CSEN1001

Computer and Network Security

Mervat AbuElkheir

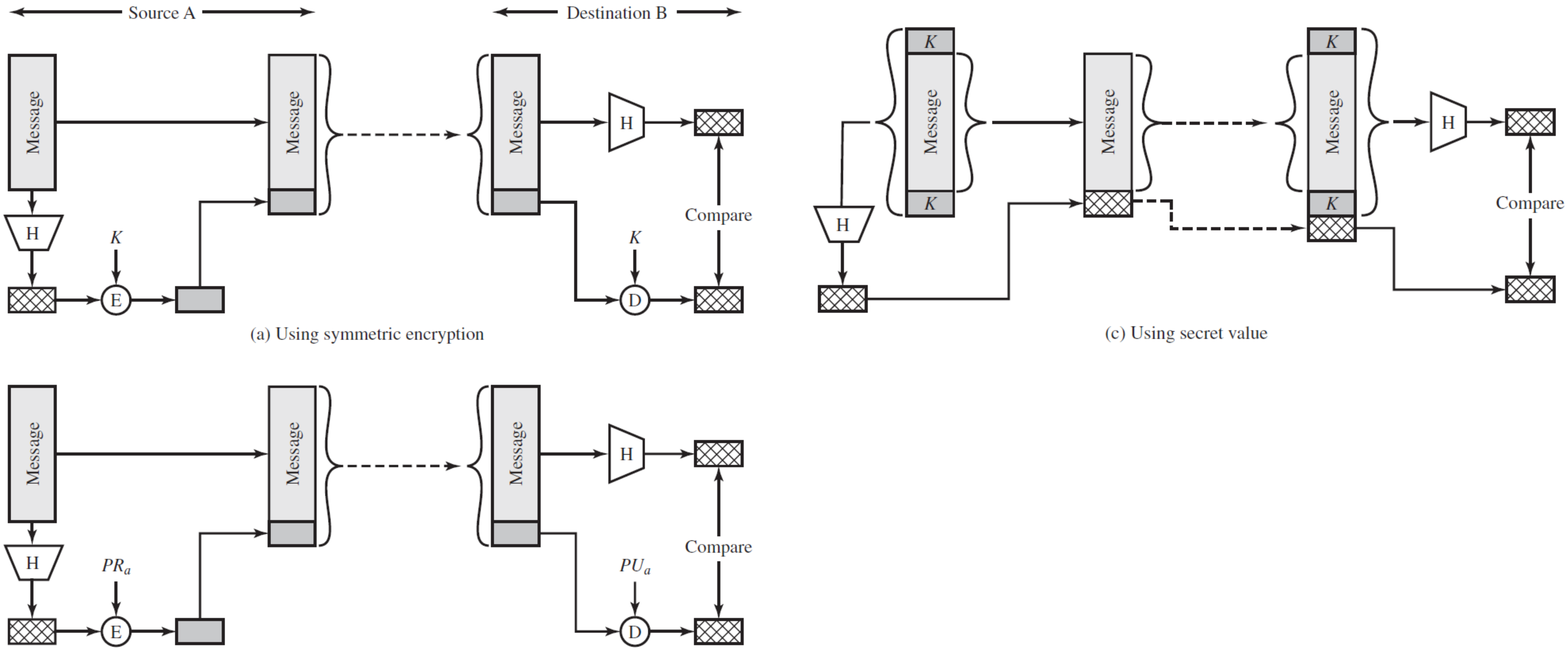
Ahmad Helmy

Mohamed Abdelrazik

Lecture (10)

User Authentication

MACs Using Hash Functions



(a) Using symmetric encryption

(b) Using public-key encryption

(c) Using secret value

Keyed-Hash Message Authentication Code (HMAC)

HMAC_K = Hash[(K$^+$ XOR opad) || Hash[(K$^+$ XOR ipad) || M)]

elements are:

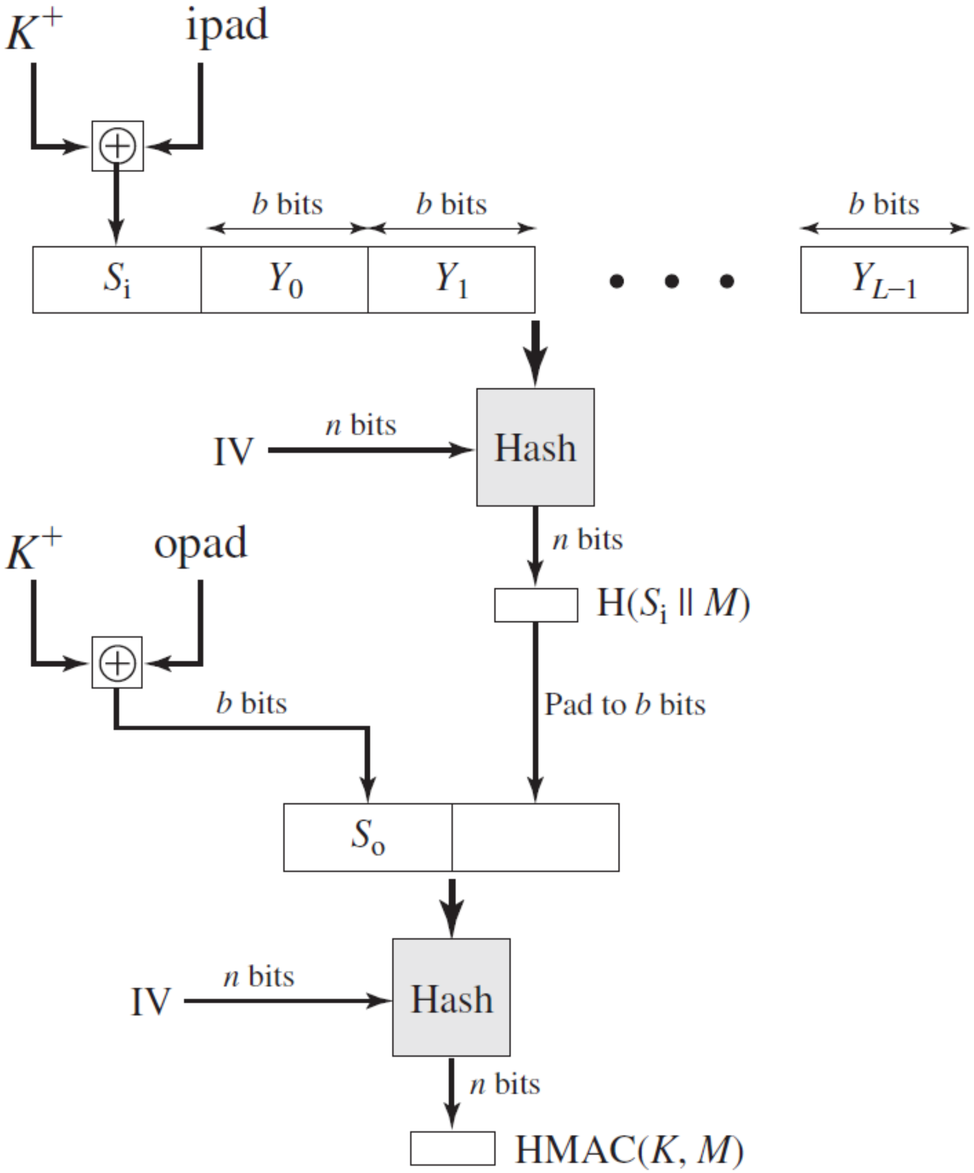
K$^+$ is K padded with zeros on the left so that the result is b bits in length

ipad is a pad value of 36 hex (00110110) repeated to fill block

opad is a pad value of 5C hex (01011100) repeated tc fill block

M is the message input to HMAC (including the padding specified in the embedded hash function)

Any hash function can be used (MD5, SHA-1, …)



User Authentication

❑ Fundamental security building block
  - basis of access control & user accountability

❑ Is the process of verifying an identity claimed by or for a system entity

❑ Has two steps:
  - identification - specify identifier
  - verification - bind entity (person) and identifier

❑ Distinct from message authentication

Means of User **(Local)** Authentication

- four means of authenticating user's identity
- based on something the individual
  - knows - e.g. password, PIN
  - possesses - e.g. key, token, smartcard
  - is (static biometrics) - e.g. fingerprint, retina
  - does (dynamic biometrics) - e.g. voice, sign
- can use alone or combined
- all can provide user authentication
- all have issues

Password Authentication

- ❑ Widely used user authentication method
  - user provides name/login 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
- ❑ System stores passwords in Password File
  - Need to protect that!

Password Vulnerabilities and Countermeasures

- Offline dictionary attack
- Specific account attack
- Popular password attack
- Password guessing against single user
- Workstation hijacking
- Exploiting user mistakes
- Exploiting multiple password use
- Electronic monitoring

Countermeasures

- Stop unauthorized access to password file
- Intrusion detection measures
- Account lockout mechanisms
- Policies against using common passwords but rather hard to guess passwords
- Training & enforcement of policies
- Automatic workstation logout
- Encrypted network links

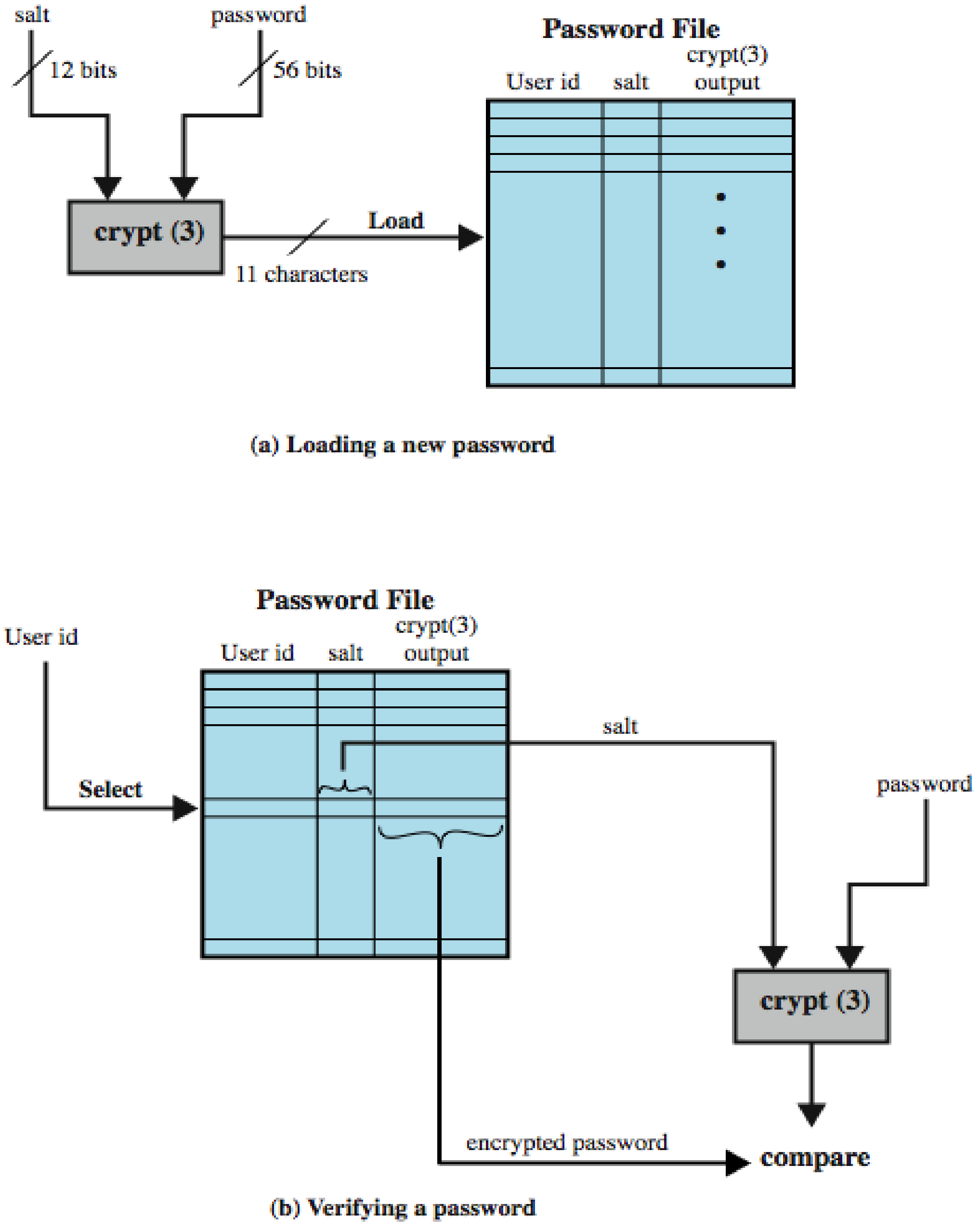
Use of Hashed Passwords

a **salt** is random data that is used as an additional input to a one-way function that "hashes" data, a password or passphrase Prevents duplicate passwords from being visible in the password file

A new salt is randomly generated for each password

defend against dictionary attacks and rainbow table attacks

Nearly impossible to tell if a person used the same password on multiple systems



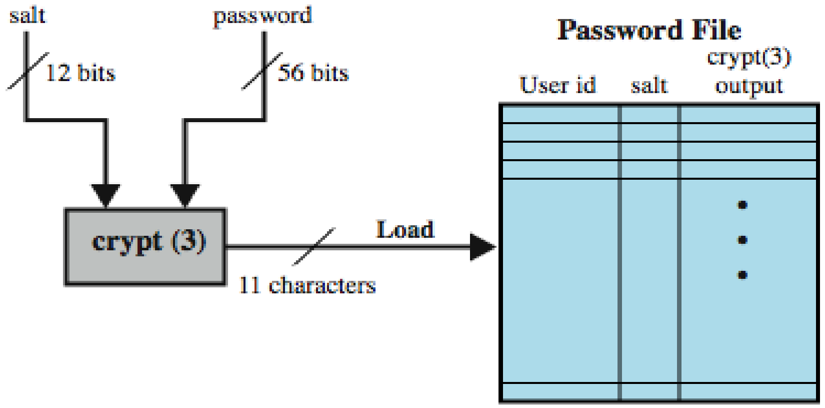
(a) Loading a new password

(b) Verifying a password

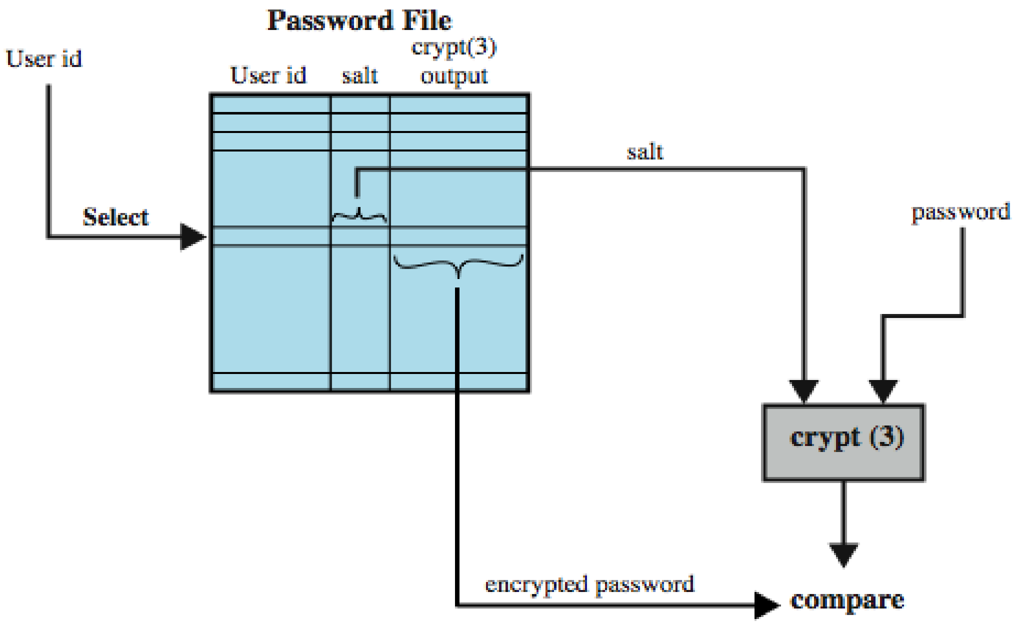
Use of Hashed Passwords

- 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



(a) Loading a new password



(b) Verifying a password

Password Cracking

❑ Dictionary attacks

- try each word then obvious variants in large dictionary against hash in password file

❑ Rainbow table attacks

- precompute tables of hash values for all salts
- 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

Password Choices

- ❑ Users may pick short passwords
  - e.g. 3% of 7000 accounts were 3 chars or less, easily guessed
  - system can reject choices that are too short
- ❑ Users may pick guessable passwords
  - so crackers use lists of likely passwords
  - e.g. one study of 14000 encrypted passwords guessed nearly 1/4 of them
  - would take about 1 hour on fastest systems to compute all variants, and only need 1 break!

MY KEYBOARD IS BROKEN. IT ONLY TYPES ASTERISKS FOR PASSWORDS.
scottadams@aol.com
www.dilbert.com
DOGBERT'S TECH SUPPORT
TRY CHANGING YOUR PASSWORD TO FIVE ASTERISKS.
© 2001 United Feature Syndicate, Inc.
9/6/01
I HOPE I CAN REMEMBER IT.

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 file
- ❑ Still have vulnerabilities
  - exploit O/S bug
  - accident with permissions making it readable
  - users with same password on other systems
  - access from unprotected backup media
  - sniff passwords in unprotected network traffic

Using Better Passwords

- Clearly have problems with passwords
- Goal to eliminate guessable passwords
- Whilst still easy for user to remember
- Techniques:
  - user education
  - computer-generated passwords
  - reactive password checking
  - proactive password checking

Token Authentication

- Object user possesses to authenticate, e.g.
  - embossed card
  - magnetic stripe card
  - memory card
  - smartcard

Memory Card

❑ Store but do not process data

❑ Magnetic stripe card, e.g. bank card

❑ Electronic memory card

❑ Used alone for physical access

❑ With password/PIN for computer use

❑ Drawbacks of memory cards include:

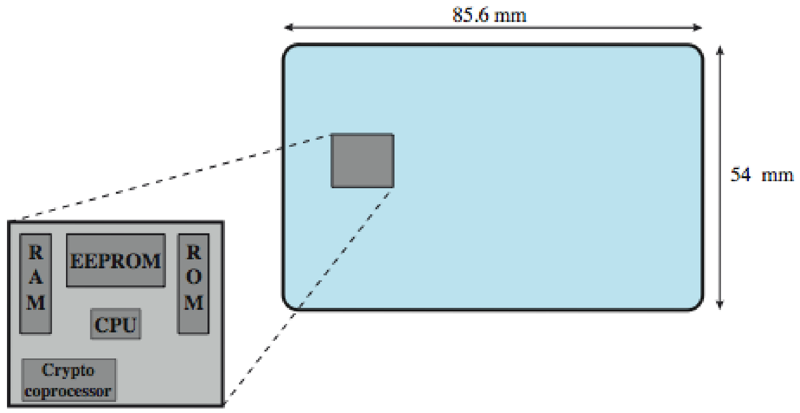
- need special reader
- loss of token issues
- user dissatisfaction

Smartcard

❑ Credit-card like

❑ Has own processor, memory, I/O ports

- wired or wireless access by reader
- may have crypto co-processor
- ROM, EEPROM, RAM memory

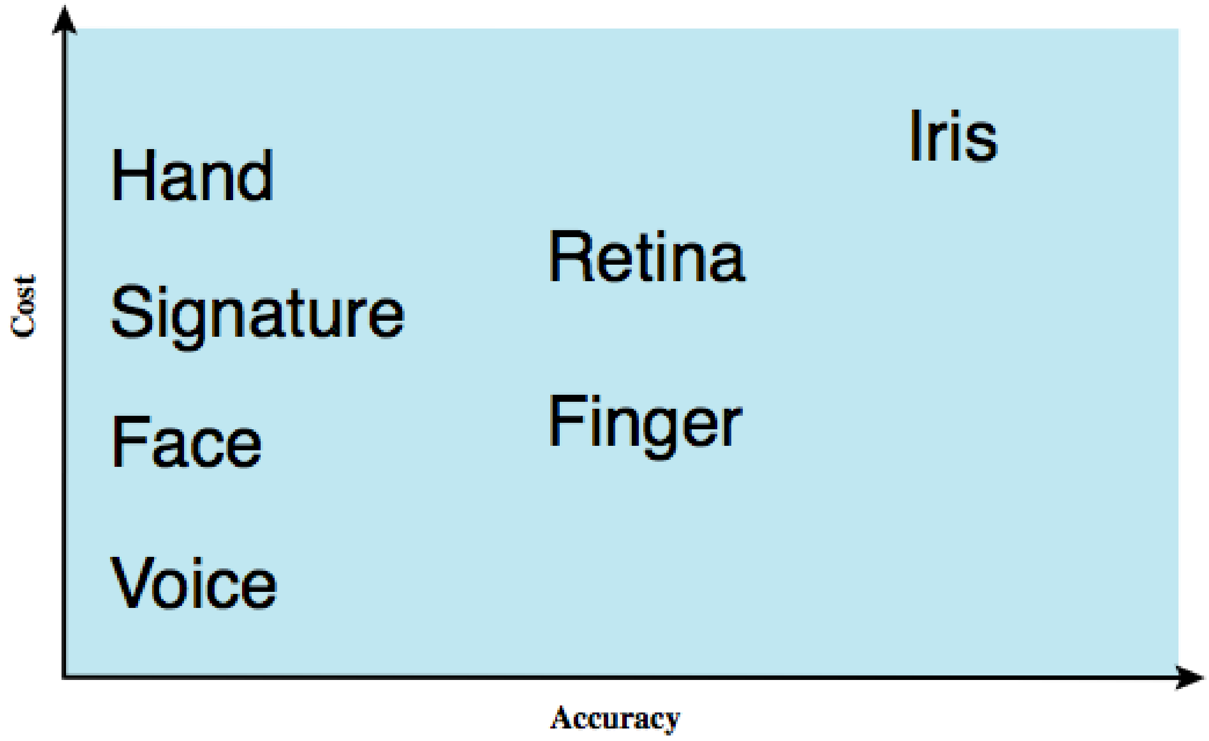


typical chip layout

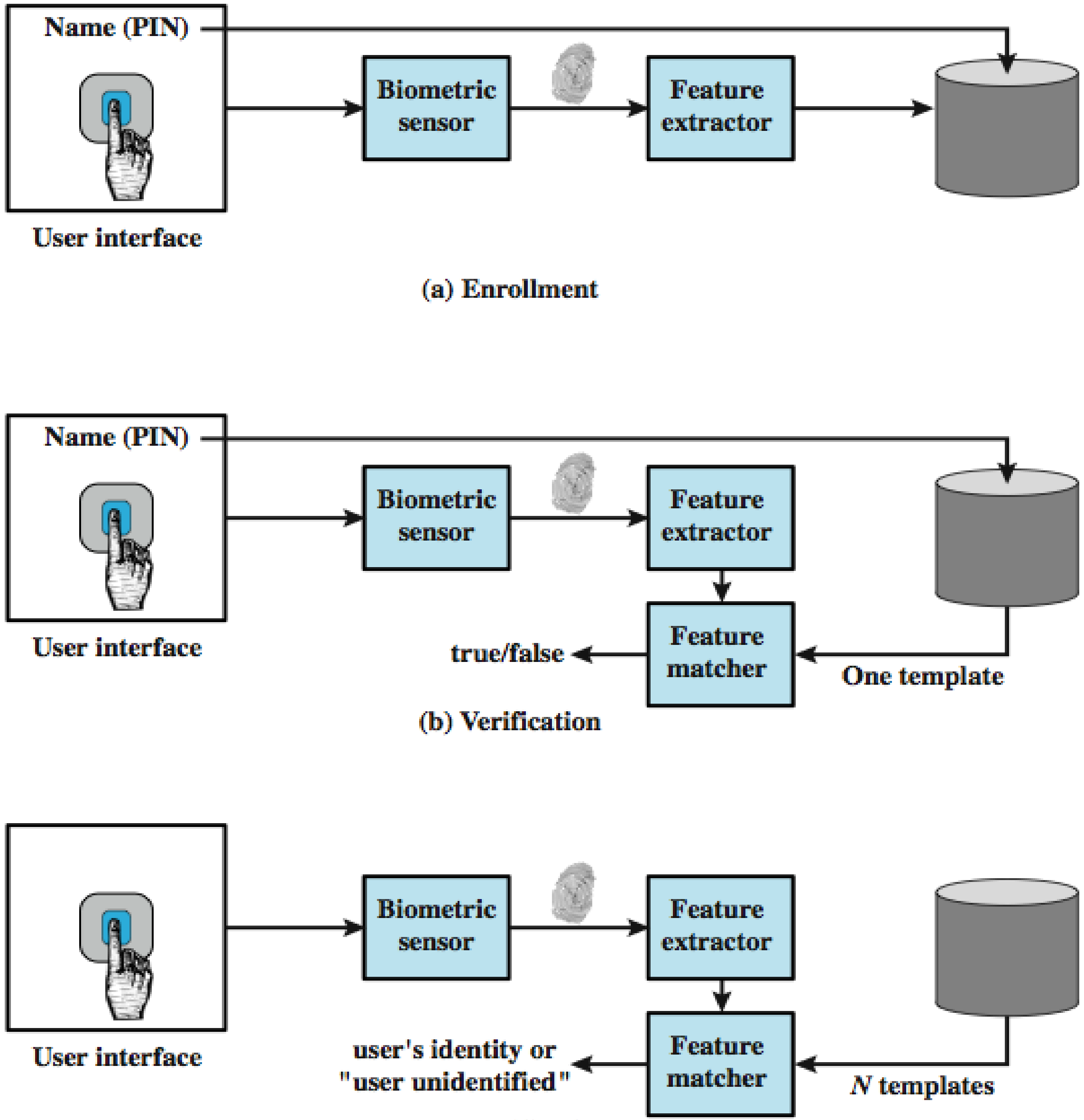
❑ Also have USB dongles

Biometric Authentication

- ❑Authenticate user based on one of their physical characteristics



Operation of a Biometric System



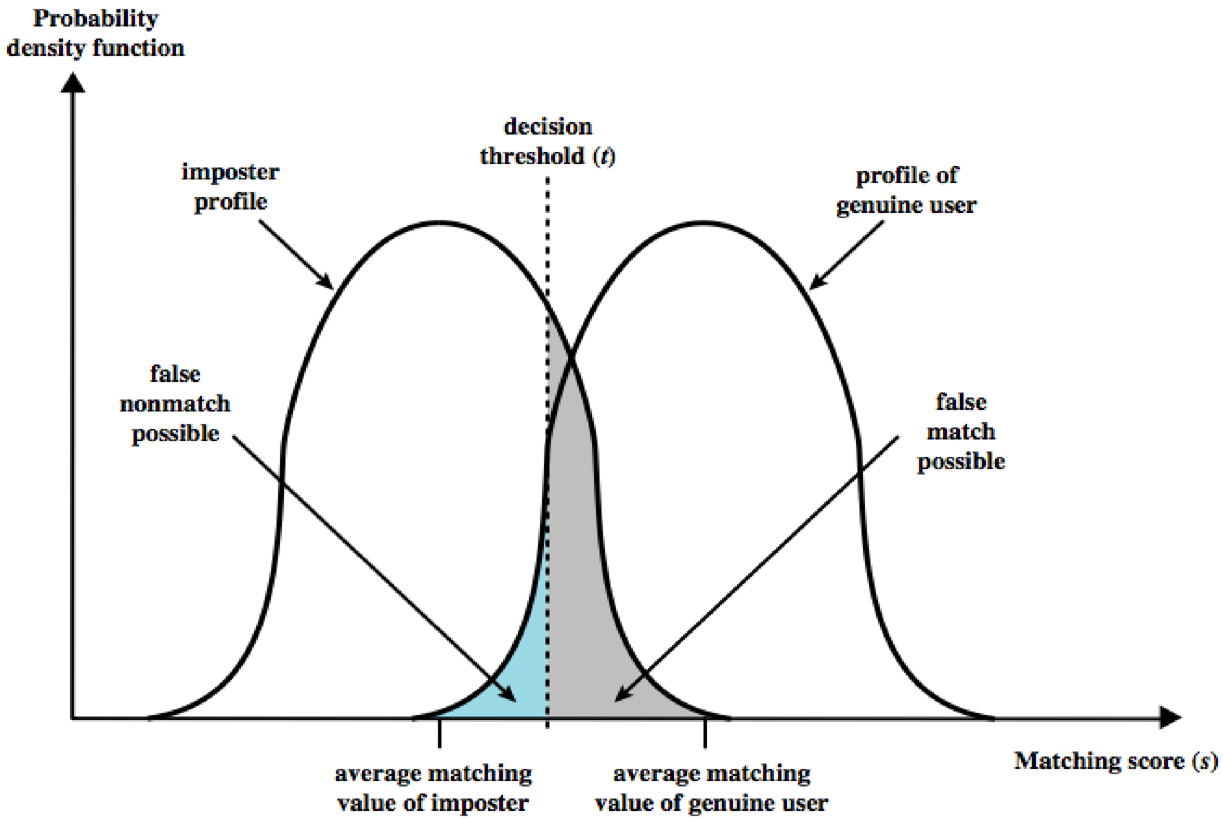
(a) Enrollment

(b) Verification

(c) Identification

Biometric Accuracy

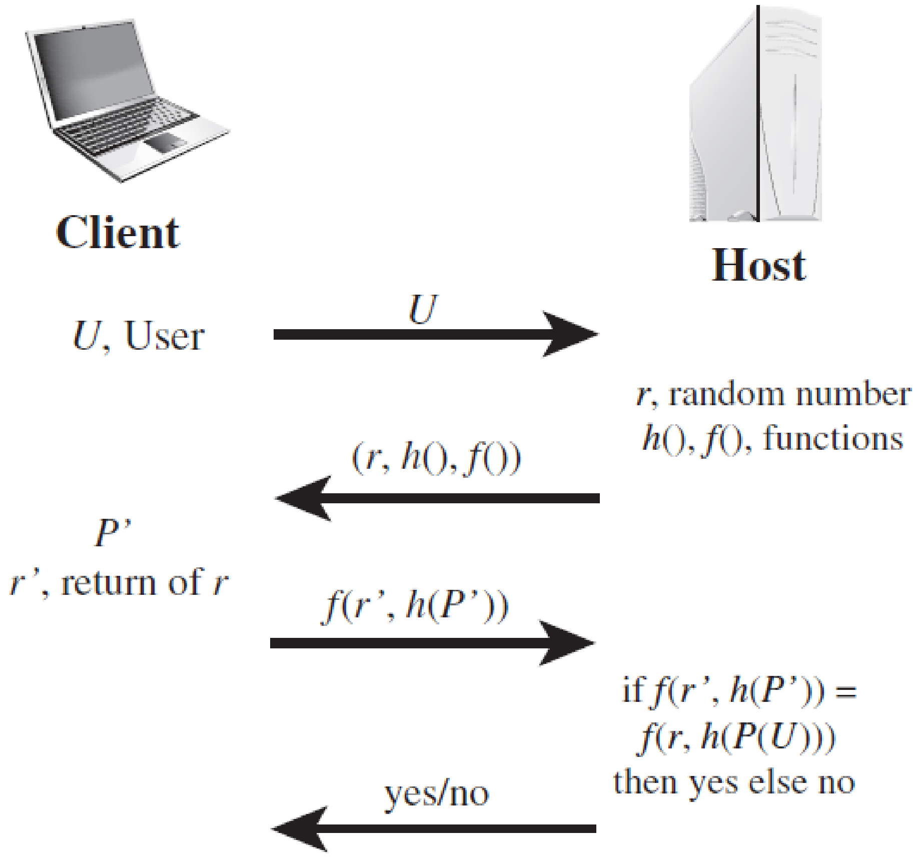
- ❑Never get identical templates
- ❑Problems of false match / false non-match



Remote User Authentication

- ❑ Authentication over network more complex
  - problems of eavesdropping, replay
- ❑ Generally use challenge-response
  - user sends identity
  - host responds with random number (nonce r)
  - user computes $f(r, h(P))$ and sends back
  - host compares value from user with own computed value, if match user authenticated

Remote User Authentication



Client

Host

U, User

U

r, random number
$h()$, $f()$, functions

$(r, h(), f())$

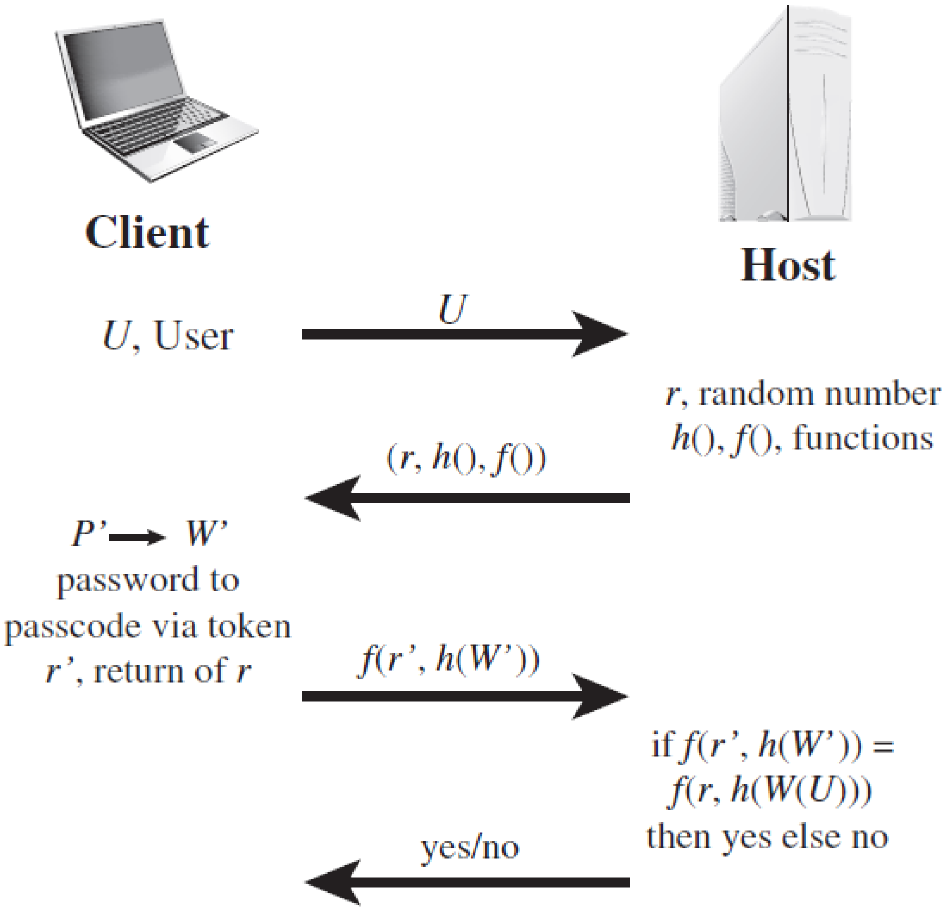
P'
r', return of r

$f(r', h(P'))$

if $f(r', h(P')) =$
$f(r, h(P(U)))$
then yes else no

yes/no

(a) Protocol for a password



Client

Host

U, User

U

r, random number
$h()$, $f()$, functions

$(r, h(), f())$

$P' \rightarrow W'$
password to
passcode via token
r', return of r

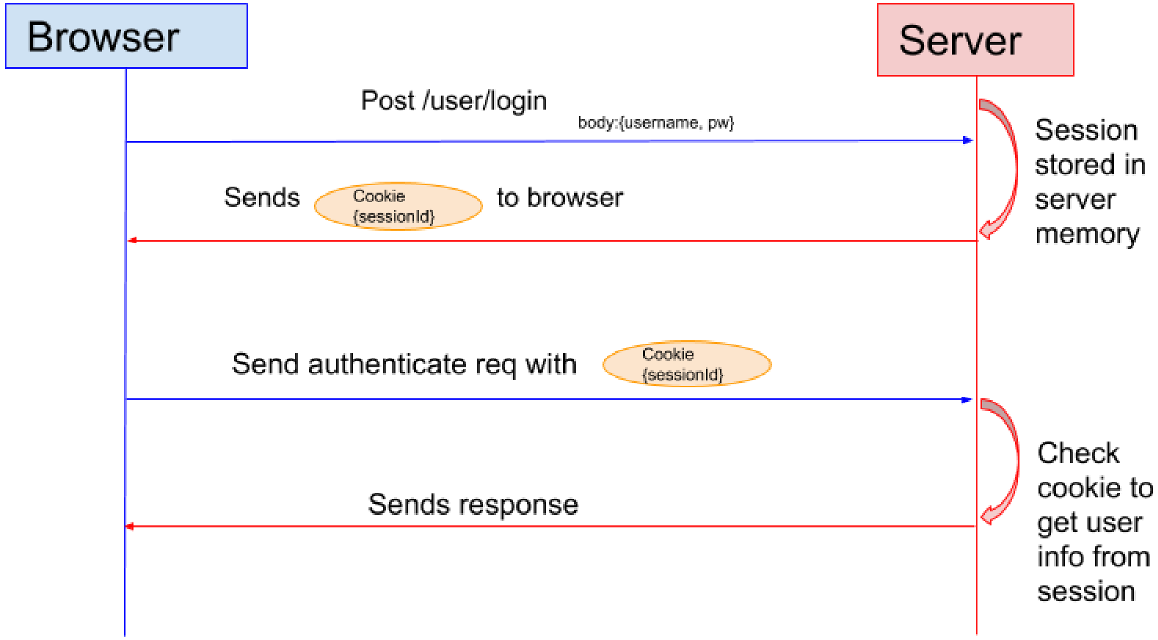
$f(r', h(W'))$

if $f(r', h(W')) =$
$f(r, h(W(U)))$
then yes else no

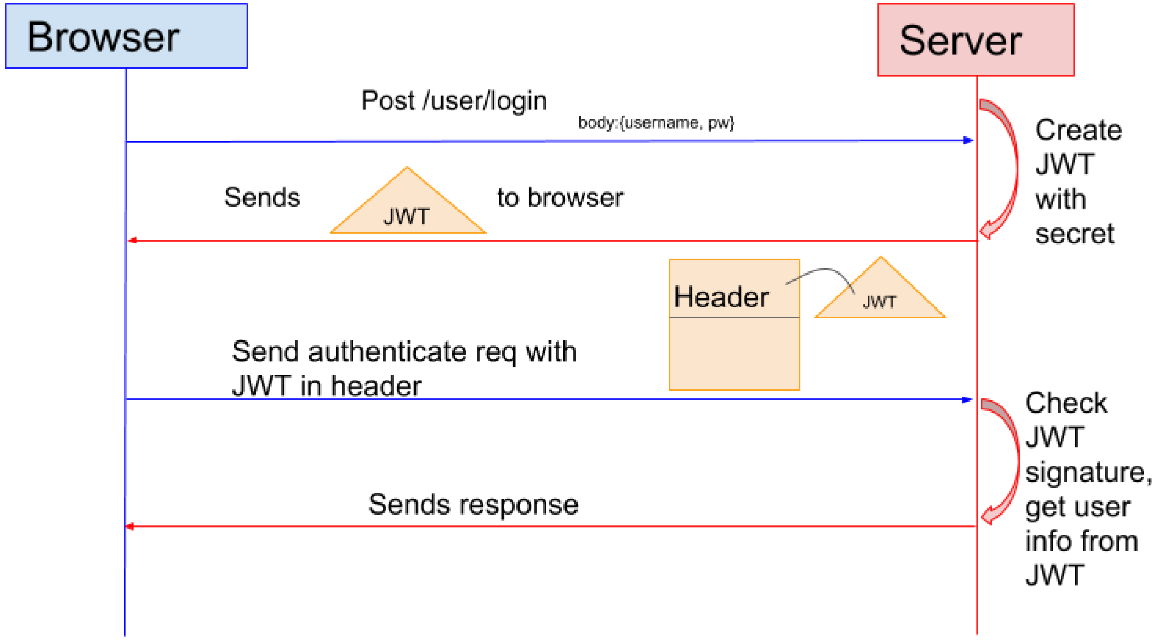
yes/no

(b) Protocol for a token

Remote User Authentication



Session Based Authentication (Server-based)



Token Based Authentication (Client-based)