USER AUTHENTICATION

MD. IFTEKHARUL ALAM EFAT

Institute of Information Technology
Noakhali Science & Technology University

LEARNING OBJECTIVES

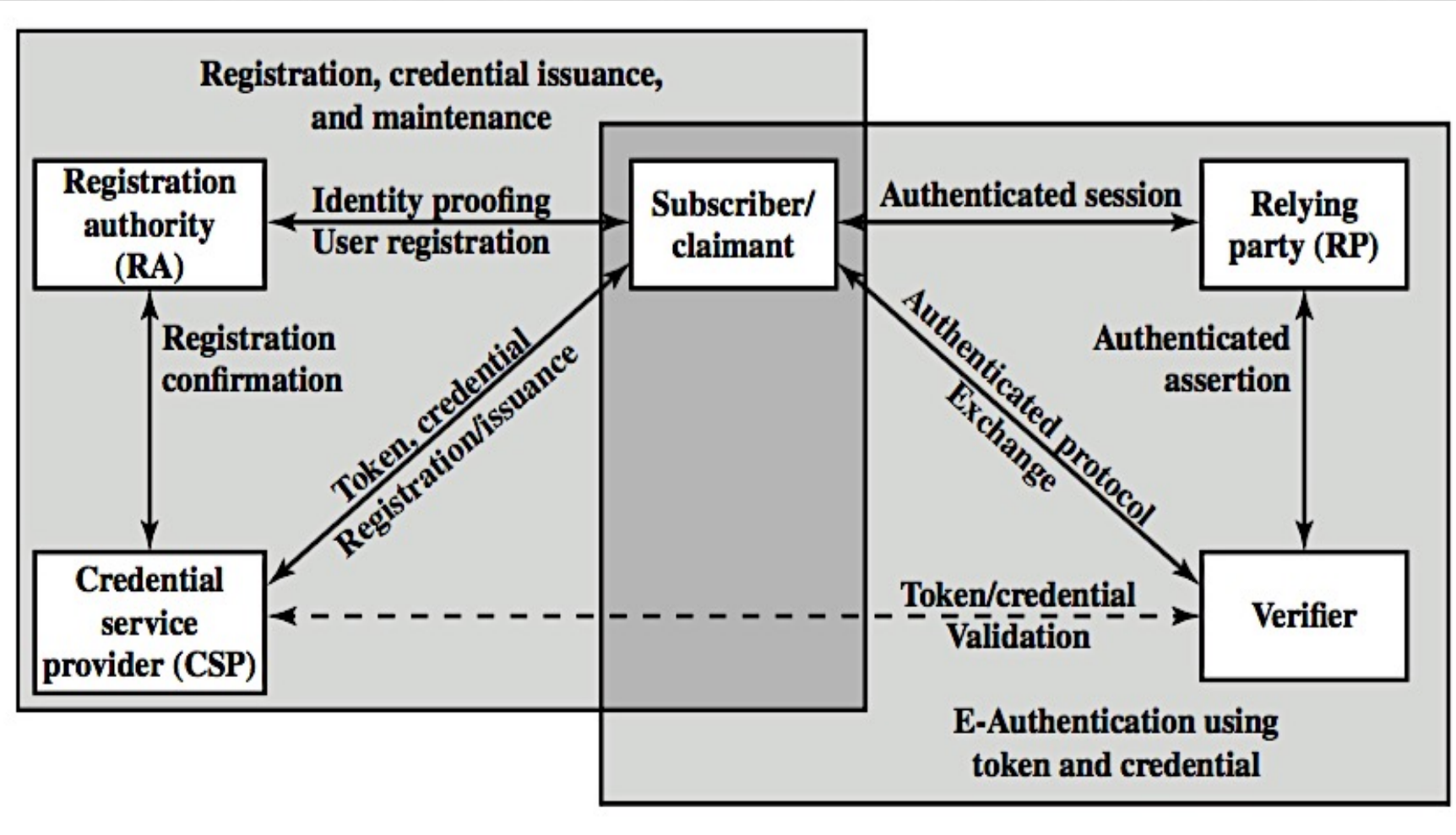
- Discuss the four general means of authenticating a user's identity
- Explain the mechanism by which hashed passwords are used for user authentication
- Understand the use of the Bloom filter in password management
- Present an overview of token-based user authentication
- Discuss the issues involved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

USER AUTHENTICATION

- User authentication is the basis for most types of access control and for user accountability
- The process of verifying an identity claimed by or for a system entity has two steps:
  - **Identification** - Presenting an identifier to the security system
  - **Verification** - Presenting or generating authentication information that corroborates the binding between the entity and the identifier
- Distinct from *Message Authentication*

ELECTRONIC USER AUTHENTICATION

E-AUTHENTICATION ARCHITECTURAL MODEL

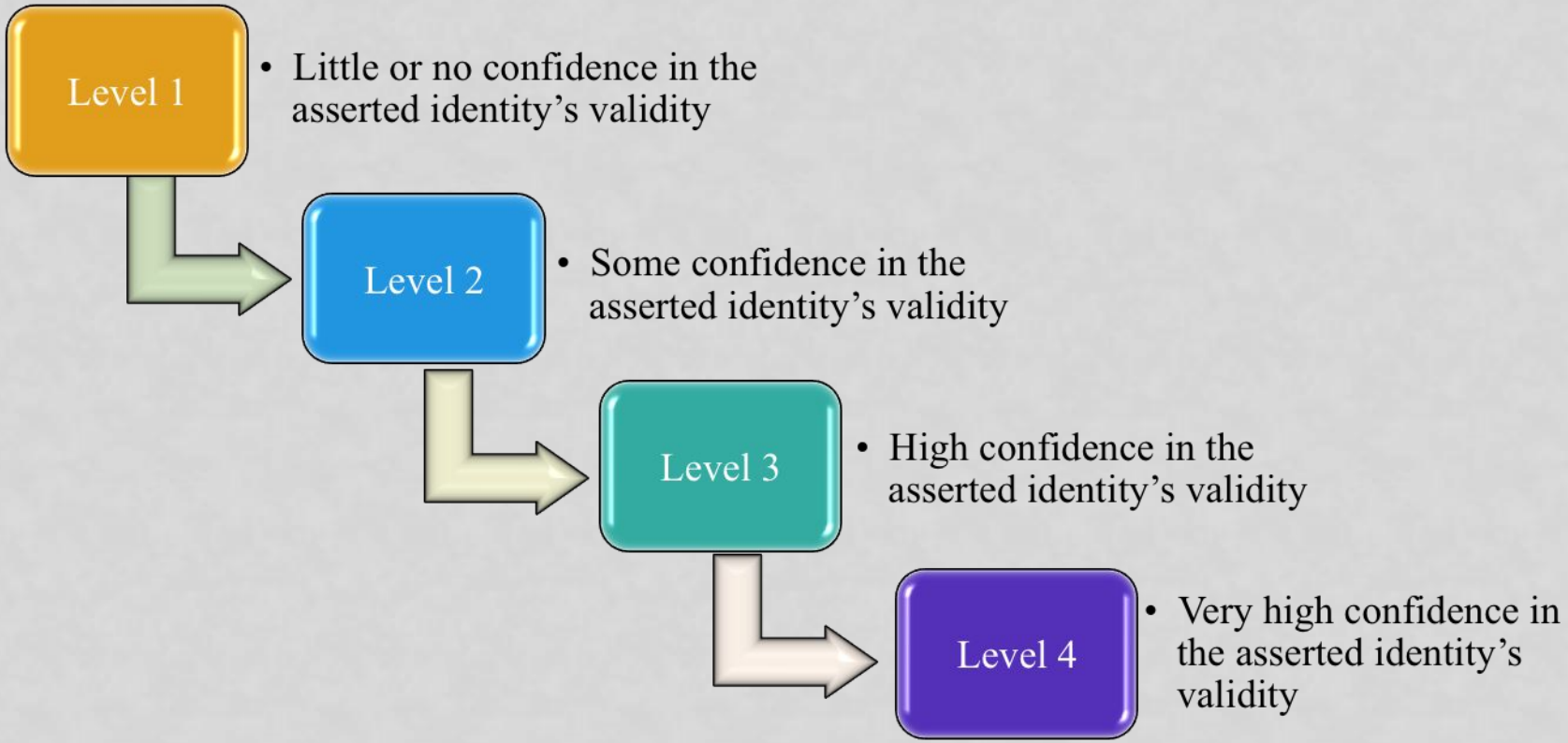


MEANS OF USER AUTHENTICATION

- Four means of authenticating user's identity based one 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

RISK ASSESSMENT FOR USER AUTHENTICATION

An assurance level describes an organization's degree of certainty that a user has presented a credential that refers to his or her identity



MAXIMUM POTENTIAL IMPACTS FOR EACH ASSURANCE LEVEL

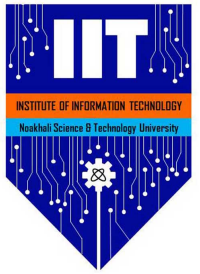
	Assurance Level Impact Profiles			
Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
Inconvenience, distress, or damage to standing or reputation	Low	Mod	Mod	High
Financial loss or organization liability	Low	Mod	Mod	High
Harm to organization programs or interests	None	Low	Mod	High
Unauthorized release of sensitive information	None	Low	Mod	High
Personal safety	None	None	Low	Mod/ High
Civil or criminal violations	None	Low	Mod	High

PASSWORD-BASED AUTHENTICATION

PASSWORD AUTHENTICATION

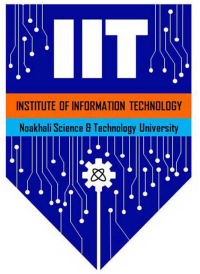
- Widely used user authentication method
  - user provides name/login and password
  - system compares password with that saved for specified login
- Authenticates ID of user logging and
  - that the user is authorized to access system
  - determines the user's privileges
  - is used in discretionary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

PASSWORD VULNERABILITIES



- 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



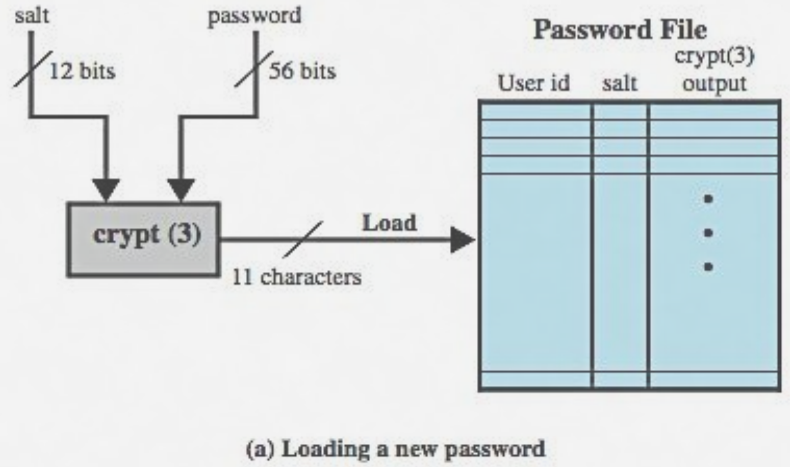
- Stop unauthorized access to password file
- Intrusion detection measures
- Account lockout mechanisms
- Policies against using common passwords but rather hard to guess passwords
- Training & enforcement of policies
- Automatic workstation logout
- Encrypted network links

USE OF HASHED PASSWORDS

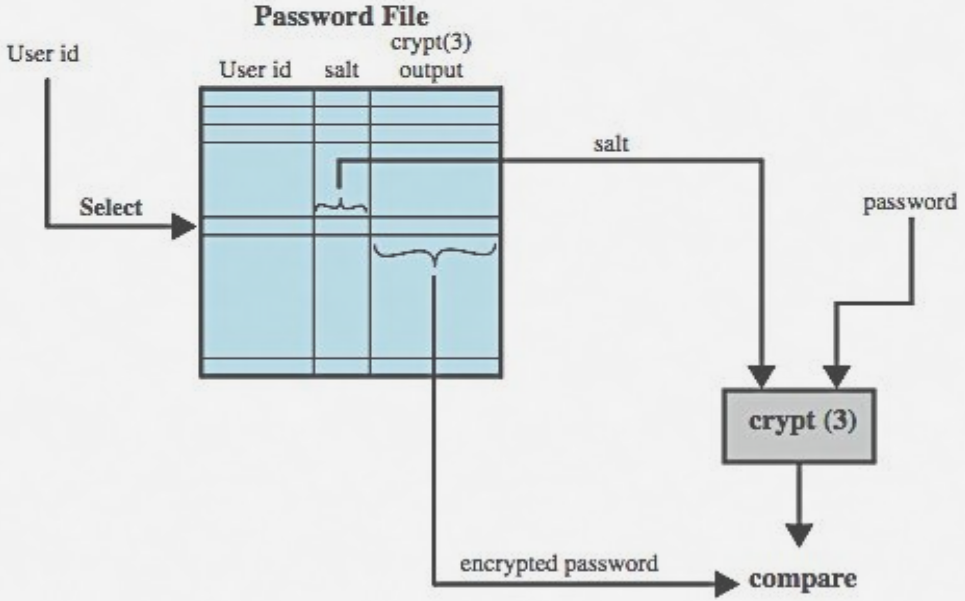
UNIX Password Scheme

Password:
`farm1990M0O`
Salt:
`f1nd1ngn3m0`
Salted input:
`f1nd1ngn3m0farm1990M0O`

Hash (SHA-256):
`7528ed35c6ebf7e4661a02fd98ab88d9`
`2ccf4e48a4b27338fcc194b90ae8855c`



(a) Loading a new password

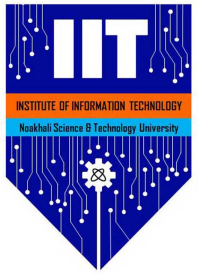


(b) Verifying a password

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
  - 0 value repeatedly encrypted 25 times
  - Output translated to 11 character sequence
- Now regarded as woefully insecure
  - E.G. Supercomputer, 50 million tests, 80 min
- Sometimes still used for compatibility

GENERATING A GOOD RANDOM SALT



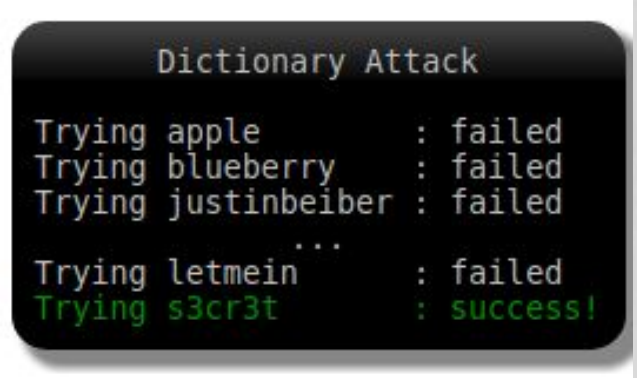
- Generate a unique salt upon creation of each stored credential (not just per user or system-wide)
- Use cryptographically-strong random data
- As storage permits, use a 32-byte or 64-byte salt (actual size dependent on protection function)
- Scheme security does not depend on hiding, splitting, or otherwise obscuring the salt

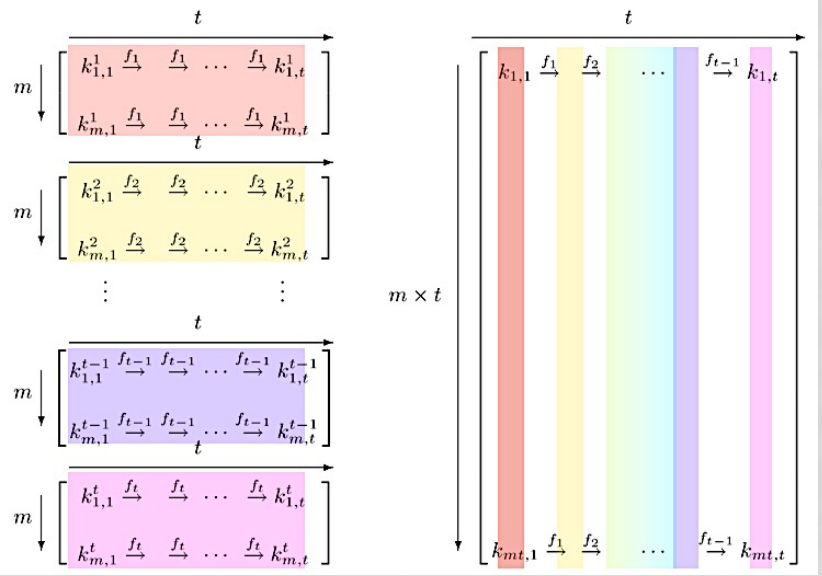
IMPROVED IMPLEMENTATIONS

- 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 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
  - 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

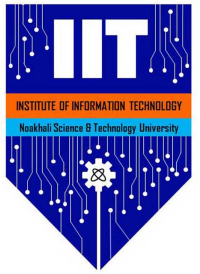




PASSWORD CHOICES

- Users may pick short passwords
  - E.G. 3% 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!

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

PASSWORD SELECTION STRATEGIES

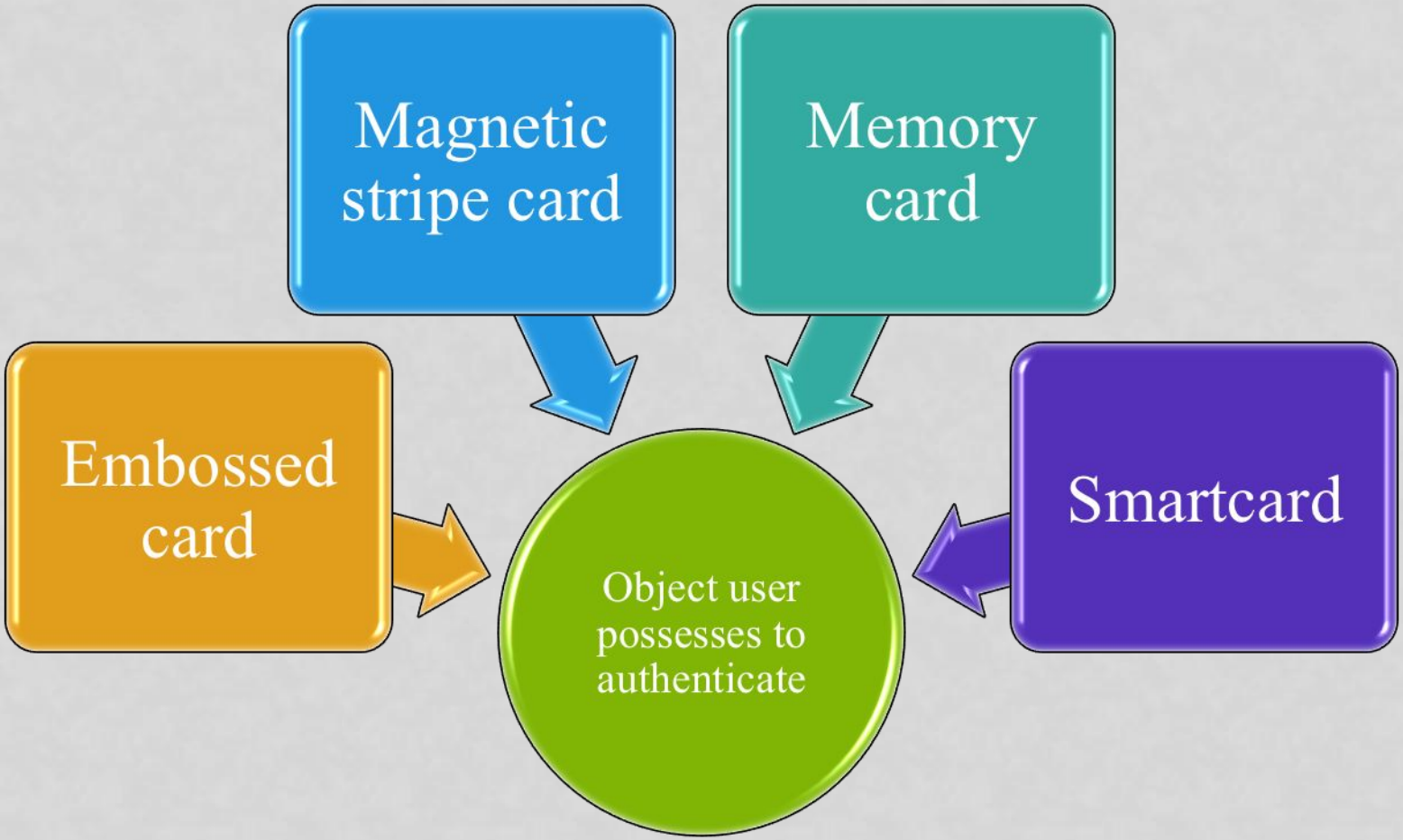
- Clearly have problems with passwords
- Goal to eliminate guessable passwords
- Whilst still easy for user to remember
- Techniques:
  - User education
  - Computer-generated passwords
  - Reactive password checking
  - Proactive password checking

PROACTIVE PASSWORD CHECKING

- Rule enforcement plus user advice, e.g.
  - 8+ chars, upper/lower/numeric/punctuation
  - May not suffice
- Password cracker
  - Time and space issues
- 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

TOKEN AUTHENTICATION

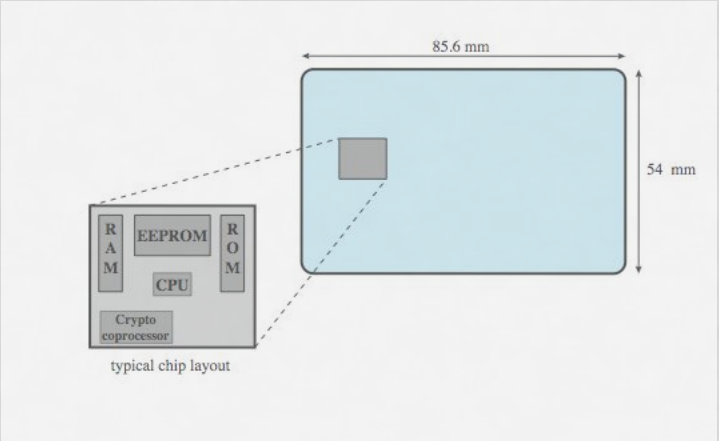


MEMORY CARD

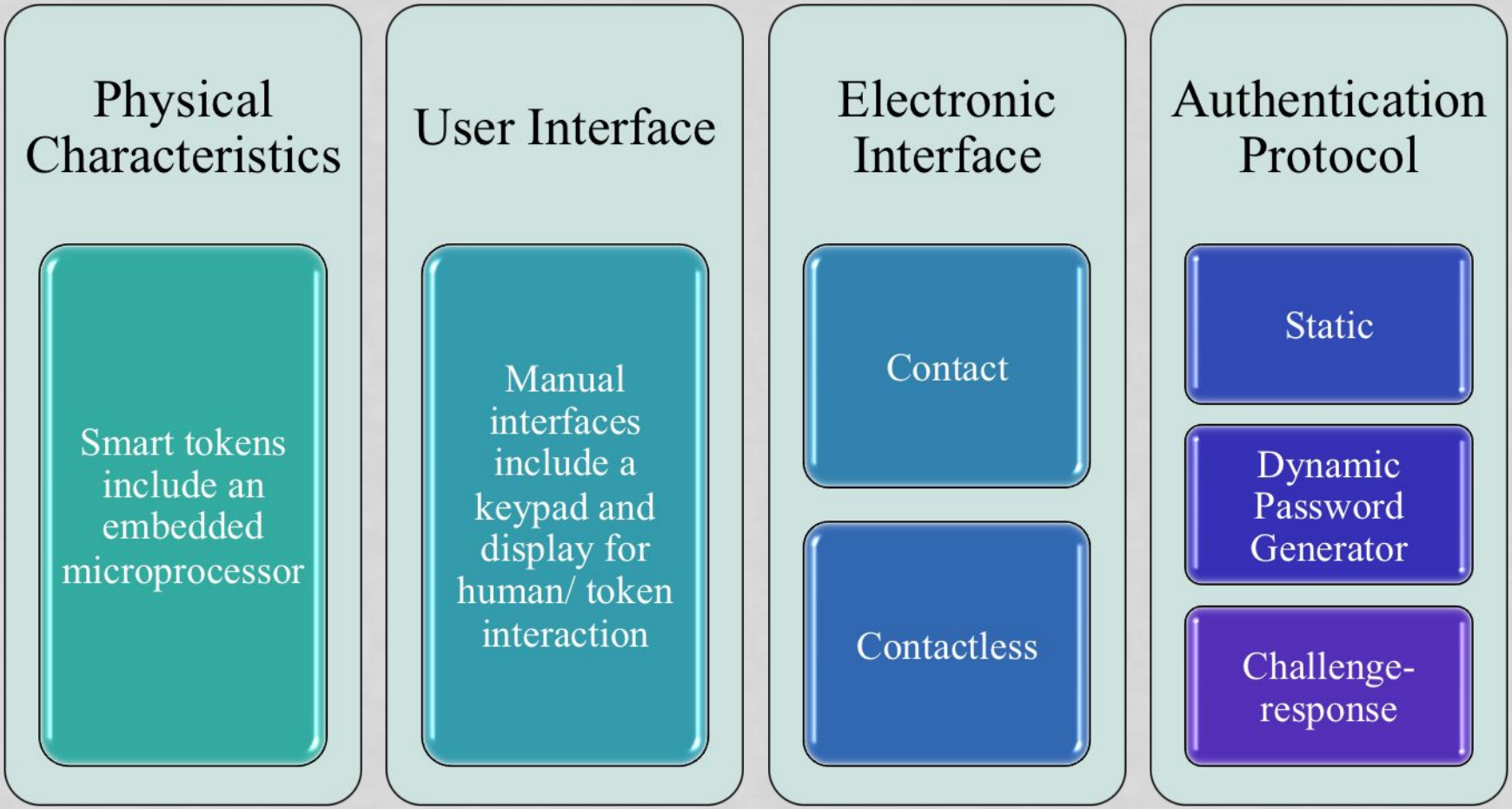
- Store but do not process data
- Magnetic stripe card, e.g. Bank card
- Electronic memory card
- Used alone for physical access
- With password/PIN for computer use
- Drawbacks of memory cards include:
  - Need special reader
  - Loss of token issues
  - User dissatisfaction

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
- Executes protocol to authenticate with reader/computer
- Also have USB dongles

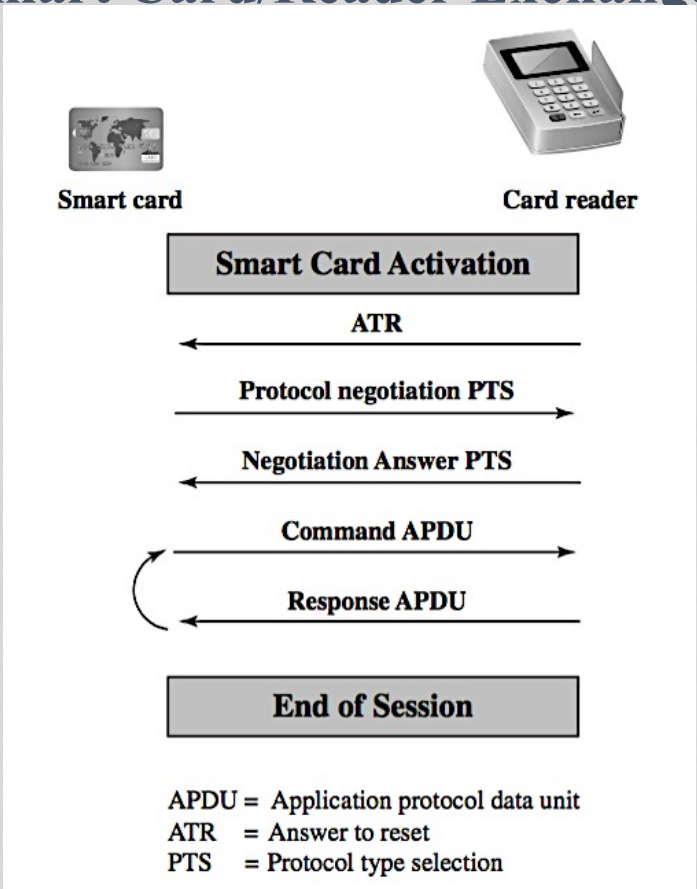


SMART CARDS

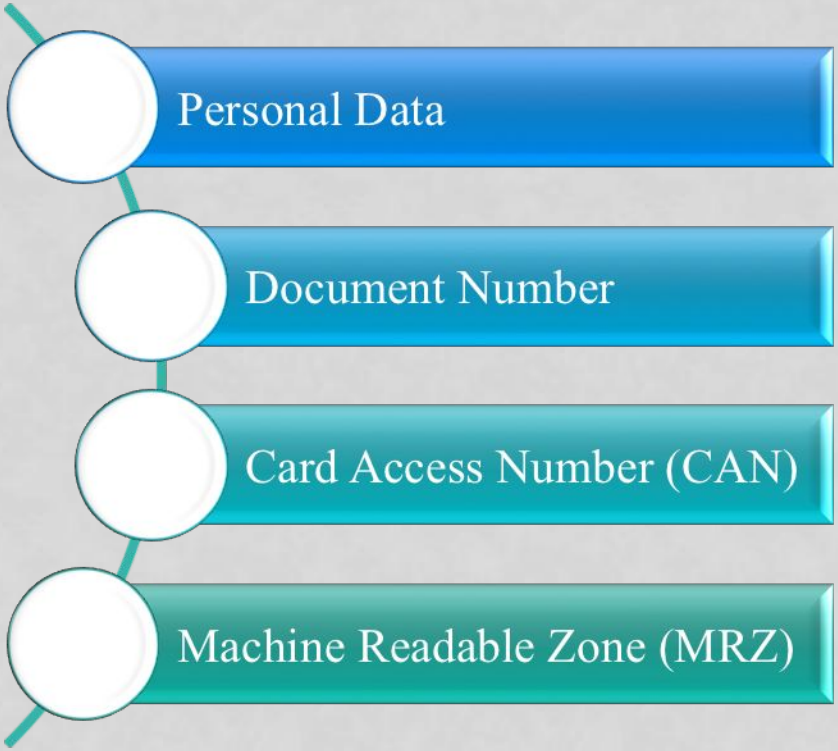

Physical Characteristics
Smart tokens include an embedded microprocessor
User Interface
Manual interfaces include a keypad and display for human/ token interaction
Electronic Interface
Contact
Contactless
Authentication Protocol
Static
Dynamic Password Generator
Challenge-response

ELECTRONIC IDENTITY CARDS

Smart Card/Reader Exchange

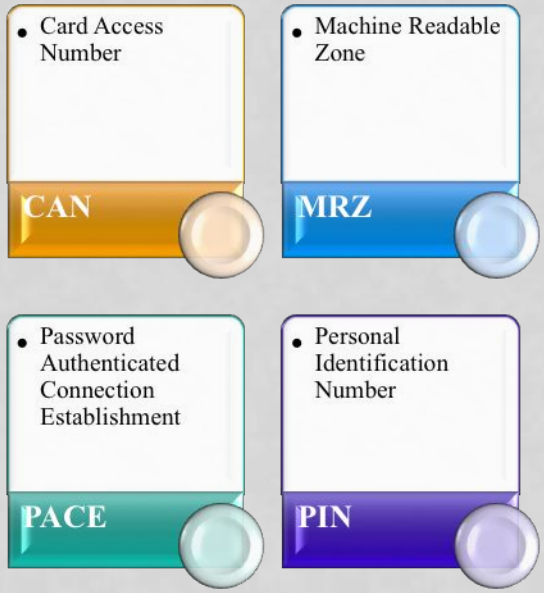


Human-Readable Data Printed on Smartcard

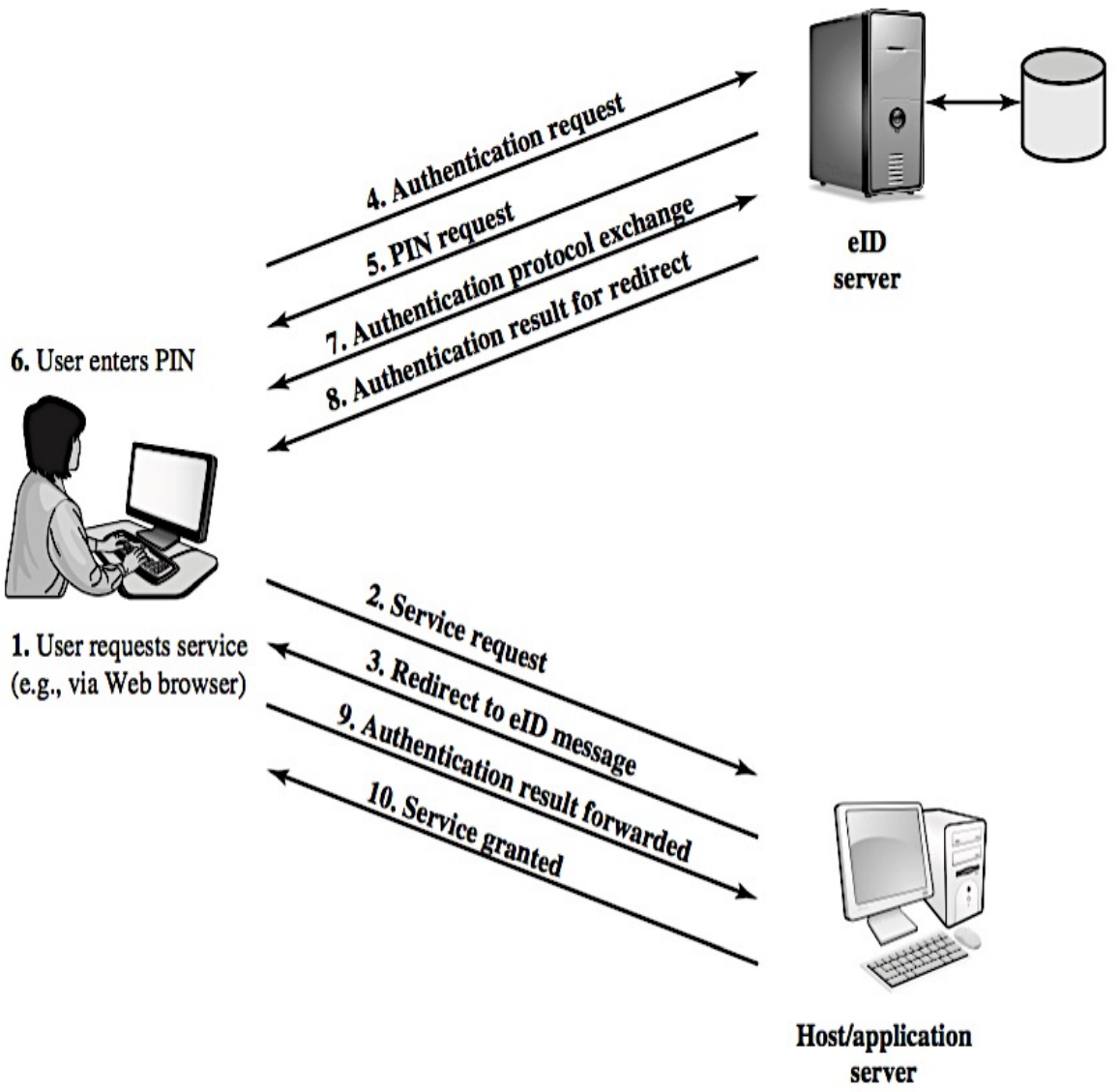


ELECTRONIC FUNCTIONS AND DATA FOR EID CARDS

Function	Purpose	PACE Password	Data	Uses
ePass (mandatory)	Authorized offline inspection systems read the data	CAN or MRZ	Face image; two fingerprint images (optional); MRZ data	Offline biometric identity verification reserved for government access
eID (activation optional)	Online applications read the data or access functions as authorized	eID PIN	Family and given names; artistic name and doctoral degree: date and place of birth; address and community ID; expiration date	Identification; age verification; community ID verification; restricted identification (pseudonym); revocation query
	Offline inspection systems read the data and update the address and community ID	CAN or MRZ		
eSign (certificate optional)	A certification authority installs the signature certificate online	eID PIN	Signature key; X.509 certificate	Electronic signature creation
	Citizens make electronic signature with eSign PIN	CAN		



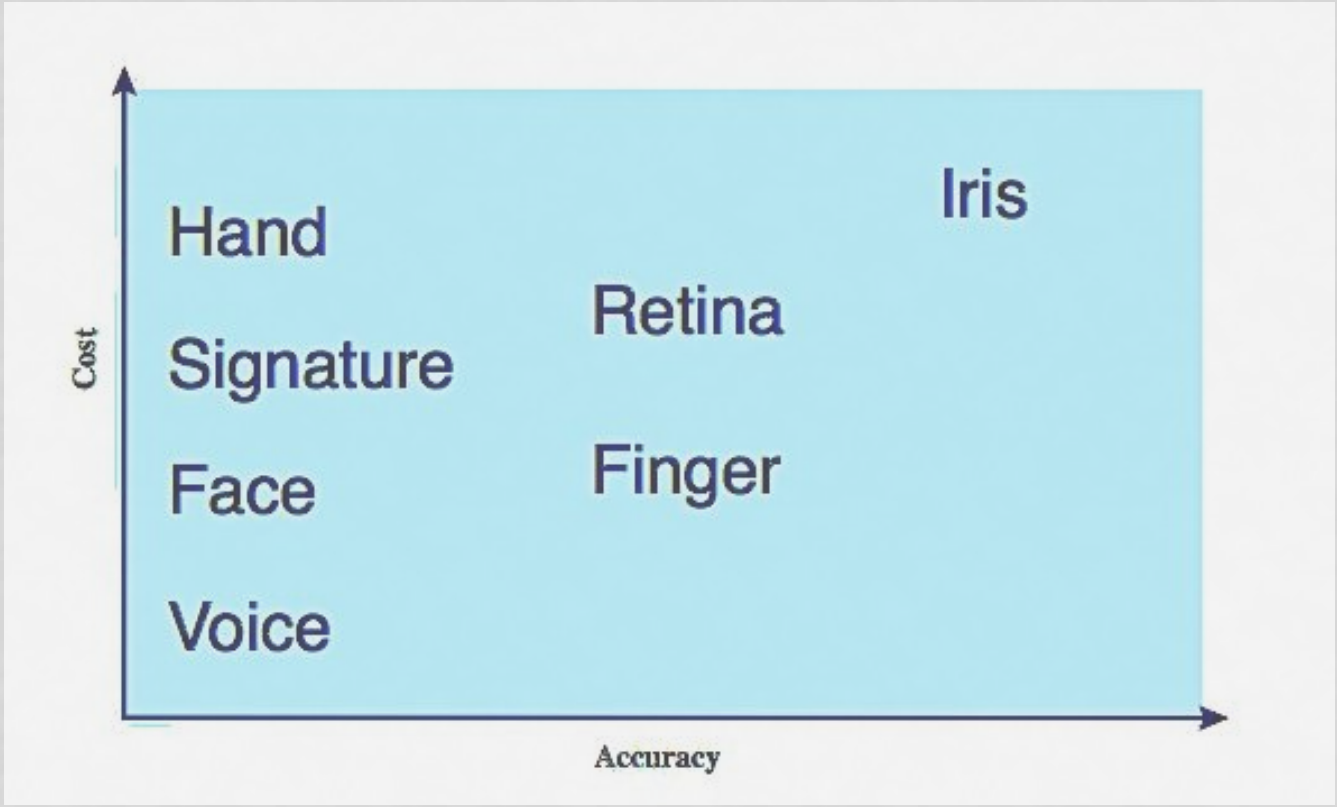
USER AUTHENTICATION WITH EID



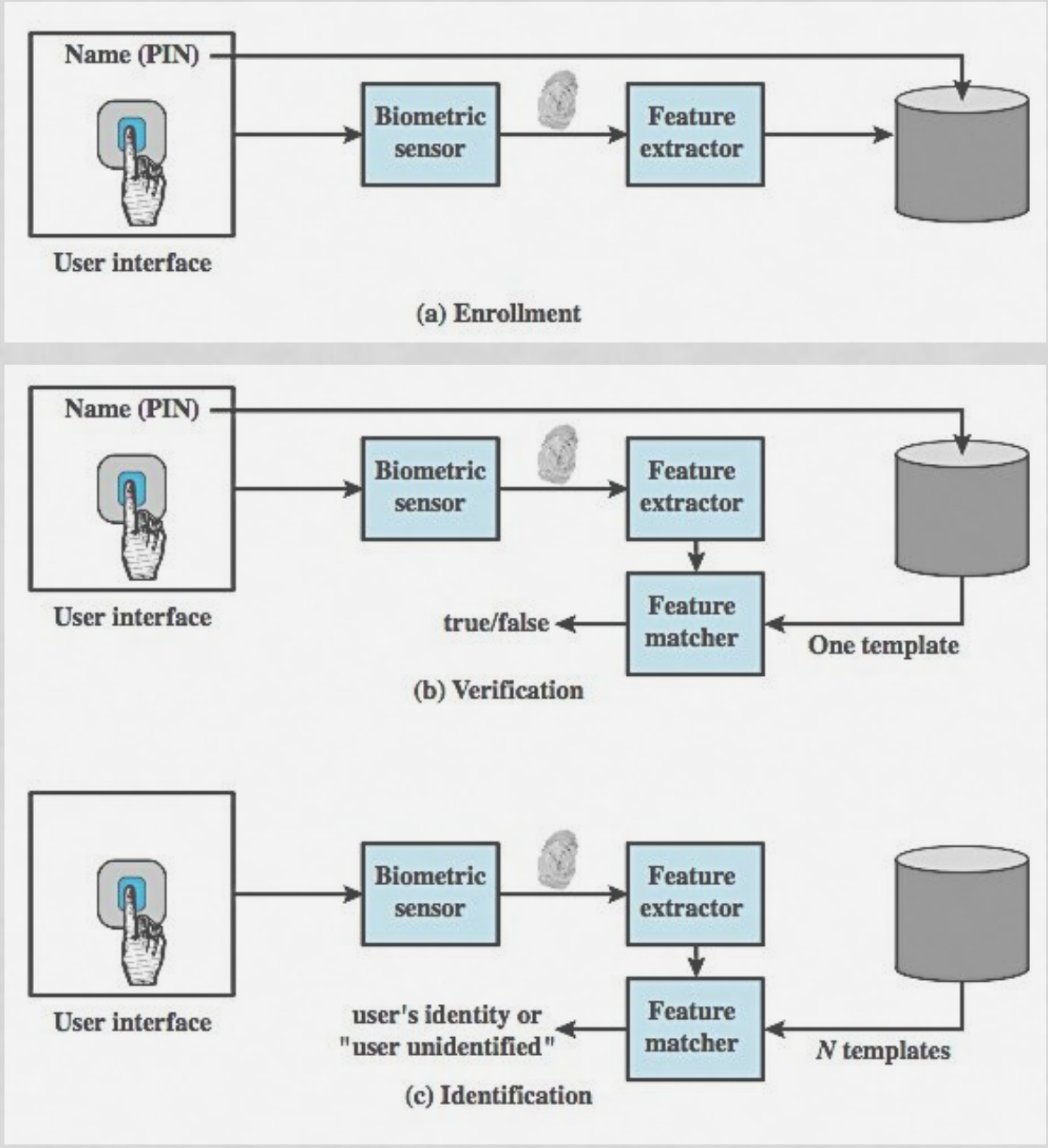
BIOMETRIC AUTHENTICATION

BIOMETRIC AUTHENTICATION

Authenticate user based on one of their physical characteristics

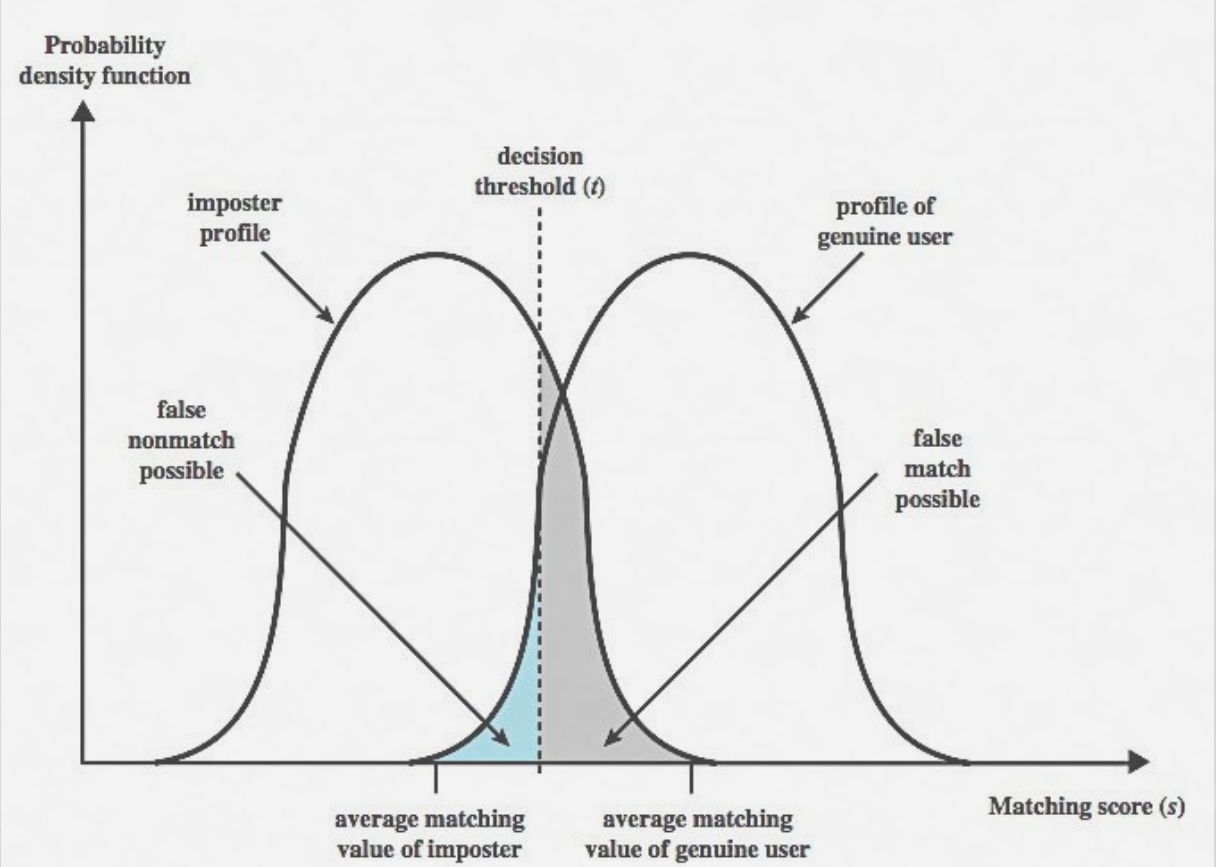


OPERATION OF A BIOMETRIC SYSTEM



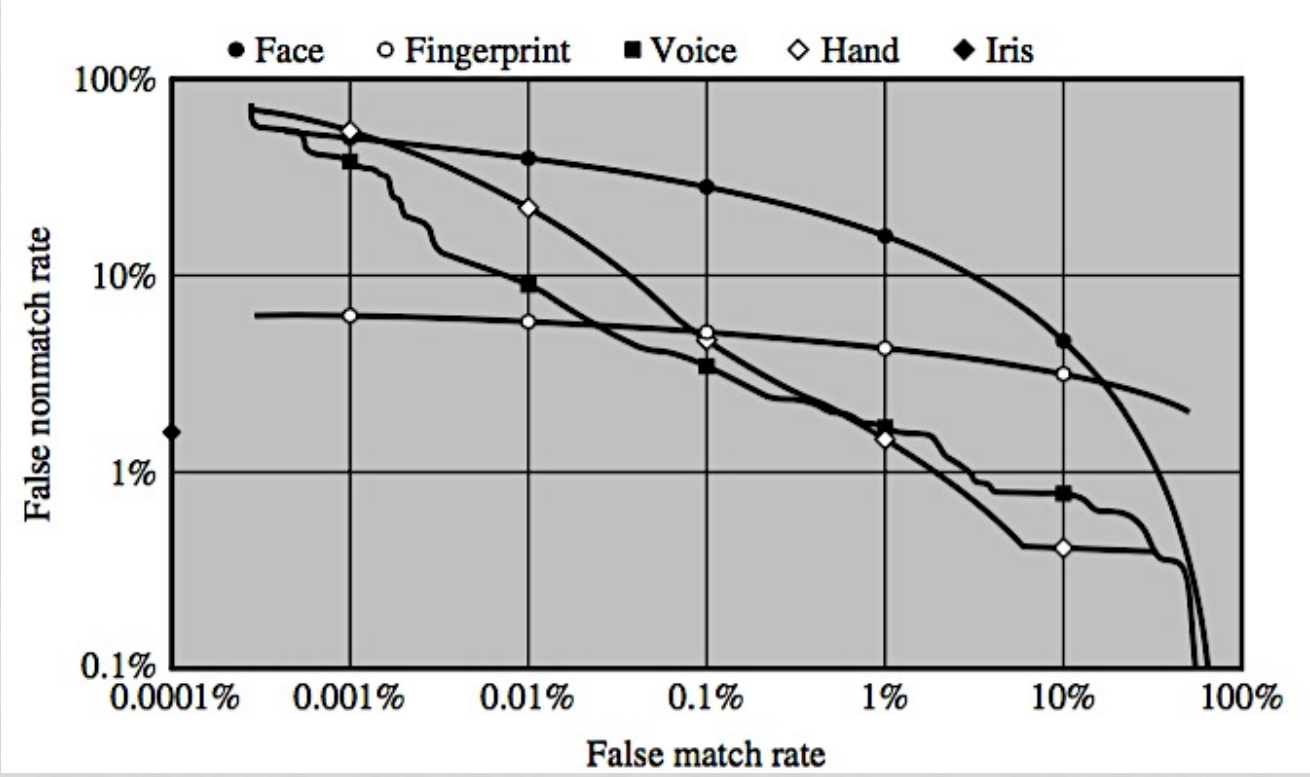
BIOMETRIC ACCURACY

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



BIOMETRIC ACCURACY

- Can plot characteristic curve
- Pick threshold balancing error rates

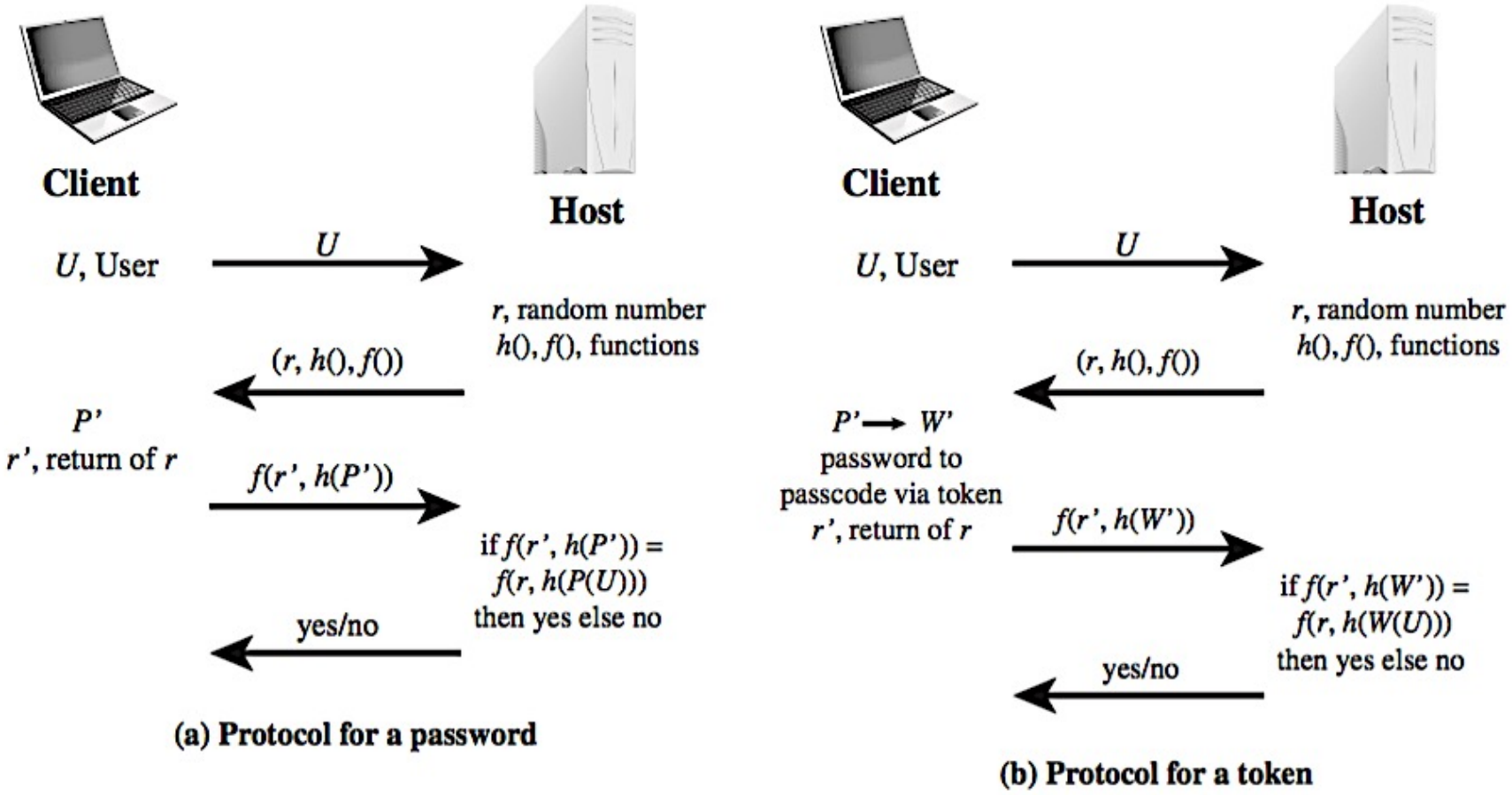


REMOTE USER AUTHENTICATION

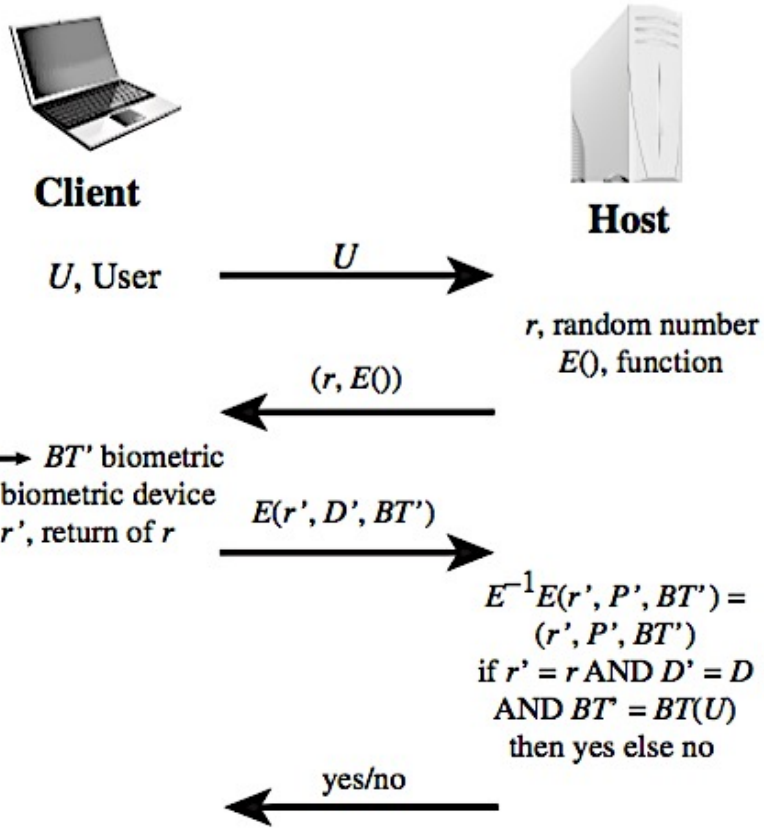
PASSWORD PROTOCOL

- Authentication over network more complex
  - Problems of eavesdropping, replay
- Generally use challenge-response
  - User sends identity
  - Host responds with random number
  - User computes f(r,h(p)) and sends back
  - Host compares value from user with own computed value, if match user authenticated
- Protects against a number of attacks

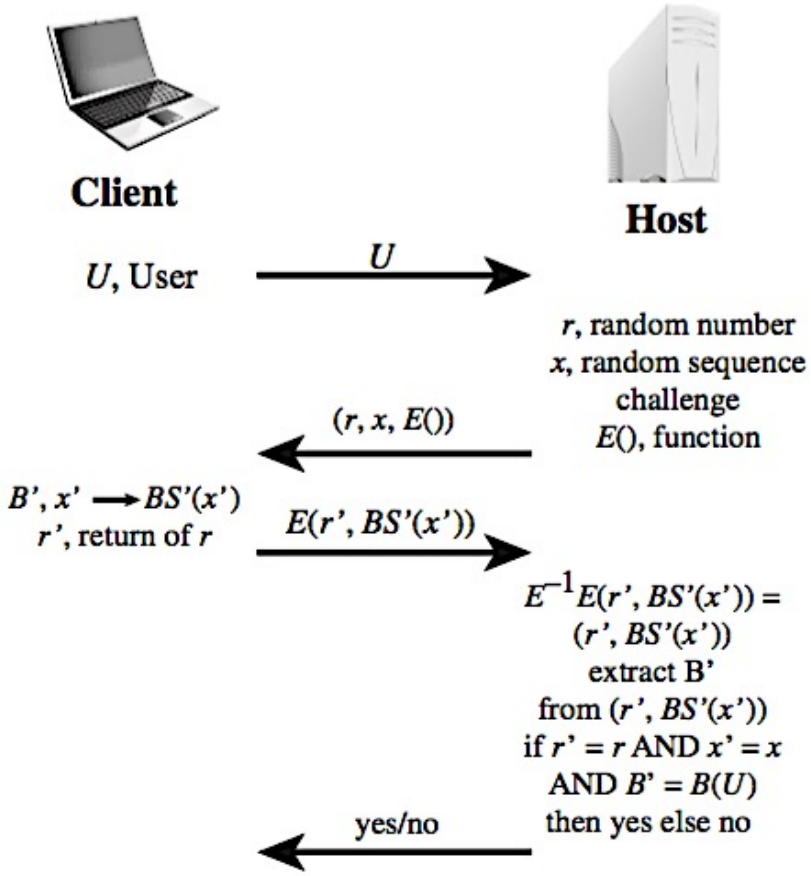
BASIC CHALLENGE-RESPONSE PROTOCOLS FOR REMOTE USER AUTHENTICATION



BASIC CHALLENGE-RESPONSE PROTOCOLS FOR REMOTE USER AUTHENTICATION



(c) Protocol for static biometric

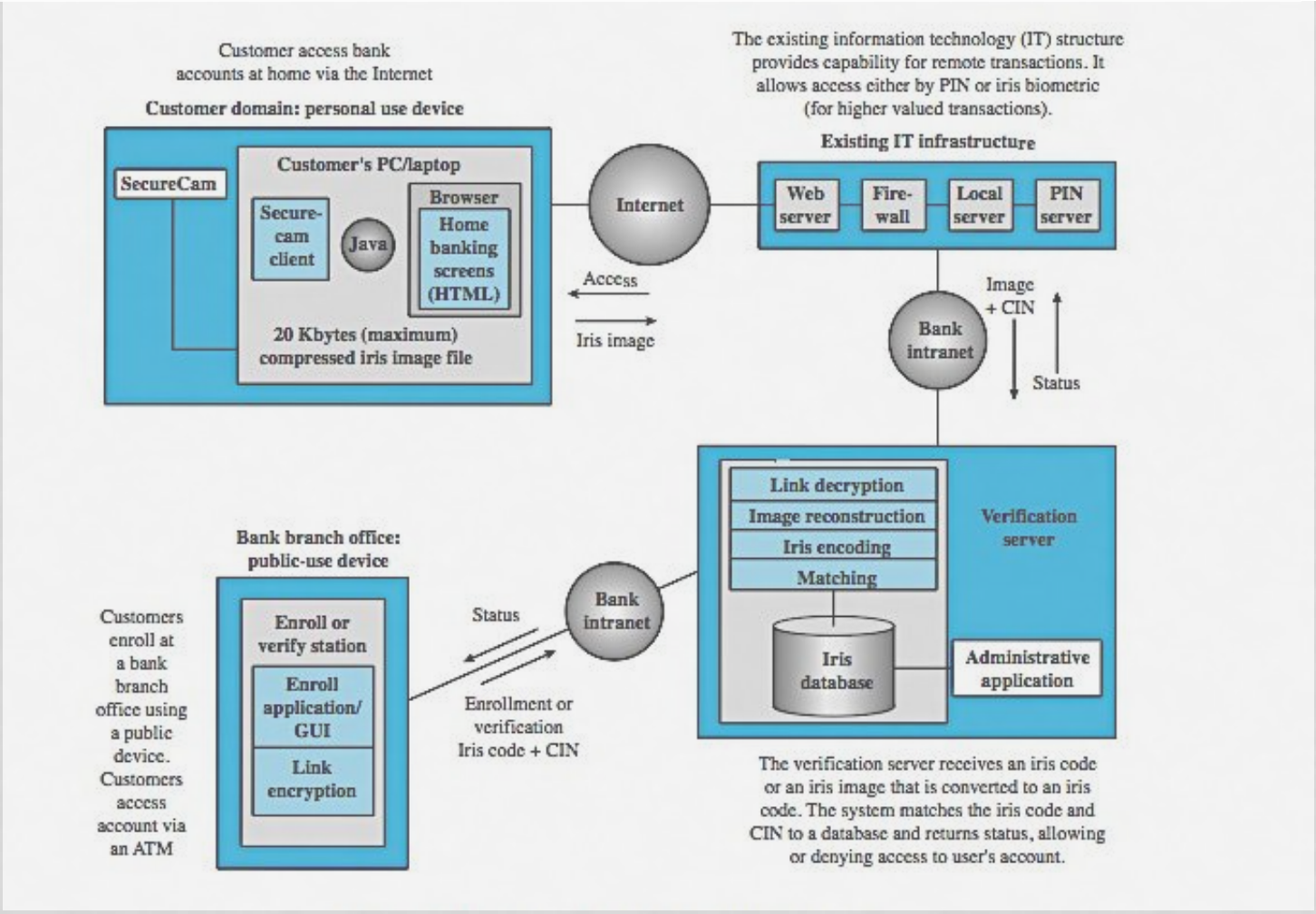


(d) Protocol for dynamic biometric

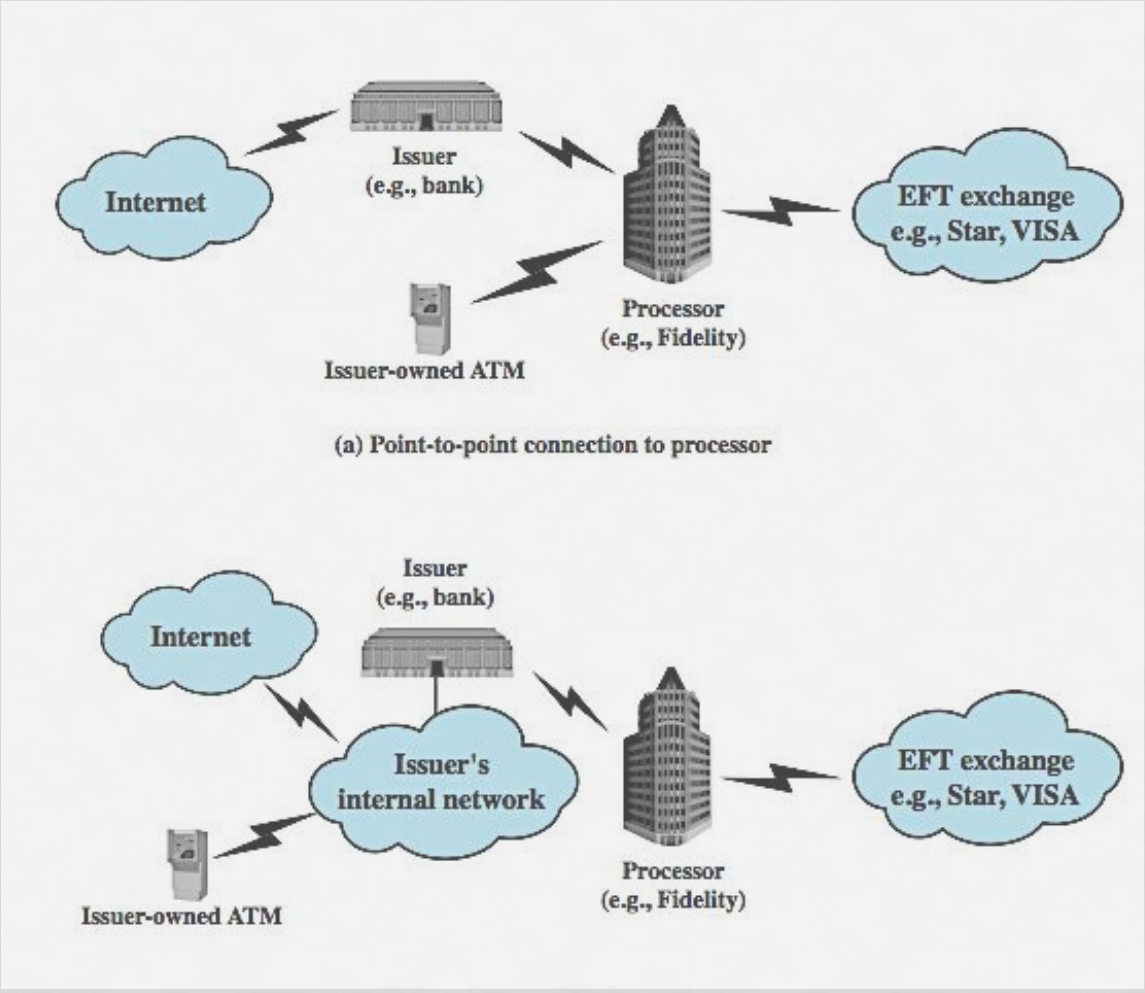
AUTHENTICATION SECURITY ISSUES

- Client Attacks
- Host Attacks
- Eavesdropping
- Replay
- Trojan Horse
- Denial-of-Service

PRACTICAL APPLICATION



CASE STUDY: ATM SECURITY



SUMMARY

- Introduced user authentication
  - Using passwords
  - Using tokens
  - Using biometrics
- Remote user authentication issues
- Example application and case study