Chapter 3

User Authentication

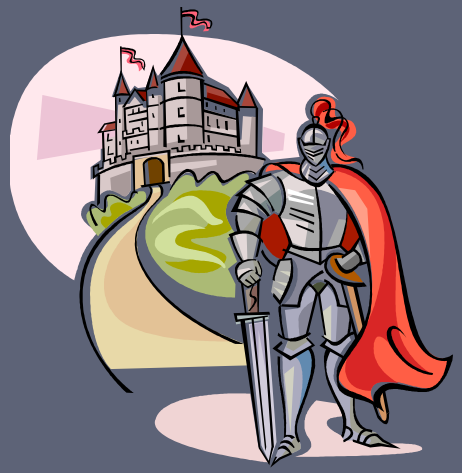
RFC 4949

RFC 4949 defines user authentication as:

"The process of verifying an identity claimed by or for a system entity."

Authentication Process

- Fundamental building block and primary line of defense
- Basis for access control and user accountability



- Identification step
  - Presenting an identifier to the security system
- Verification step
  - Presenting or generating authentication information that corroborates the binding between the entity and the identifier

The four means of authenticating user identity are based on:

Something the individual knows

- Password, PIN, answers to prearranged questions

Something the individual possesses (token)

- Smartcard, electronic keycard, physical key

Something the individual is (static biometrics)

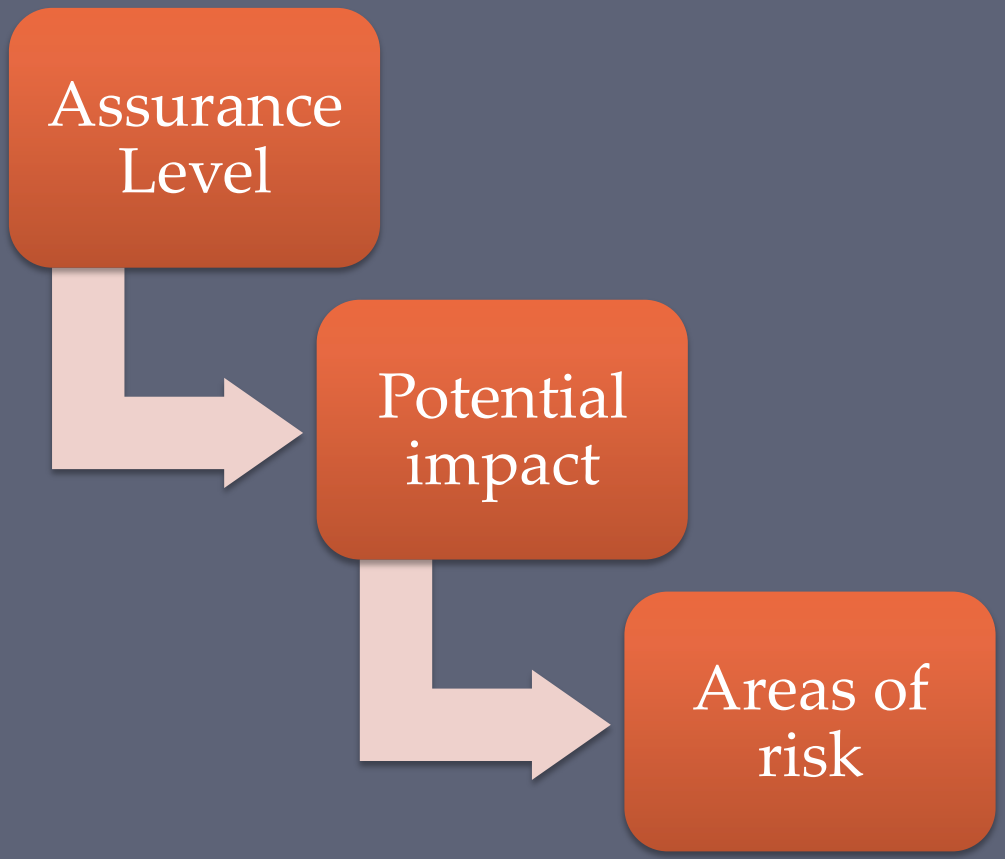
- Fingerprint, retina, face

Something the individual does (dynamic biometrics)

- Voice pattern, handwriting, typing rhythm

Risk Assessment for User Authentication

- There are three separate concepts:



Assurance Level

Describes an organization's degree of certainty that a user has presented a credential that refers to his or her identity

More specifically is defined as:

The degree of confidence in the vetting process used to establish the identity of the individual to whom the credential was issued

The degree of confidence that the individual who uses the credential is the individual to whom the credential was issued

Four levels of assurance

Level 1
- Little or no confidence in the asserted identity's validity

Level 2
- Some confidence in the asserted identity's validity

Level 3
- High confidence in the asserted identity's validity

Level 4
- Very high confidence in the asserted identity's validity

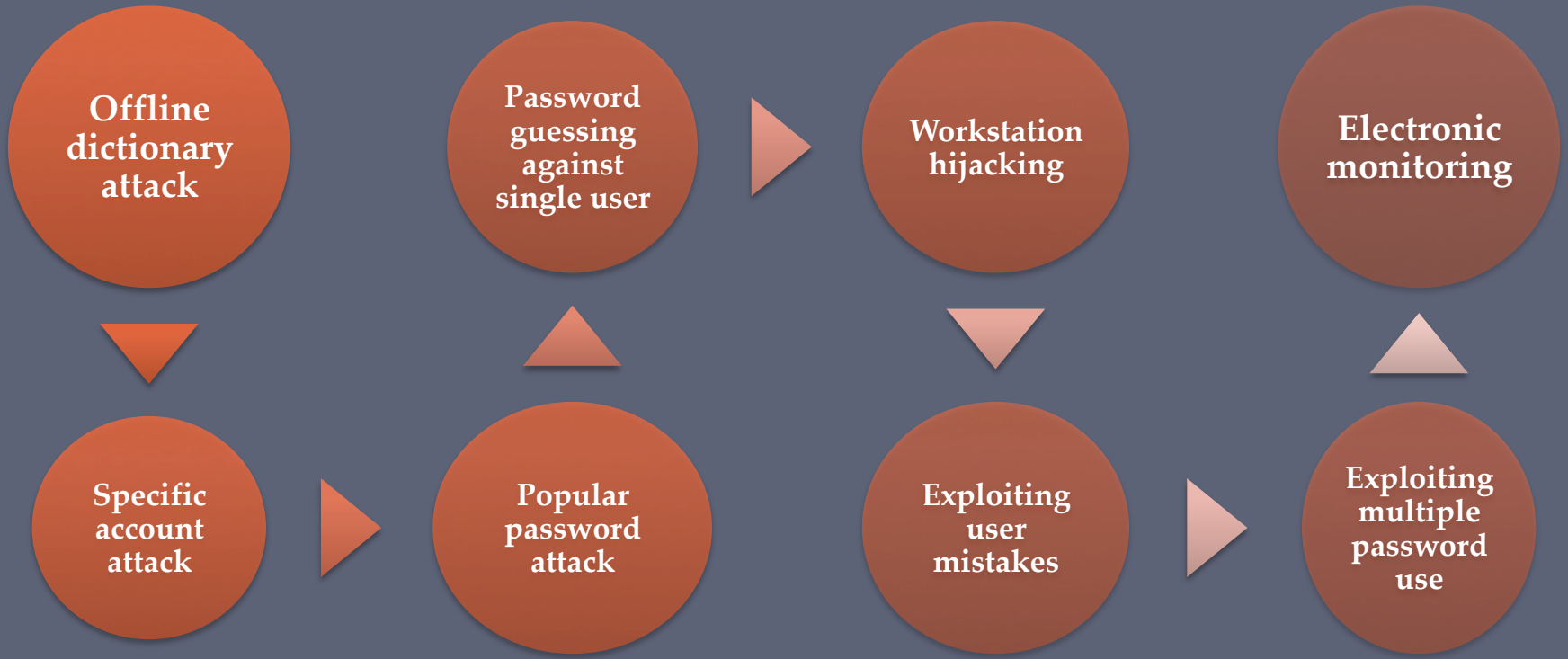
Potential Impact

- FIPS 199 defines three levels of potential impact on organizations or individuals should there be a breach of security:
  - Low
    - An authentication error could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals
  - Moderate
    - An authentication error could be expected to have a serious adverse effect
  - High
    - An authentication error could be expected to have a severe or catastrophic adverse effect

Password Authentication

- Widely used line of defense against intruders
  - User provides name/login and password
  - System compares password with the one stored for that specified login
- The user ID:
  - Determines that the user is authorized to access the system
  - Determines the user's privileges
  - Is used in discretionary access control

Password Vulnerabilities



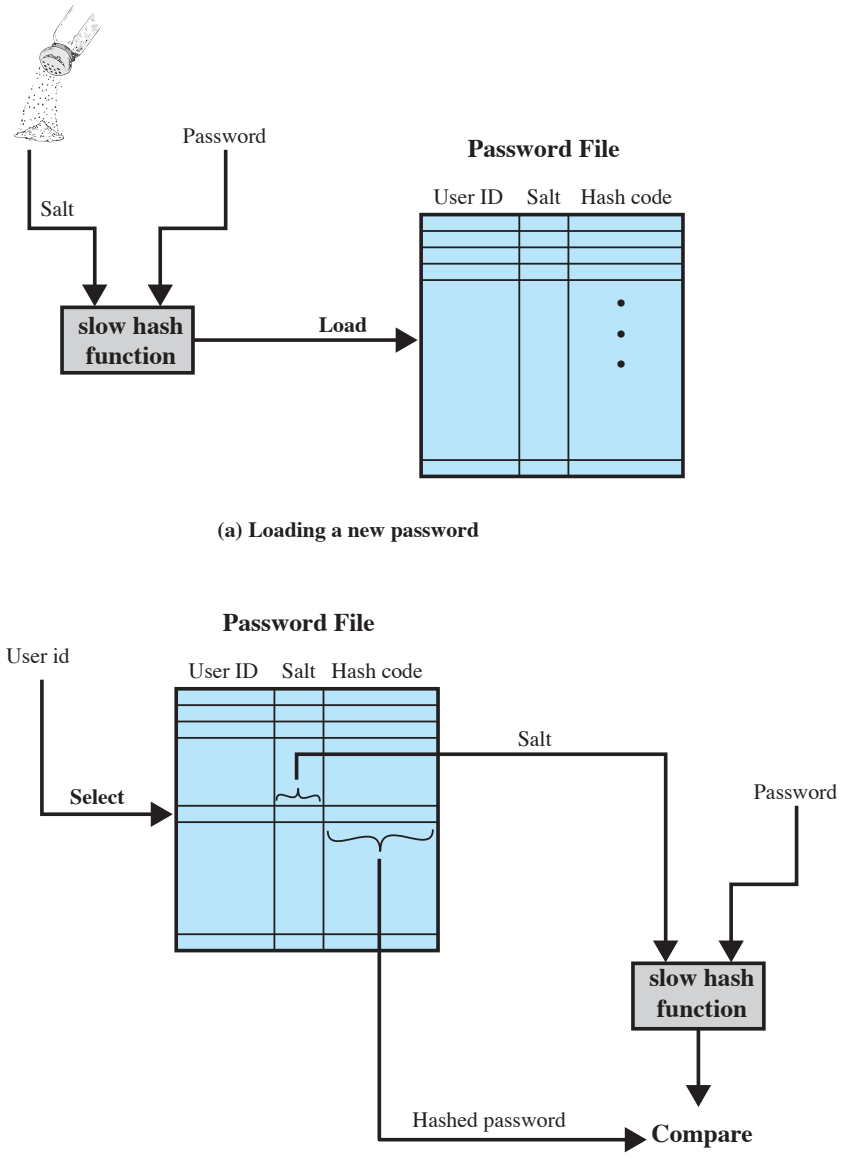
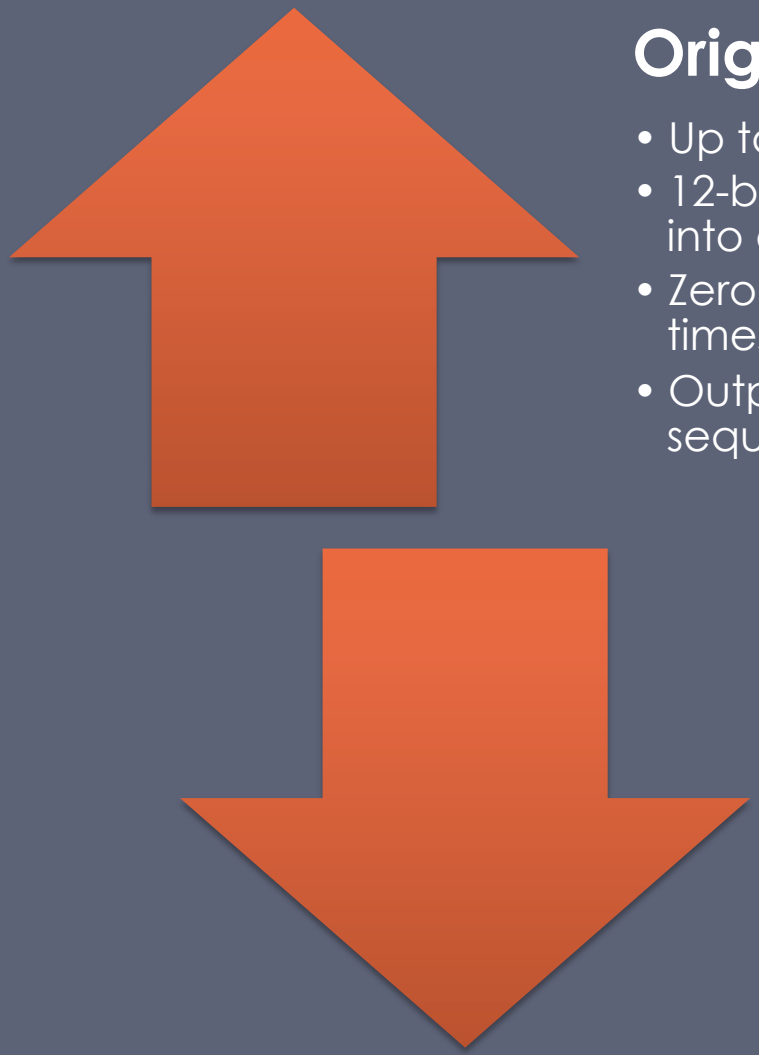


Figure 3.2 UNIX Password Scheme

UNIX Implementation



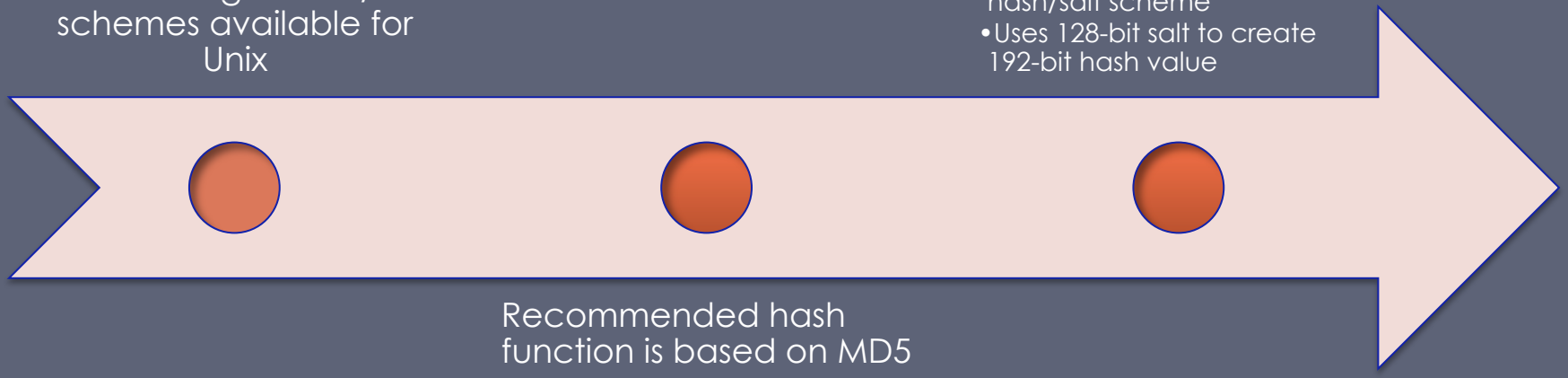
Original scheme

- Up to eight printable characters in length
- 12-bit salt used to modify DES encryption into a one-way hash function
- Zero value repeatedly encrypted 25 times
- Output translated to 11 character sequence

Now regarded as inadequate

- Still often required for compatibility with existing account management software or multivendor environments

Improved Implementations



Much stronger hash/salt schemes available for Unix

Recommended hash function is based on MD5

- Salt of up to 48-bits
- Password length is unlimited
- Produces 128-bit hash
- Uses an inner loop with 1000 iterations to achieve slowdown

OpenBSD uses Blowfish block cipher based hash algorithm called Bcrypt

- Most secure version of Unix hash/salt scheme
- Uses 128-bit salt to create 192-bit hash value

Password Cracking

Dictionary attacks

- Develop a large dictionary of possible passwords and try each against the password file
- Each password must be hashed using each salt value and then compared to stored hash values

Rainbow table attacks

- Pre-compute tables of hash values for all salts
- A mammoth table of hash values
- Can be countered by using a sufficiently large salt value and a sufficiently large hash length

Password crackers exploit the fact that people choose easily guessable passwords

- Shorter password lengths are also easier to crack

John the Ripper

- Open-source password cracker first developed in in 1996
- Uses a combination of brute-force and dictionary techniques

Modern Approaches

- Complex password policy
  - Forcing users to pick stronger passwords
- However password-cracking techniques have also improved
  - The processing capacity available for password cracking has increased dramatically
  - A PC running a single AMD Radeon HD7970 GPU, for instance, can try on average an $8.2 * 10^9$ password combinations each second
  - The use of sophisticated algorithms to generate potential passwords
  - Studying examples and structures of actual passwords in use

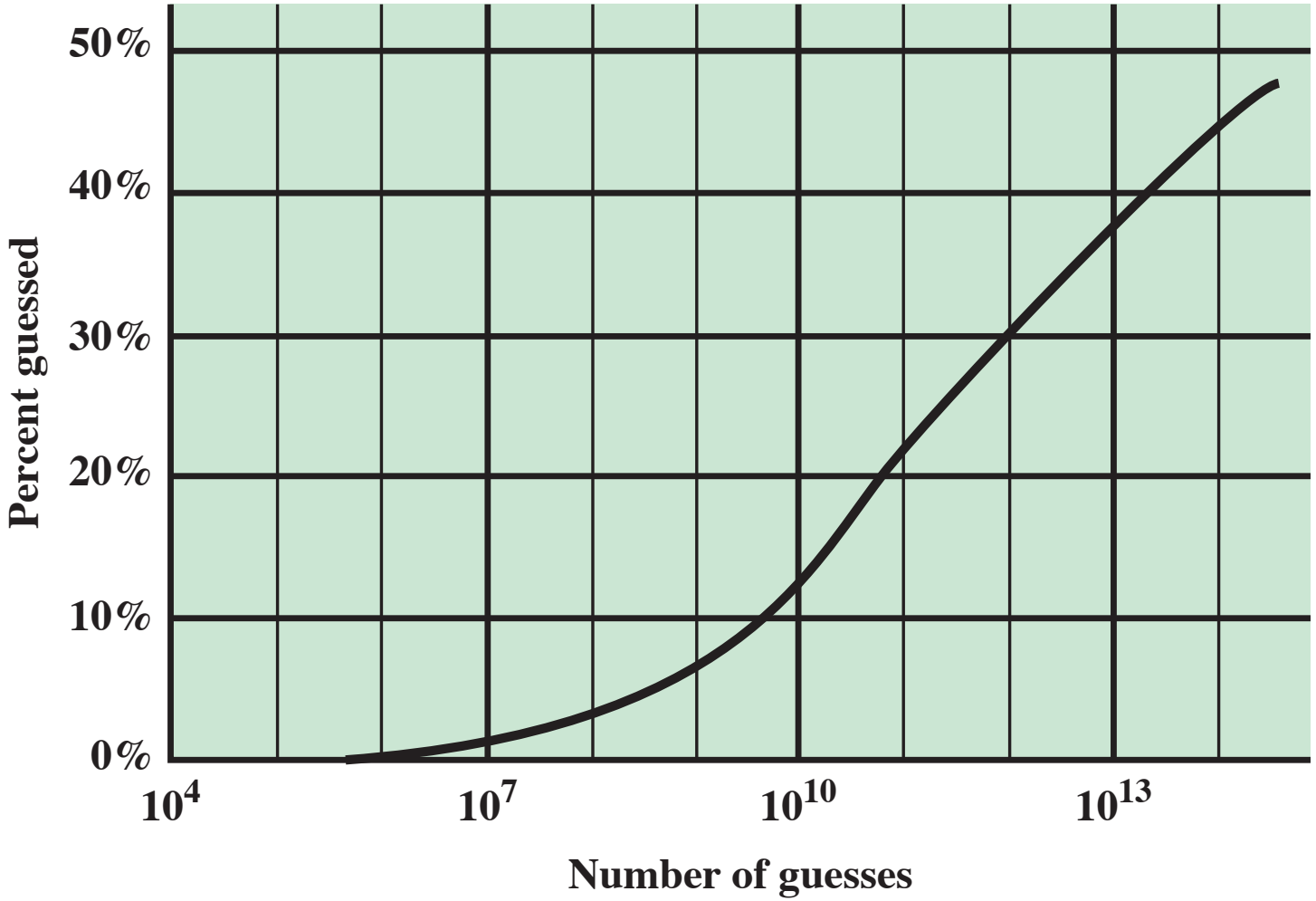


Figure 3.3 The Percentage of Passwords Guessed After a Given Number of Guesses

Password Resilience to Cracking

- Define a metric of how well the system can withstand dictionary attacks
  - Could be function of time
- Anderson's formula
  - Information Security in a Multi-User Computing Environment, Advances in computers 1972

Password Resilience to Cracking

- We want to estimate *P*, where *P* is the probability of guessing a password in specified period of time given:
  - *G* number of guesses tested in 1 time unit
  - *T* number of time units
  - *N* number of possible passwords (| *A* |)
- We can estimate

$$P \geq \frac{TG}{N}$$

Example

- Passwords drawn from a 96-char alphabet
- Can test 10^4 guesses per second
- Probability of a success to be 0.5 over a 365 day period
- What is minimum password length?
- Solution
  - $N \geq TG/P = (365 \times 24 \times 60 \times 60) \times 10^4 / 0.5 = 6.31 \times 10^{11}$
  - Choose s such that $96^j \geq 6.31 \times 10^{11}$
  - So $j \geq 5.95$, meaning passwords must be at least 6 chars long

Assumption

- Time needed to test a password is constant
  - Reasonable (average time to test a password will do)
- All passwords are equally likely to be selected
  - Not necessarily true!

Password File Access Control

Can block offline guessing attacks by denying access to encrypted passwords

- Make available only to privileged users
- Shadow password file

Vulnerabilities

- Weakness in the OS that allows access to the file
- Accident with permissions making it readable
- Users with same password on other systems
- Access from backup media
- Sniff passwords in network traffic

Password Selection Strategies

User education

Users can be told the importance of using hard to guess passwords and can be provided with guidelines for selecting strong passwords

Computer generated passwords

Users have trouble remembering them

Reactive password checking

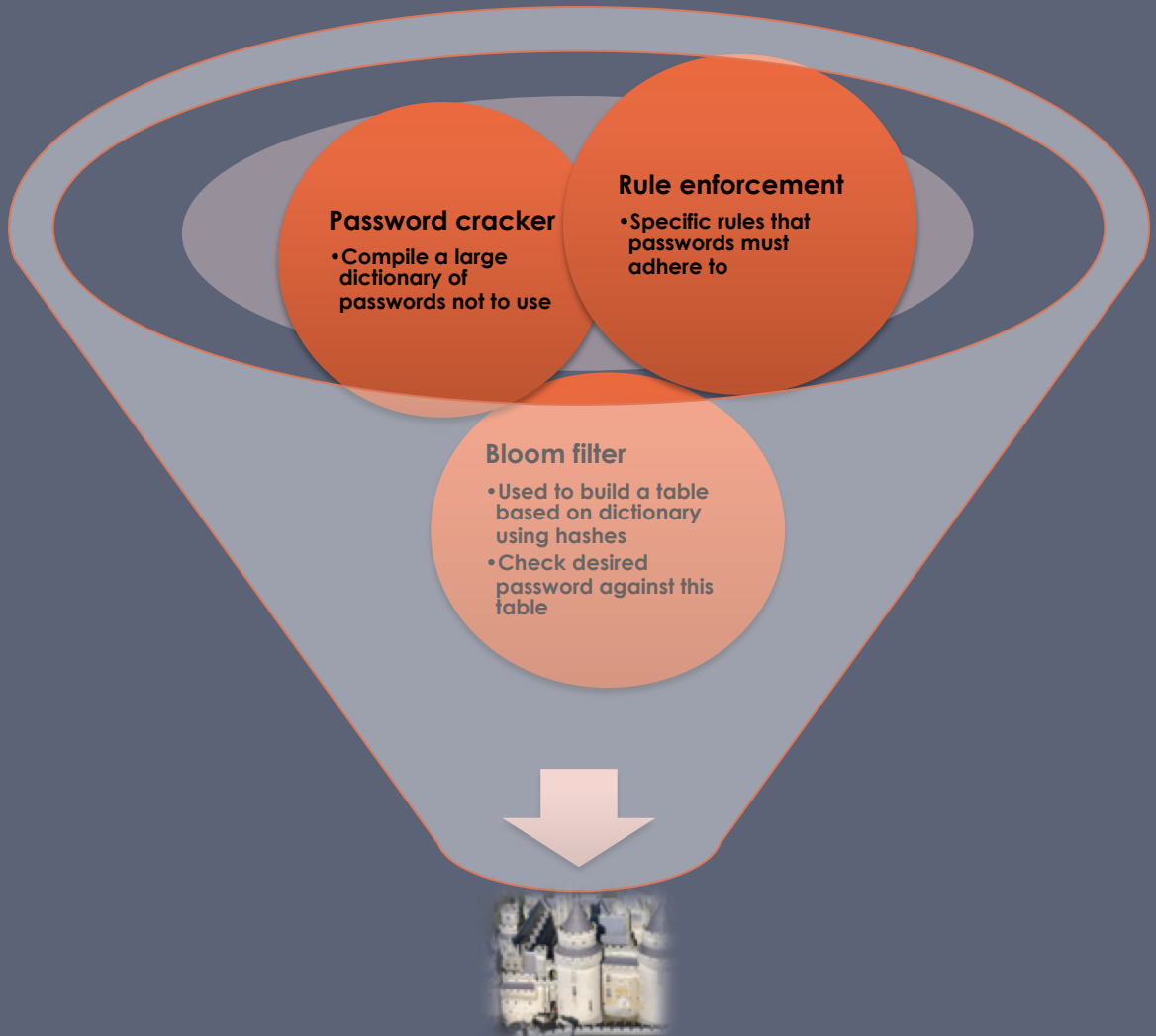
System periodically runs its own password cracker to find guessable passwords

Complex password policy

User is allowed to select their own password, however the system checks to see if the password is allowable, and if not, rejects it

Goal is to eliminate guessable passwords while allowing the user to select a password that is memorable

Proactive Password Checking



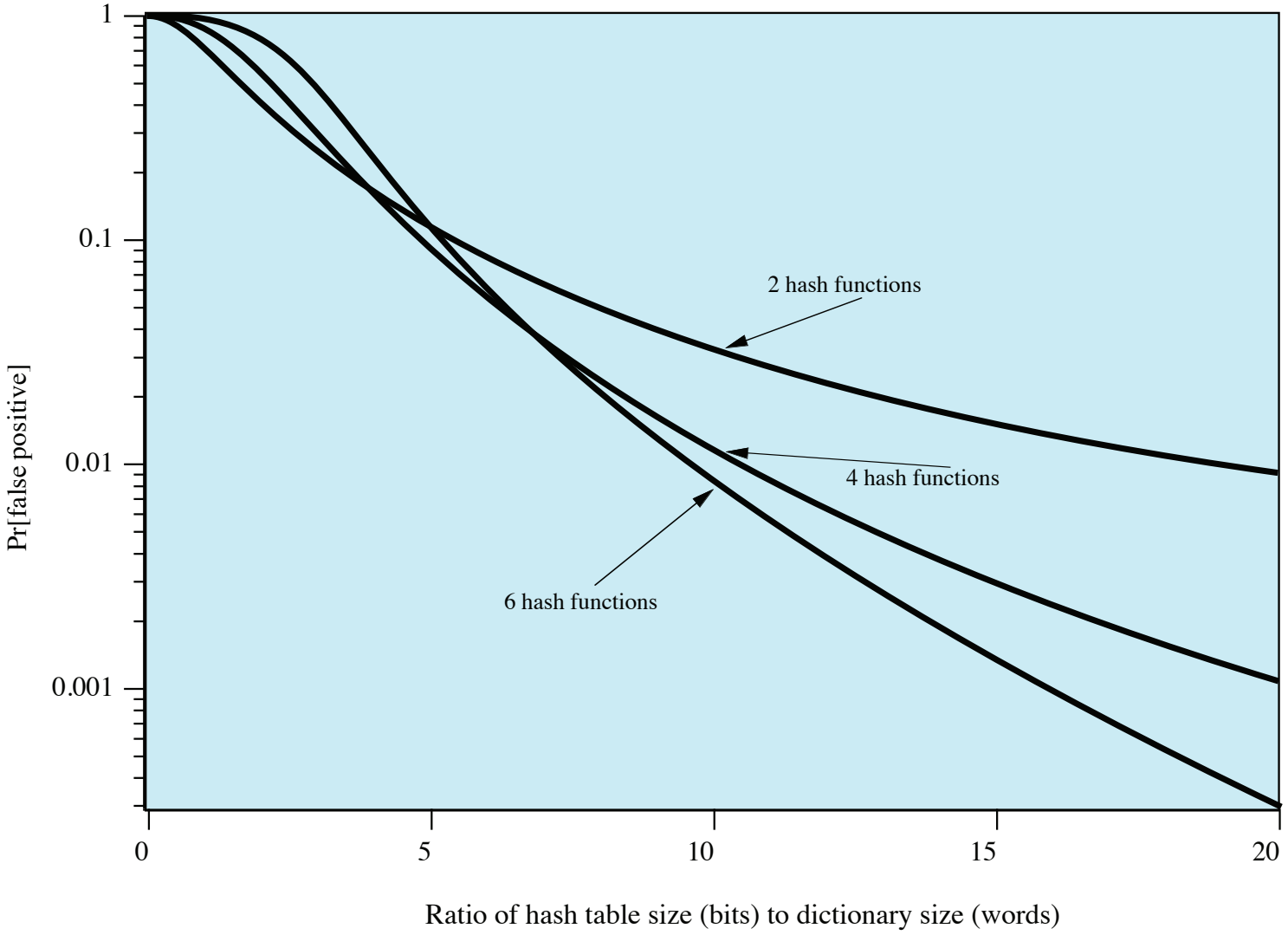


Figure 3.4 Performance of Bloom Filter

Bloom Filters

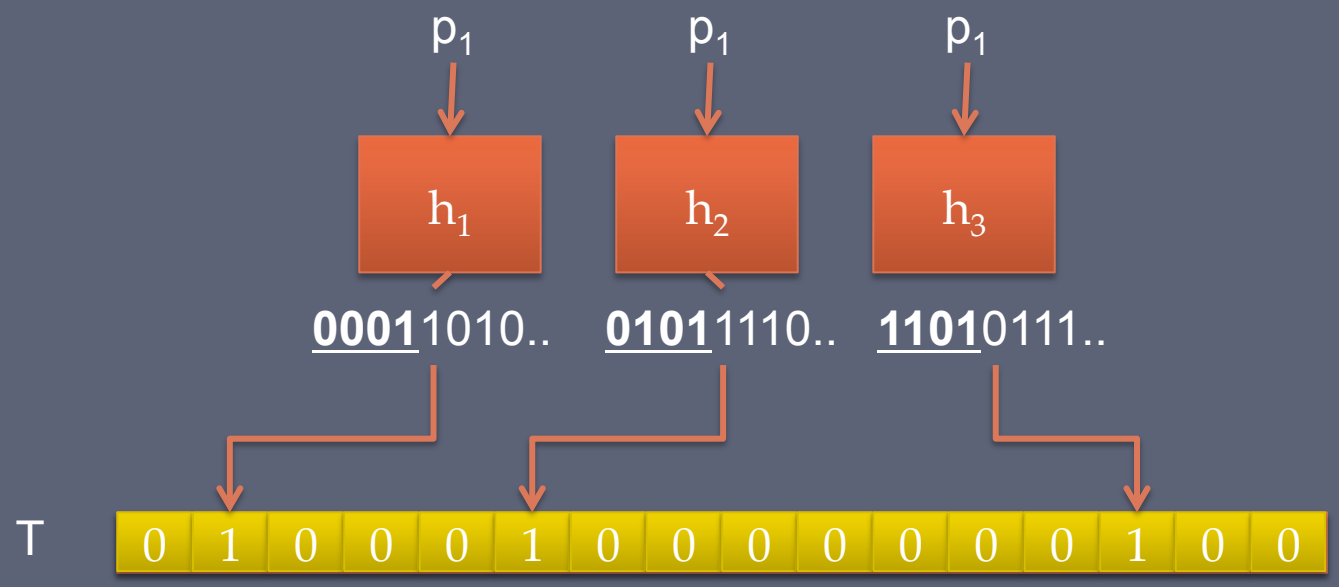
- Filter of order *k* has *k* indepenedent hash functions
  - $H_i(x),\ 1 <= i <= k$
  - Each hash function maps x to a value [0, N-1]
- Given a dictionary of *D* words
- A table T of 1 x *N* is defined
  - T(N) is value at column N
- For each given word *w* in the dictionary *D*, calculate
  - $w_i = H_i(w),\ 1 <= i <= k$
  - Set $T(w_i) = 1$

Bloom Filters: Bootstrapping

- Dictionary file: p_1, p_2, p_3
- 3 Hashes: h_1, h_2, h_3
- N=16

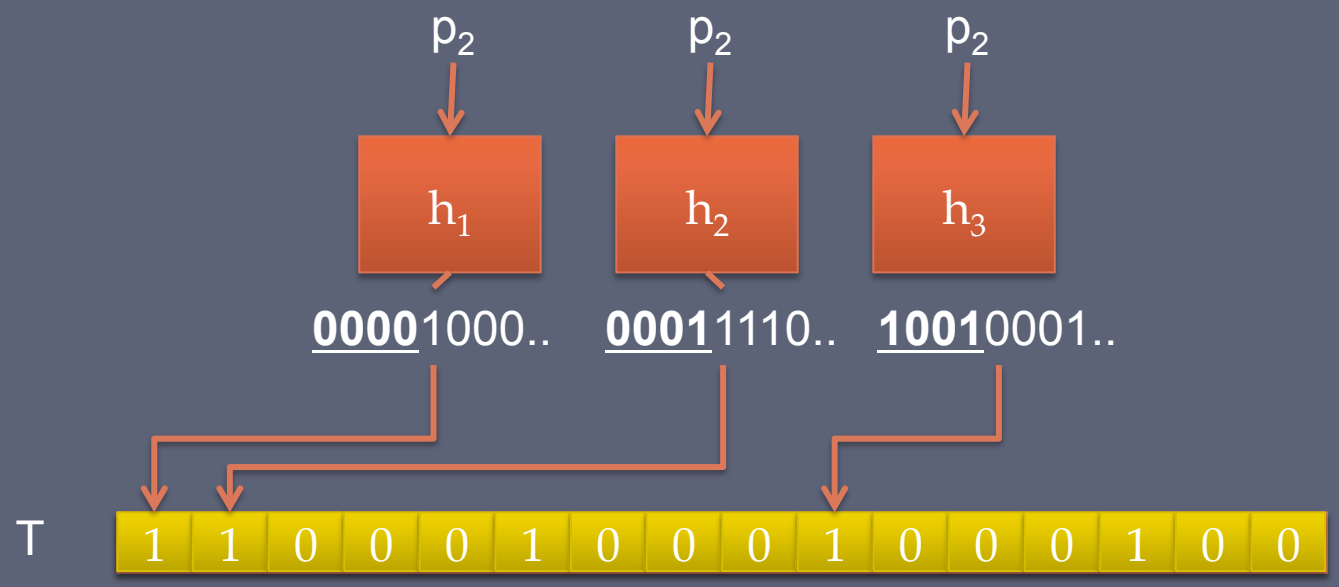
Bloom Filters: Bootstrapping

- Dictionary file: p_1, p_2, p_3



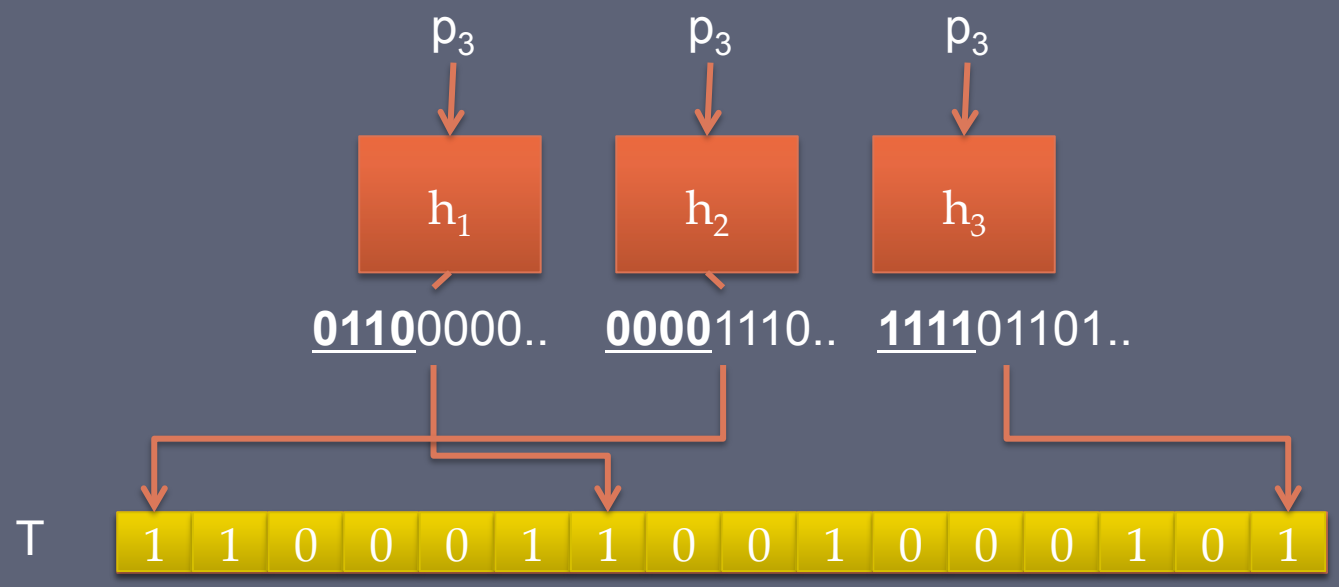
Bloom Filters: Bootstrapping

- Dictionary file: p_1, p_2, p_3



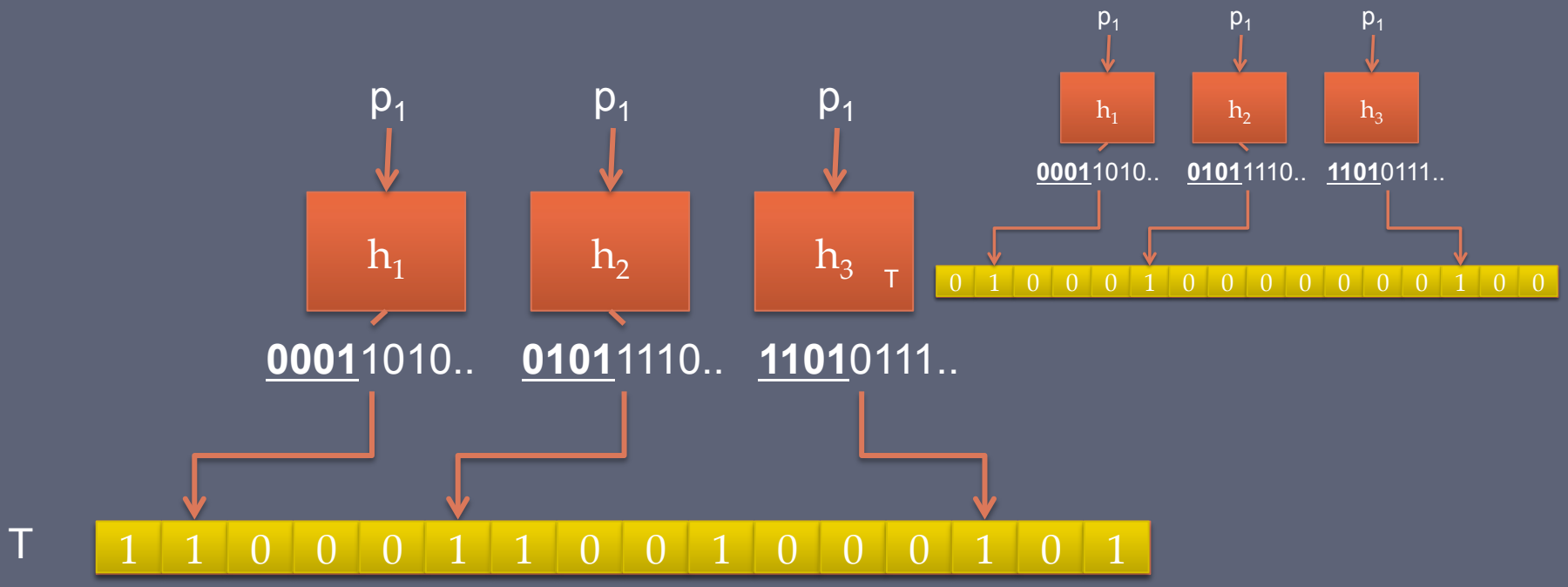
Bloom Filters: Bootstrapping

- Dictionary file: p_1, p_2, p_3



Bloom Filters: Operation

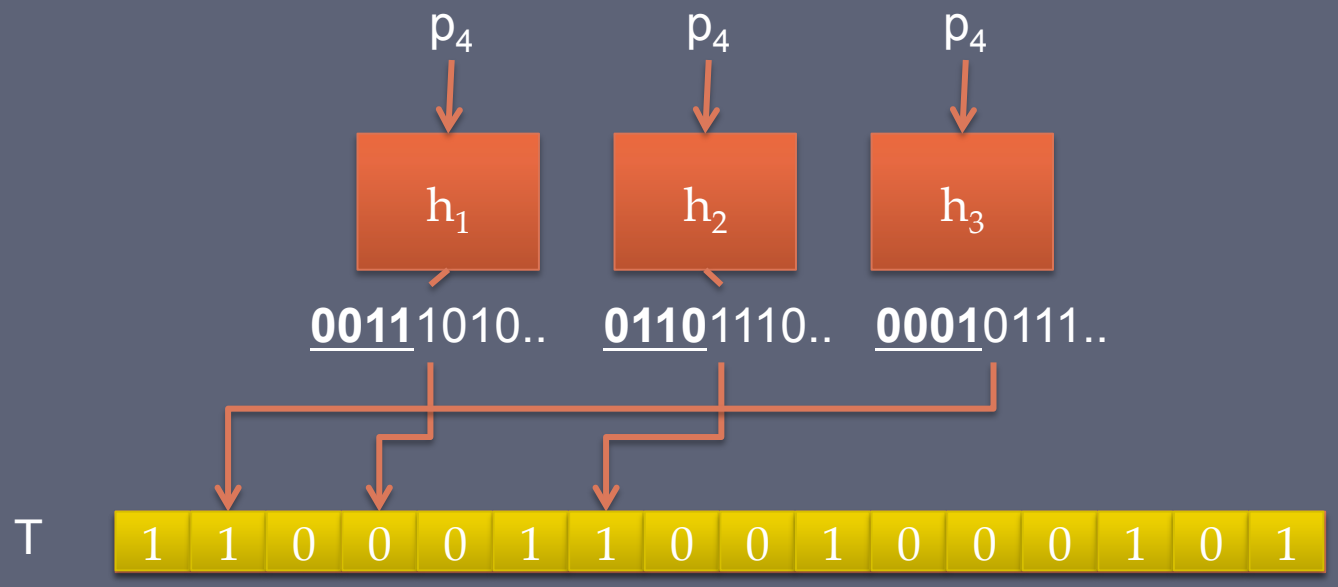
- User supplied passwords: p_1, p_4, p_5



REJECT

Bloom Filters: Operation

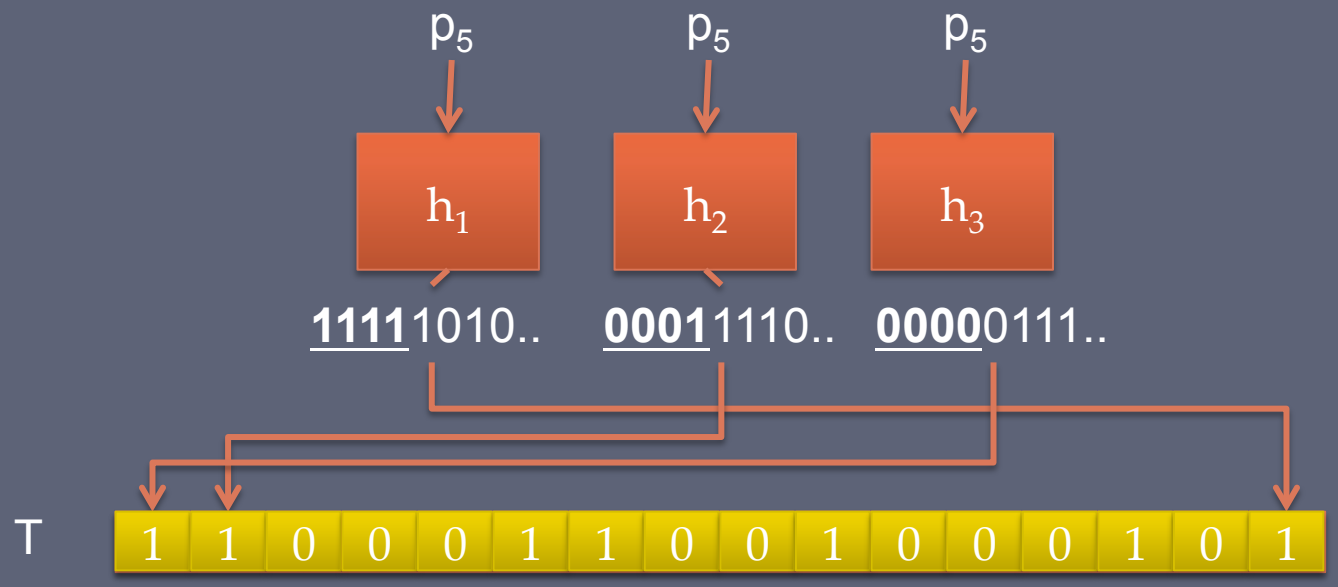
- User supplied passwords: p_1, p_4, p_5



ACCEPT

Bloom Filters: Operation

- User supplied passwords: p_1, p_4, p_5



REJECT

Bloom Filters

- Probability of false positives estimation
  - $N/D \sim= -k / \ln (1 - p^{1/k})$
  - N: number of columns in T (i.e. number of bits in hash table)
  - D: number of words in dictionary
  - k: number of hash functions
  - p: probability of false positives
- Example:
  - D: 1 million words (if words are 5 letters in average, 5MBytes)
  - p: 0.01
  - k: 6
  - What value of N to use?
  - $N = (-6 / \ln(1 - 0.01^{1/6}))* 10^6$
  - $9.6 * 10^6$ = 9.6 Mbits = 1.2 MBytes

Don’t use the first function in the book

Challenge-Response

- Passwords are reusable (until they expire). Once found, an attacker can use for the duration of the password.
- The idea of Challenge-response is to change the password every time the user logs in.

S/Key

- One-time password scheme
- *h* one-way hash function
- User chooses initial seed *k*
- System calculates:

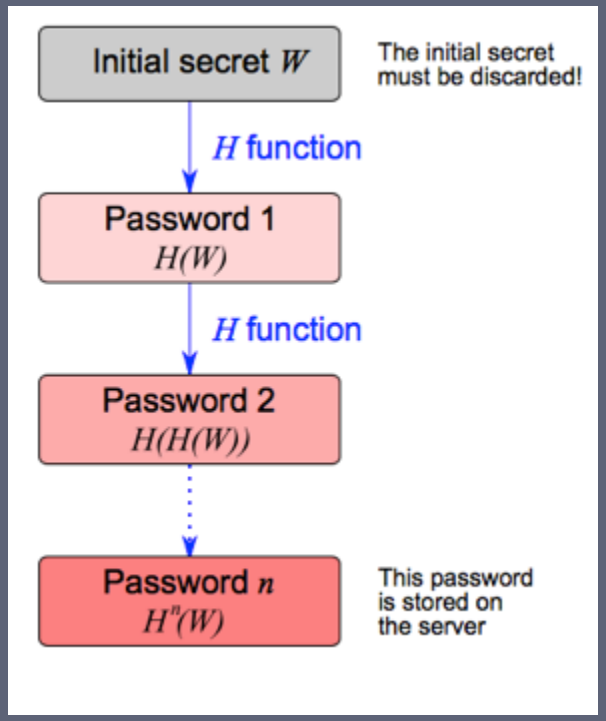
$$h(k) = k_1,\ h(k_1) = k_2,\ \ldots,\ h(k_{n-1}) = k_n$$

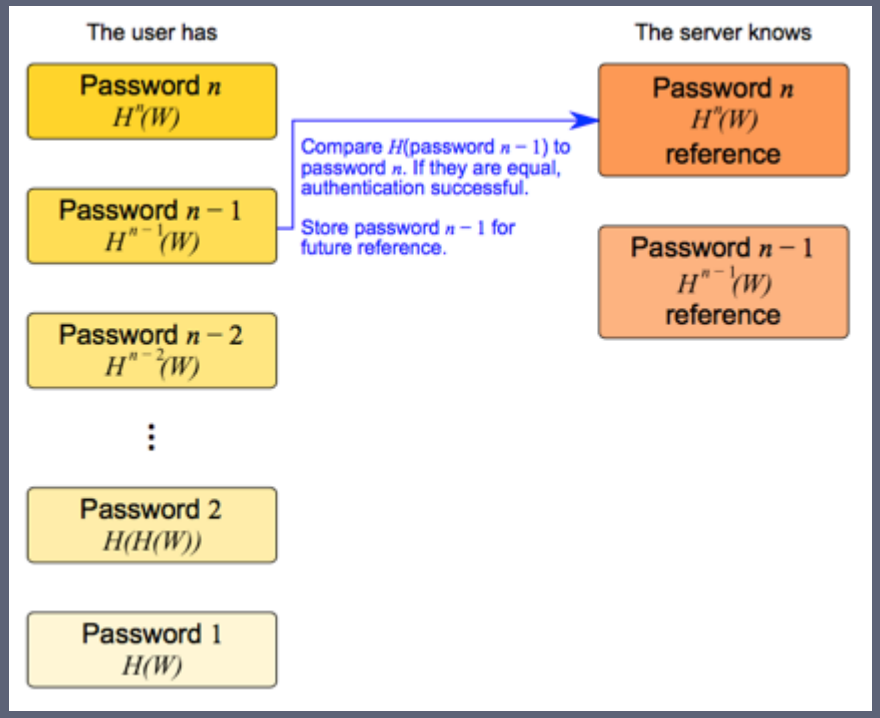
- Passwords are reverse order:

$$p_1 = k_n,\ p_2 = k_{n-1},\ \ldots,\ p_{n-1} = k_2,\ p_n = k_1$$

S/Key

- User authentication to server
  - User has all passwords
  - Server only has Password n
- User discloses passwords in reverse order





https://en.wikipedia.org/wiki/S/KEY

Table 3.2

Card Type	Defining Feature	Example
Embossed	Raised characters only, on front	Old credit card
Magnetic stripe	Magnetic bar on back, characters on front	Bank card
Memory	Electronic memory inside	Prepaid phone card
Smart Contact Contactless	Electronic memory and processor inside Electrical contacts exposed on surface Radio antenna embedded inside	Biometric ID card

Types of Cards Used as Tokens

Memory Cards

- Can store but do not process data
- The most common is the magnetic stripe card
- Can include an internal electronic memory
- Can be used alone for physical access
  - Hotel room
  - ATM
- Provides significantly greater security when combined with a password or PIN
- Drawbacks of memory cards include:
  - Requires a special reader
  - Loss of token
  - User dissatisfaction

Smart Tokens

- Physical characteristics:
  - Include an embedded microprocessor
  - A smart token that looks like a bank card
  - Can look like calculators, keys, small portable objects

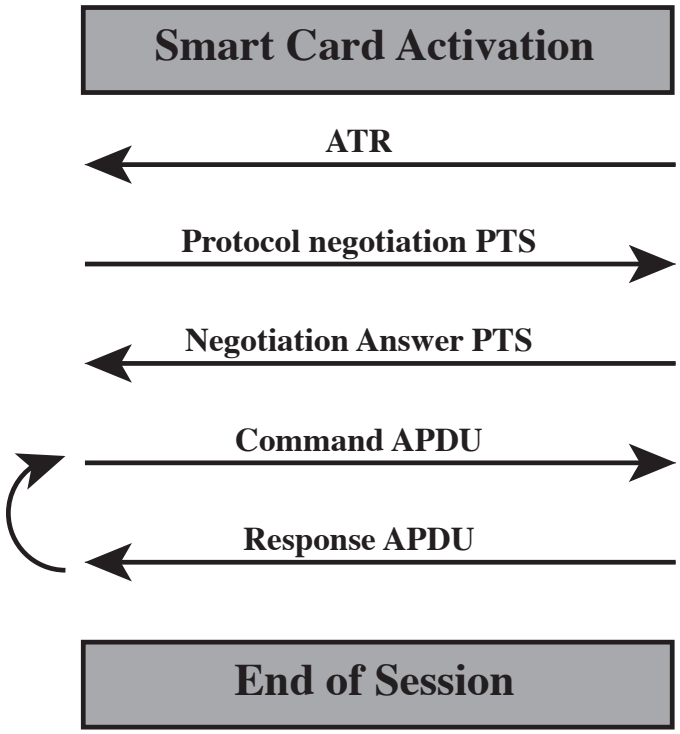
- Interface:
  - Manual interfaces include a keypad and display for interaction
  - Electronic interfaces communicate with a compatible reader/ writer
- Authentication protocol:
  - Classified into three categories:
    - Static
    - Dynamic password generator
    - Challenge-response

Smart Cards

- Most important category of smart token
  - Has the appearance of a credit card
  - Has an electronic interface
  - May use any of the smart token protocols
- Contain:
  - An entire microprocessor
    - Processor
    - Memory
    - I/O ports
- Typically include three types of memory:
  - Read-only memory (ROM)
    - Stores data that does not change during the card's life
  - Electrically erasable programmable ROM (EEPROM)
    - Holds application data and programs
  - Random access memory (RAM)
    - Holds temporary data generated when applications are executed

Smart card

Card reader



APDU = application protocol data unit
ATR = Answer to reset
PTS = Protocol type selection

Figure 3.5 Smart Card/Reader Exchange

Biometric Authentication

- Attempts to authenticate an individual based on unique physical characteristics
- Based on pattern recognition
- Is technically complex and expensive when compared to passwords and tokens
- Physical characteristics used include:
  - Facial characteristics
  - Fingerprints
  - Hand geometry
  - Retinal pattern
  - Iris
  - Signature
  - Voice

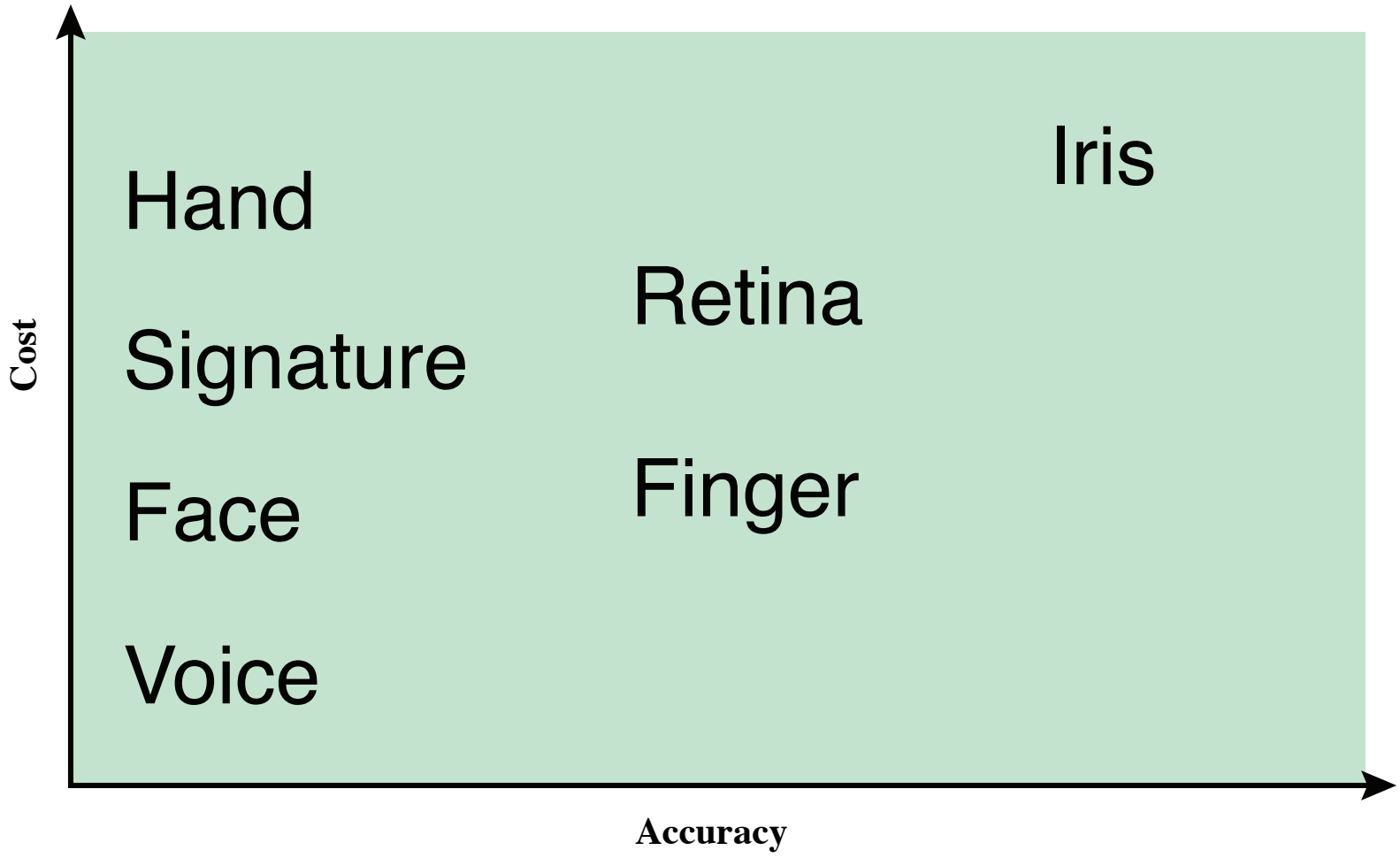


Figure 3.7 Cost Versus Accuracy of Various Biometric Characteristics in User Authentication Schemes.

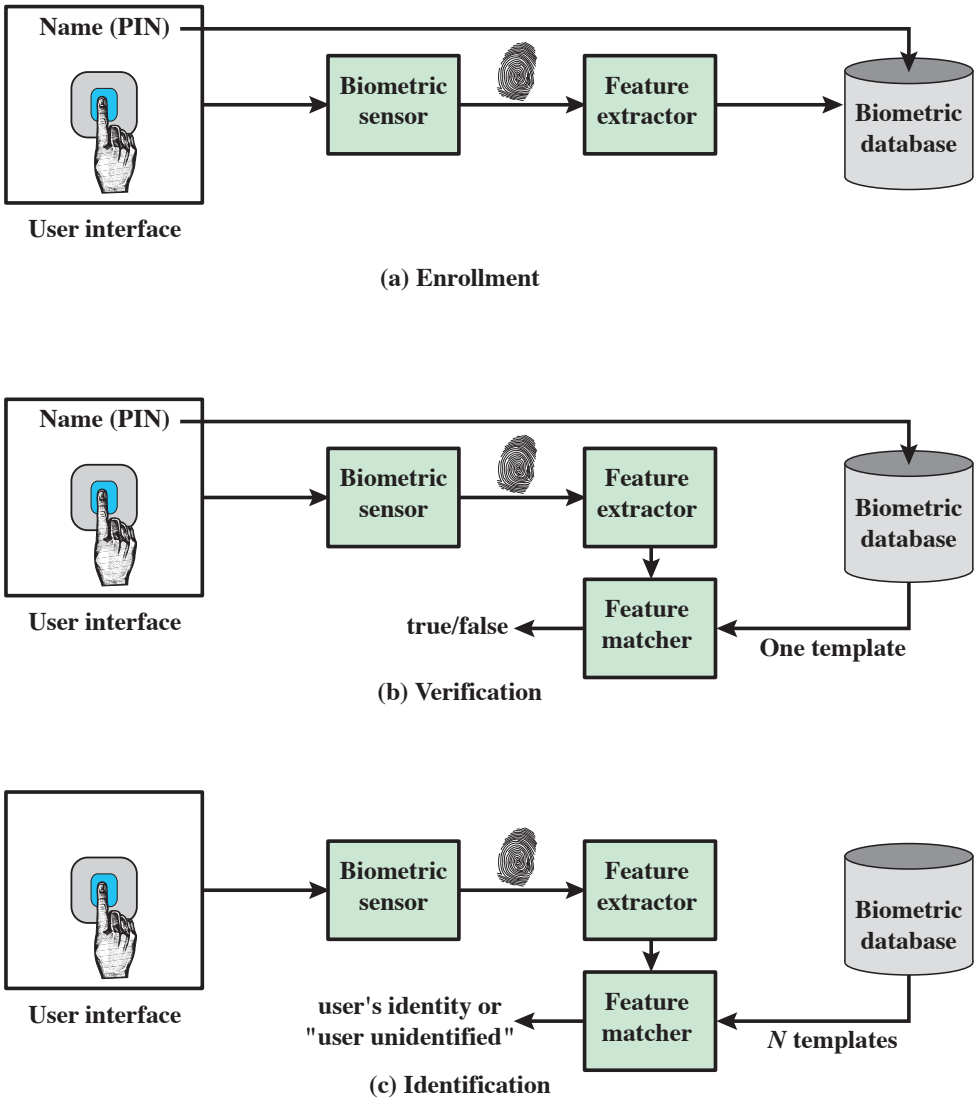


Figure 3.8 A Generic Biometric System. Enrollment creates an association between a user and the user's biometric characteristics. Depending on the application, user authentication either involves verifying that a claimed user is the actual user or identifying an unknown user.

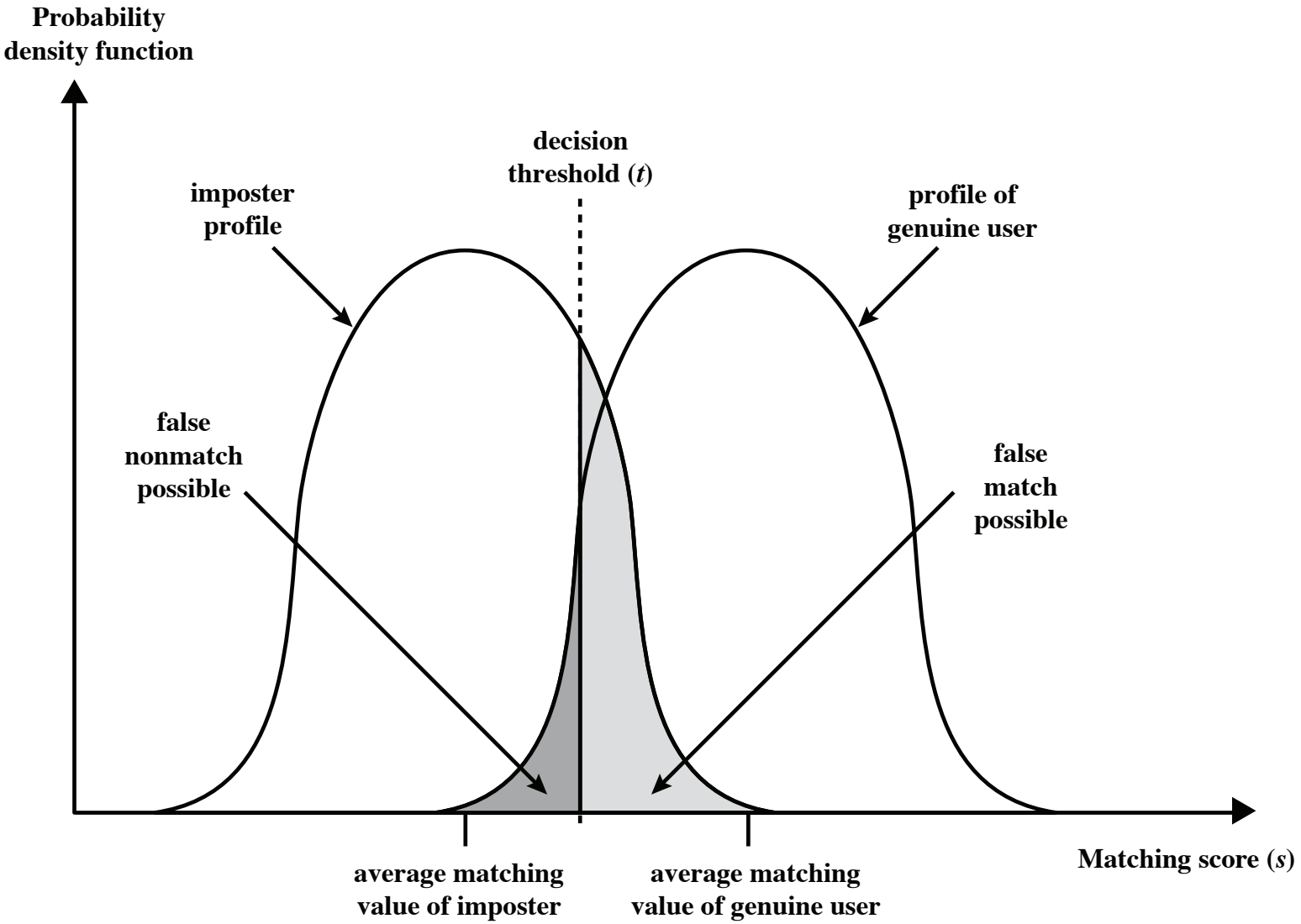


Figure 3.9 Profiles of a Biometric Characteristic of an Imposter and an Authorized Users In this depiction, the comparison between presented feature and a reference feature is reduced to a single numeric value. If the input value (s) is greater than a preassigned threshold (t), a match is declared.

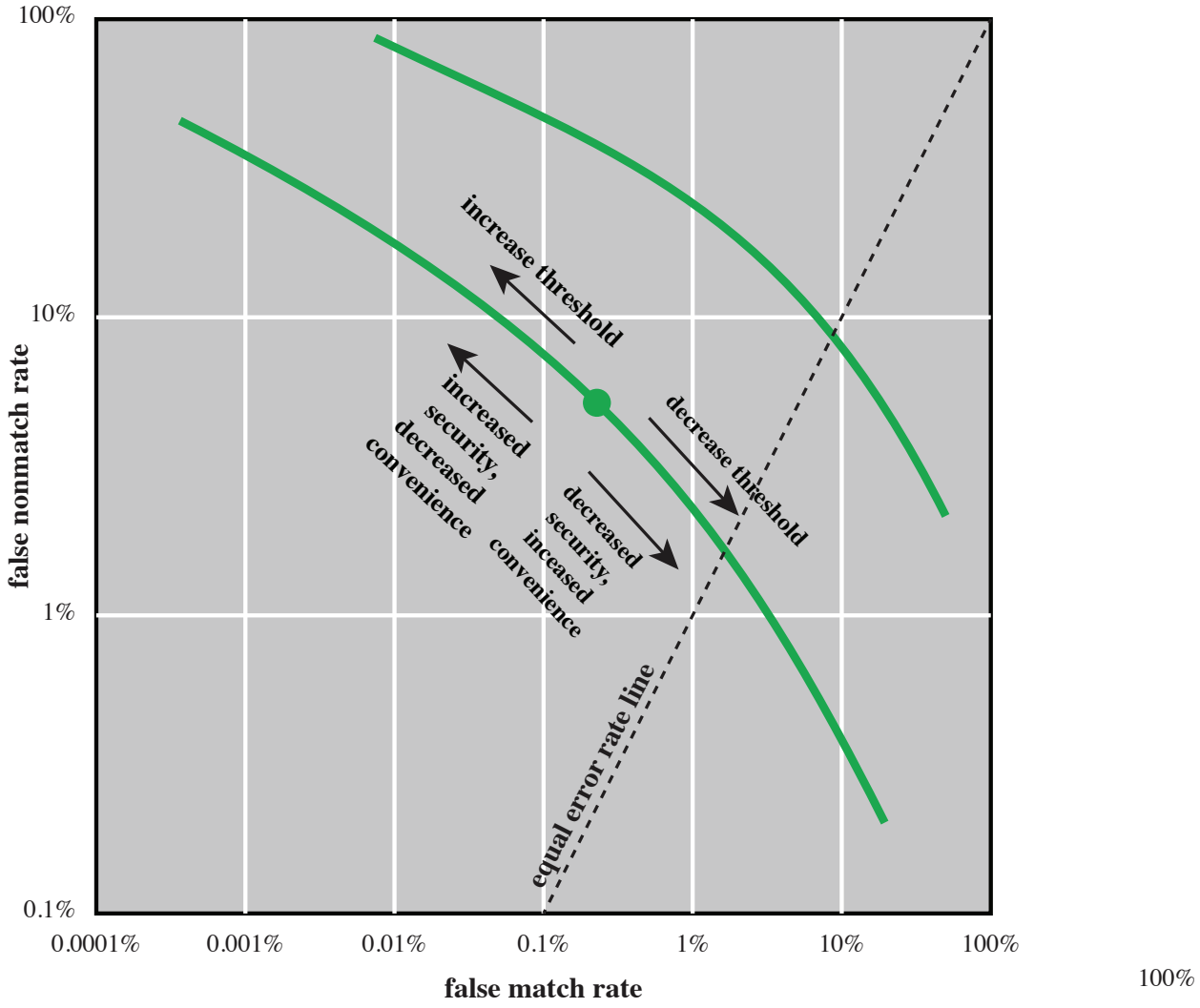


Figure 3.10 Idealized Biometric Measurement Operating Characteristic Curves (log-log scale)

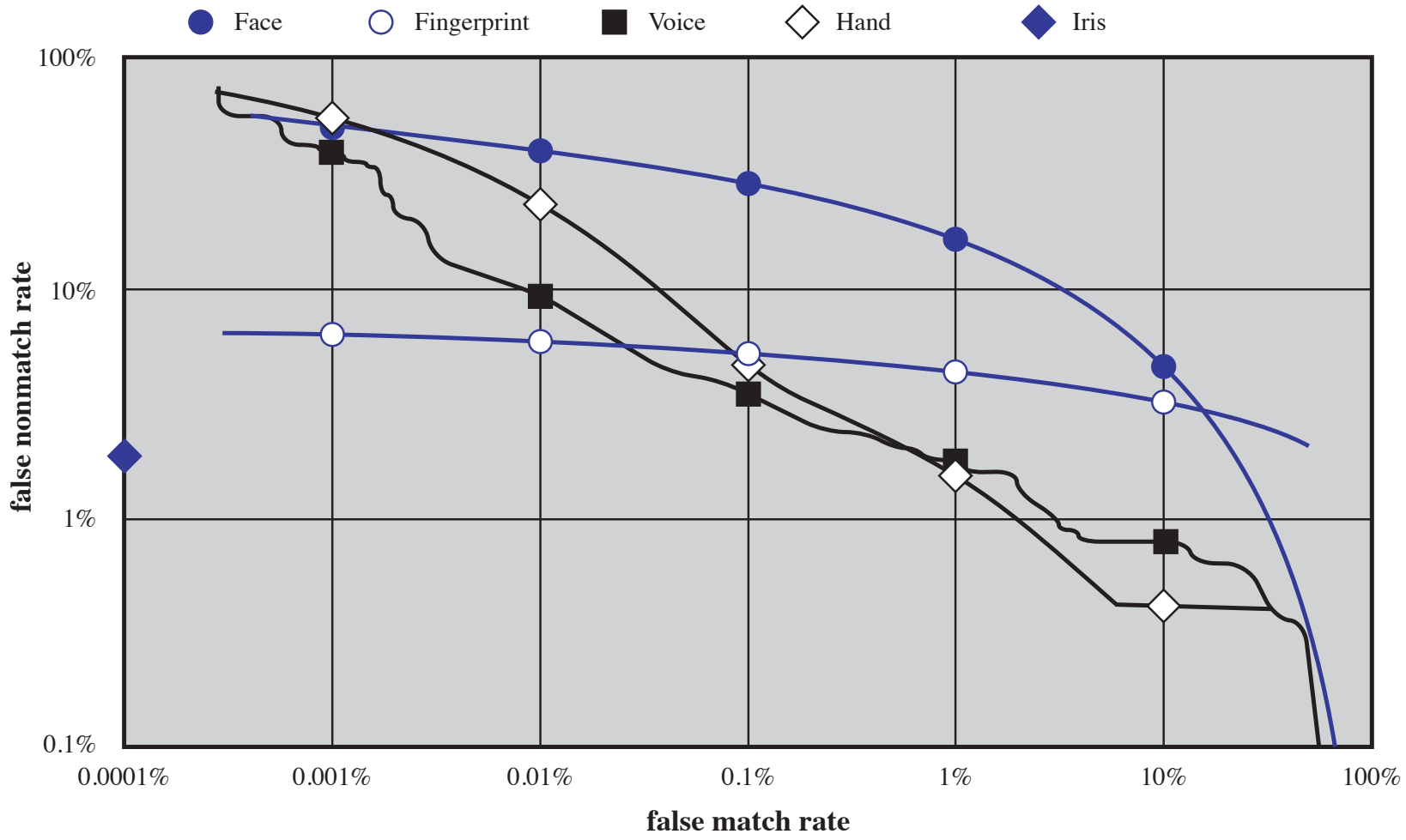


Figure 3.11 Actual Biometric Measurement Operating Characteristic Curves, reported in [MANS01]. To clarify differences among systems, a log-log scale is used.

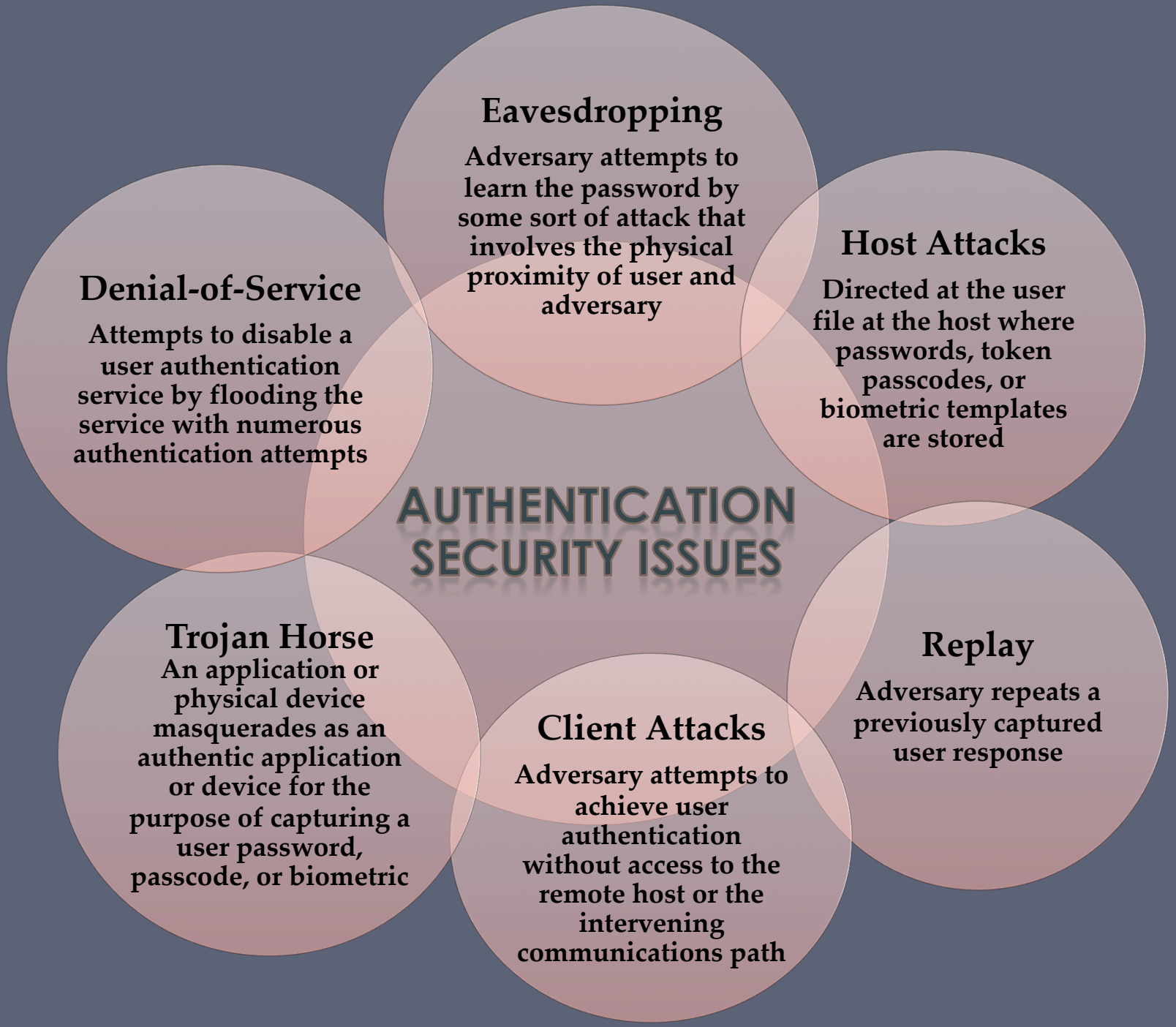
Remote User Authentication

- Authentication over a network, the Internet, or a communications link is more complex
- Additional security threats such as:
  - Eavesdropping, capturing a password, replaying an authentication sequence that has been observed
- Generally rely on some form of a challenge-response protocol to counter threats

Attacks	Authenticators	Examples	Typical defenses
Client attack	Password	Guessing, exhaustive search	Large entropy; limited attempts
	Token	Exhaustive search	Large entropy; limited attempts, theft of object requires presence
	Biometric	False match	Large entropy; limited attempts
Host attack	Password	Plaintext theft, dictionary/exhaustive search	Hashing; large entropy; protection of password database
	Token	Passcode theft	Same as password; 1-time passcode
	Biometric	Template theft	Capture device authentication; challenge response
Eavesdropping, theft, and copying	Password	"Shoulder surfing"	User diligence to keep secret; administrator diligence to quickly revoke compromised passwords; multifactor authentication
	Token	Theft, counterfeiting hardware	Multifactor authentication; tamper resistant/evident token
	Biometric	Copying (spoofing) biometric	Copy detection at capture device and capture device authentication
Replay	Password	Replay stolen password response	Challenge-response protocol
	Token	Replay stolen passcode response	Challenge-response protocol; 1-time passcode
	Biometric	Replay stolen biometric template response	Copy detection at capture device and capture device authentication via challenge-response protocol
Trojan horse	Password, token, biometric	Installation of rogue client or capture device	Authentication of client or capture device within trusted security perimeter

Table 3.4

Some Potential Attacks, Susceptible Authenticators, and Typical Defenses


AUTHENTICATION SECURITY ISSUES
Eavesdropping
Adversary attempts to learn the password by some sort of attack that involves the physical proximity of user and adversary
Host Attacks
Directed at the user file at the host where passwords, token passcodes, or biometric templates are stored
Denial-of-Service
Attempts to disable a user authentication service by flooding the service with numerous authentication attempts
Trojan Horse
An application or physical device masquerades as an authentic application or device for the purpose of capturing a user password, passcode, or biometric
Client Attacks
Adversary attempts to achieve user authentication without access to the remote host or the intervening communications path
Replay
Adversary repeats a previously captured user response

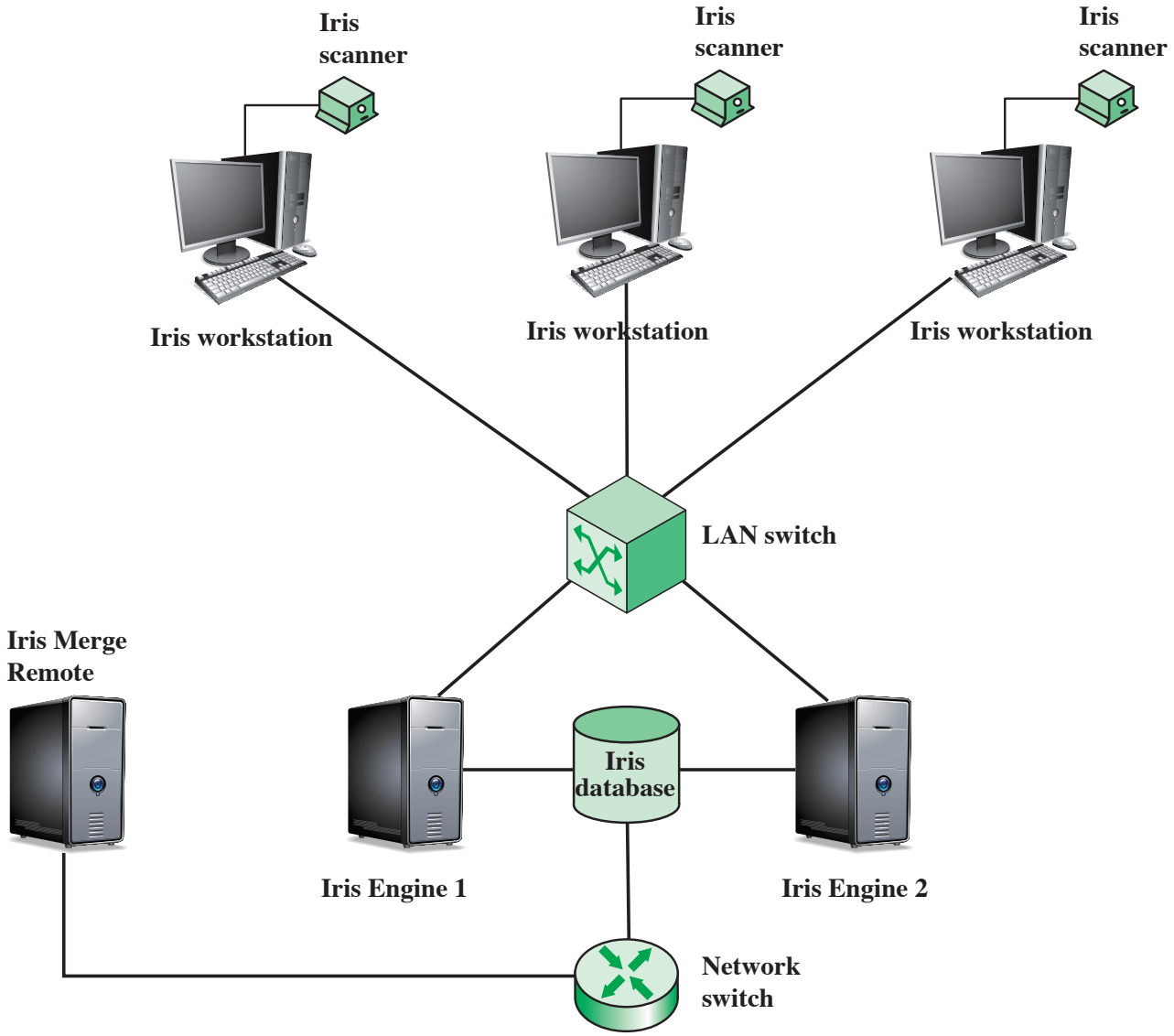


Figure 3.13 General Iris Scan Site Architecture for UAE System

Summary

- Electronic user authentication principles
  - A model for electronic user authentication
  - Means of authentication
  - Risk assessment for user authentication
- Password-based authentication
  - The vulnerability of passwords
  - The use of hashed passwords
  - Password cracking of user-chosen passwords
  - Password file access control
  - Password selection strategies
- Token-based authentication
  - Memory cards
  - Smart cards
  - Electronic identity cards

- Biometric authentication
  - Physical characteristics used in biometric applications
  - Operation of a biometric authentication system
  - Biometric accuracy
- Remote user authentication
  - Password protocol
  - Token protocol
  - Static biometric protocol
  - Dynamic biometric protocol
- Security issues for user authentication