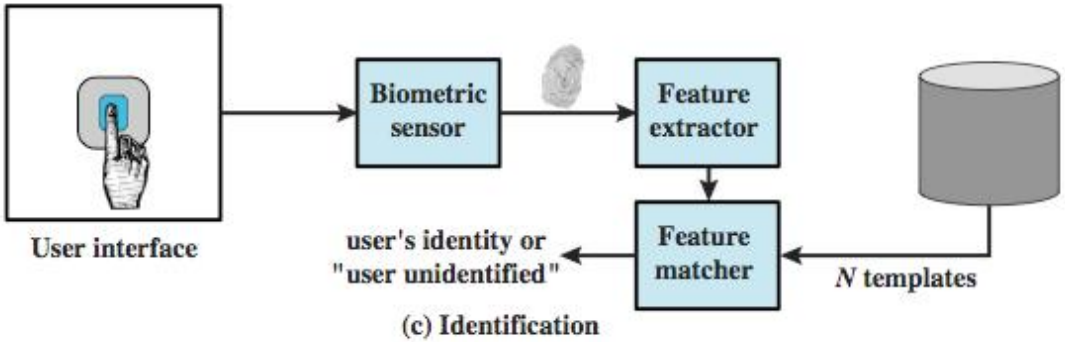
Operation of a biometric system



(a) Enrollment



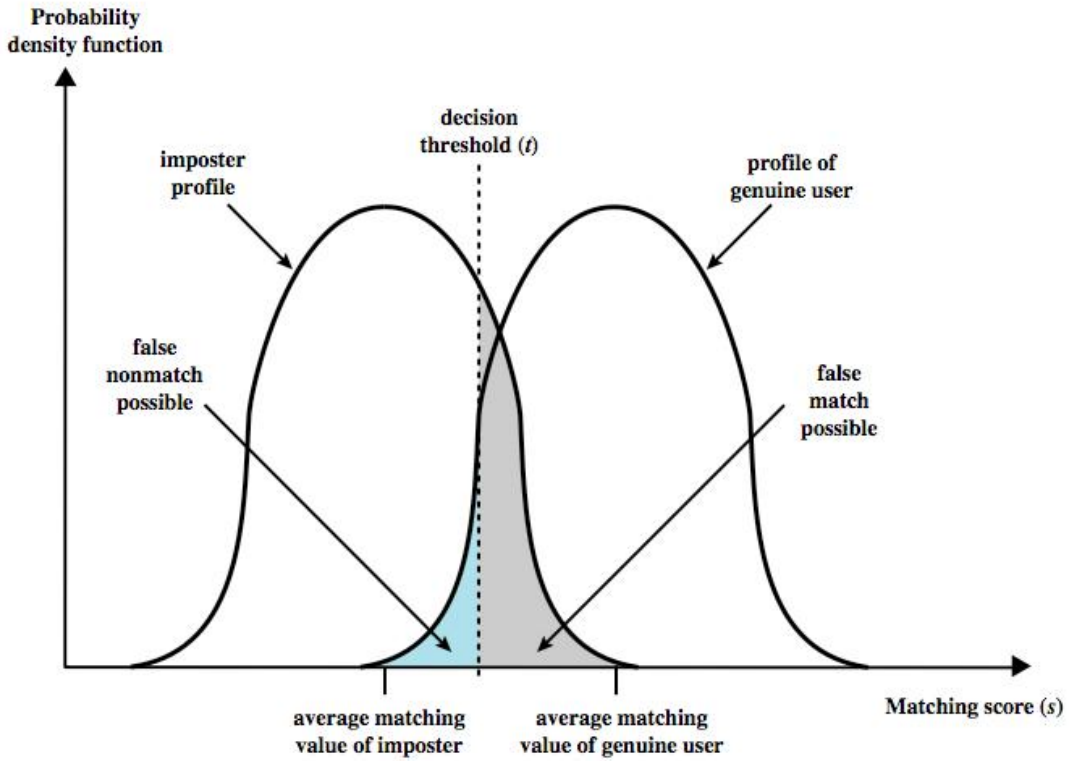
(b) Verification



(c) Identification

Biometric Accuracy

- The system generates a matching score (a number) that quantifies similarity between the input and the stored template
- Concerns: sensor noise and detection inaccuracy
- Problems of false match/false non-match



Remote User Authentication

- Authentication over network: eavesdropping, replay unavoidable
- Generally use challenge-response
  - user sends his/her identity
  - host responds with random number challenge
  - user replies with a value calculated from the random number plus possessed token/password/biometric values
- Protects against a number of attacks

A Protocol for Remote User Password Authentication

- $User \longrightarrow Host : UID$
- *Host* generates random value r, function $f()$ and hash function $h()$
- $Host \longrightarrow User : r, f(), h()$
- $User \longrightarrow Host : f(r, h(password))$
- *Host* decides whether to accept or deny *UID*'s access based on its knowledge of r, stored *password*, $f()$ and $h()$.

Types of Attacks on Remote User Authentication

- Attacker masquerades as a legitimate user (e.g., password guessing)
  - **Countermeasures: strong passwords; limit number of attempts**

Types of Attacks on Remote User Authentication

- Attacker masquerades as a legitimate user (e.g., password guessing)
  - **Countermeasures: strong passwords; limit number of attempts**
- Attacker attacks the host where passwords are stored
  - **Countermeasures: hashing, protect password databases**

Types of Attacks on Remote User Authentication

- Attacker masquerades as a legitimate user (e.g., password guessing)
  - **Countermeasures: strong passwords; limit number of attempts**
- Attacker attacks the host where passwords are stored
  - **Countermeasures: hashing, protect password databases**
- Eavesdropping: attacker attempts to learn passwords by observing the user, finding written passwords, keylogging
  - **Countermeasures: diligence to keep passwords secure, multifactor authentication (e.g., password + token, password + cellphone message), admin revoke compromised passwords**

Types of Attacks on Remote User Authentication

- Attacker masquerades as a legitimate user (e.g., password guessing)
  - **Countermeasures: strong passwords; limit number of attempts**
- Attacker attacks the host where passwords are stored
  - **Countermeasures: hashing, protect password databases**
- Eavesdropping: attacker attempts to learn passwords by observing the user, finding written passwords, keylogging
  - **Countermeasures: diligence to keep passwords secure, multifactor authentication (e.g., password + token, password + cellphone message), admin revoke compromised passwords**
- Replay: attacker repeats a previously captured user response
  - **Countermeasures: challenge-response, one-time passcode** r

Real Life Tokens

- Password devices (used by e.g., ICBC at *Phone banking*)
  - A random serial number N_c for each client c is stored in the device as well as stored at the bank
  - A button cell battery keeps an internal clock running (both parties synchronize on their clocks)
  - At time t, the device generates $f(t, N_c)$, which is to be checked at the bank's side in order to authenticate client c

Real Life Tokens

- Password devices (used by e.g., ICBC at *Phone banking*)
  - A random serial number N_c for each client c is stored in the device as well as stored at the bank
  - A button cell battery keeps an internal clock running (both parties synchronize on their clocks)
  - At time t, the device generates $f(t, N_c)$, which is to be checked at the bank's side in order to authenticate client c
- USB Key
  - Stores client's private key for mutual authentication
  - The USB Key cannot be copied and is tamper resistant
  - Key screen avoids certain man-in-the-middle attacks (session code goes round trip to shown on the screen, which value the user has to confirm by pressing a button)
  - To be combined with SSL/TLS web-based authentication when applied at *Internet banking*

Summary

- Introduced user authentication
  - using passwords
  - using tokens
  - using biometrics
- Remote user authentication
- Attacks and countermeasures