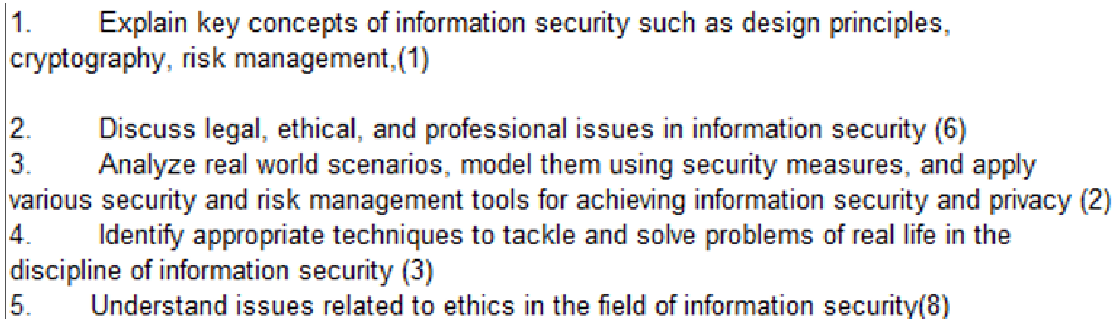
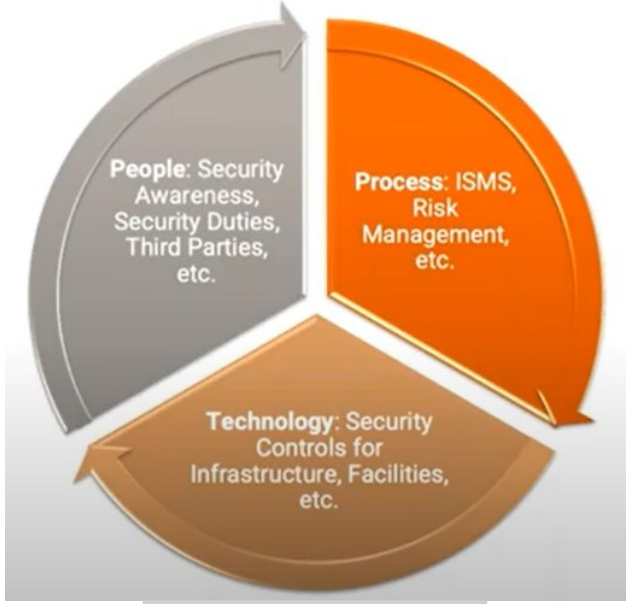
CS 3002 Information Security

Fall 2023



1. Explain key concepts of information security such as design principles, cryptography, risk management,(1)

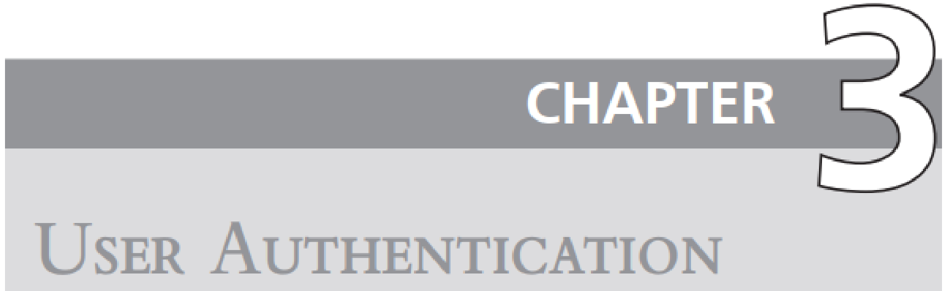
2. Discuss legal, ethical, and professional issues in information security (6)
3. Analyze real world scenarios, model them using security measures, and apply various security and risk management tools for achieving information security and privacy (2)
4. Identify appropriate techniques to tackle and solve problems of real life in the discipline of information security (3)
5. Understand issues related to ethics in the field of information security(8)


People: Security Awareness, Security Duties, Third Parties, etc.
Process: ISMS, Risk Management, etc.
Technology: Security Controls for Infrastructure, Facilities, etc.

ISO/IEC 27001: 2013

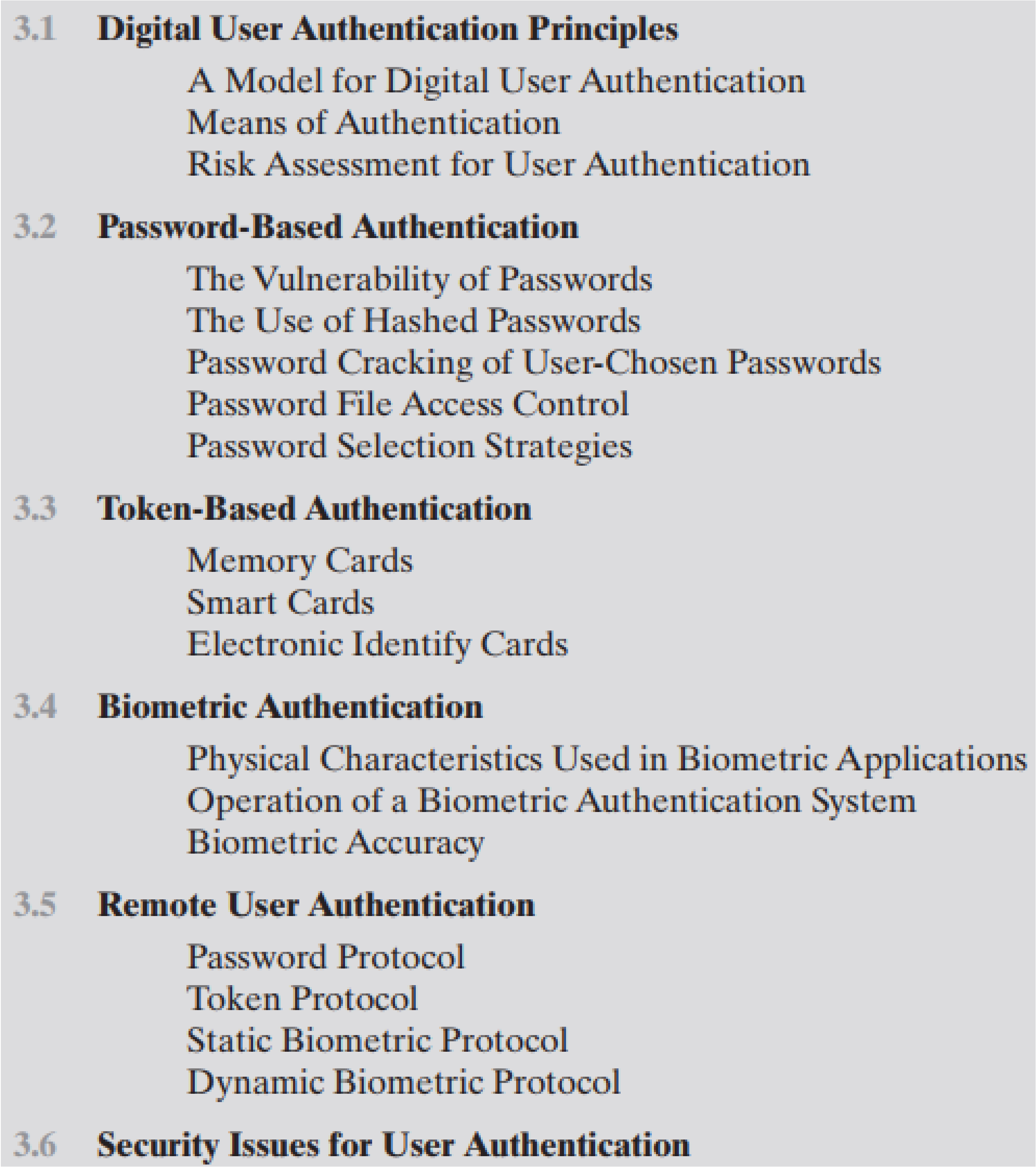
Week # 5

Dr. Nadeem Kafi Khan



CHAPTER 3

User Authentication



Multifactor authentication refers to the use of more than one of the authentication means in the preceding list. Implementations that use two factors are considered to be stronger than those that use only one factor; systems that incorporate three factors are stronger than systems that only incorporate two of the factors, and so on.

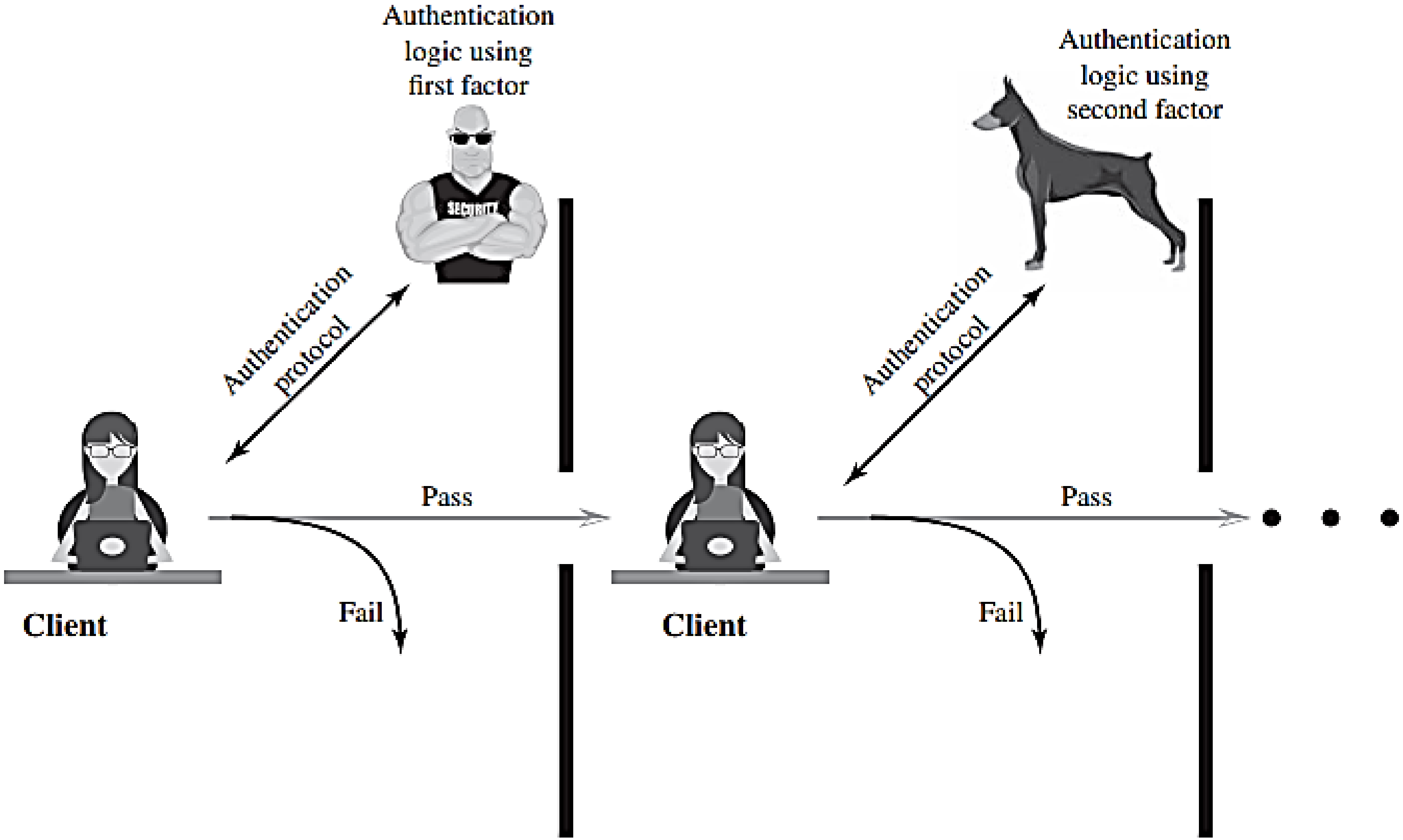
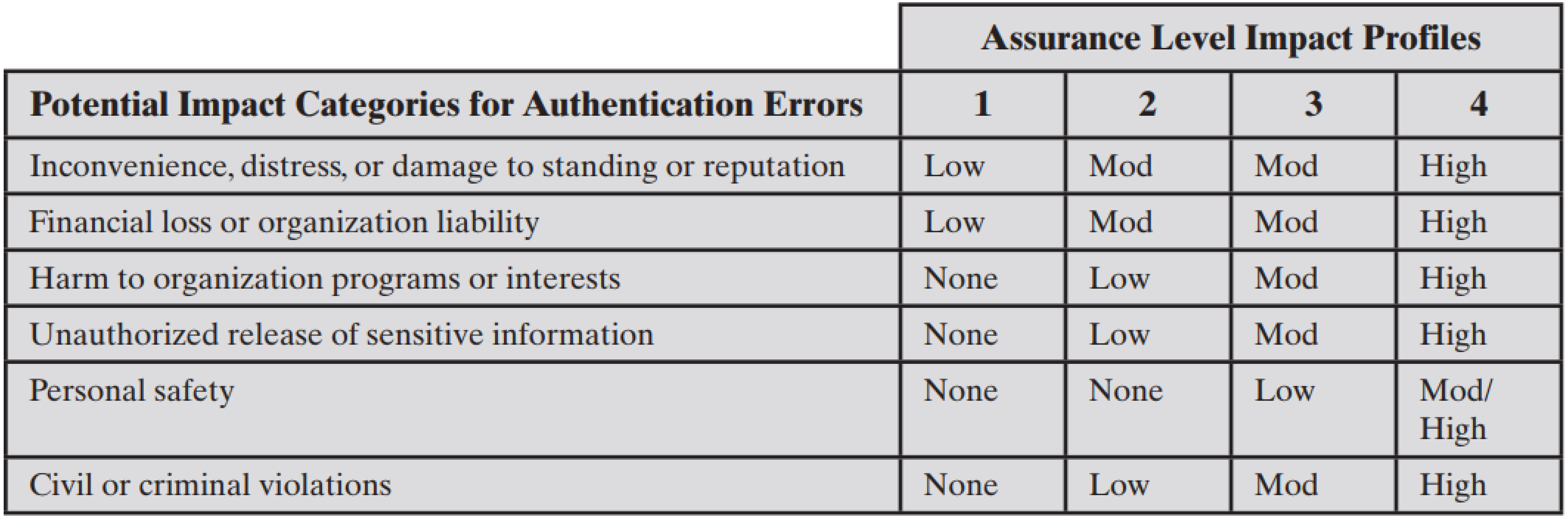


Figure 3.2 **Multifactor Authentication**

Risk Assessment for User Authentication

assurance level, potential impact, and areas of risk.

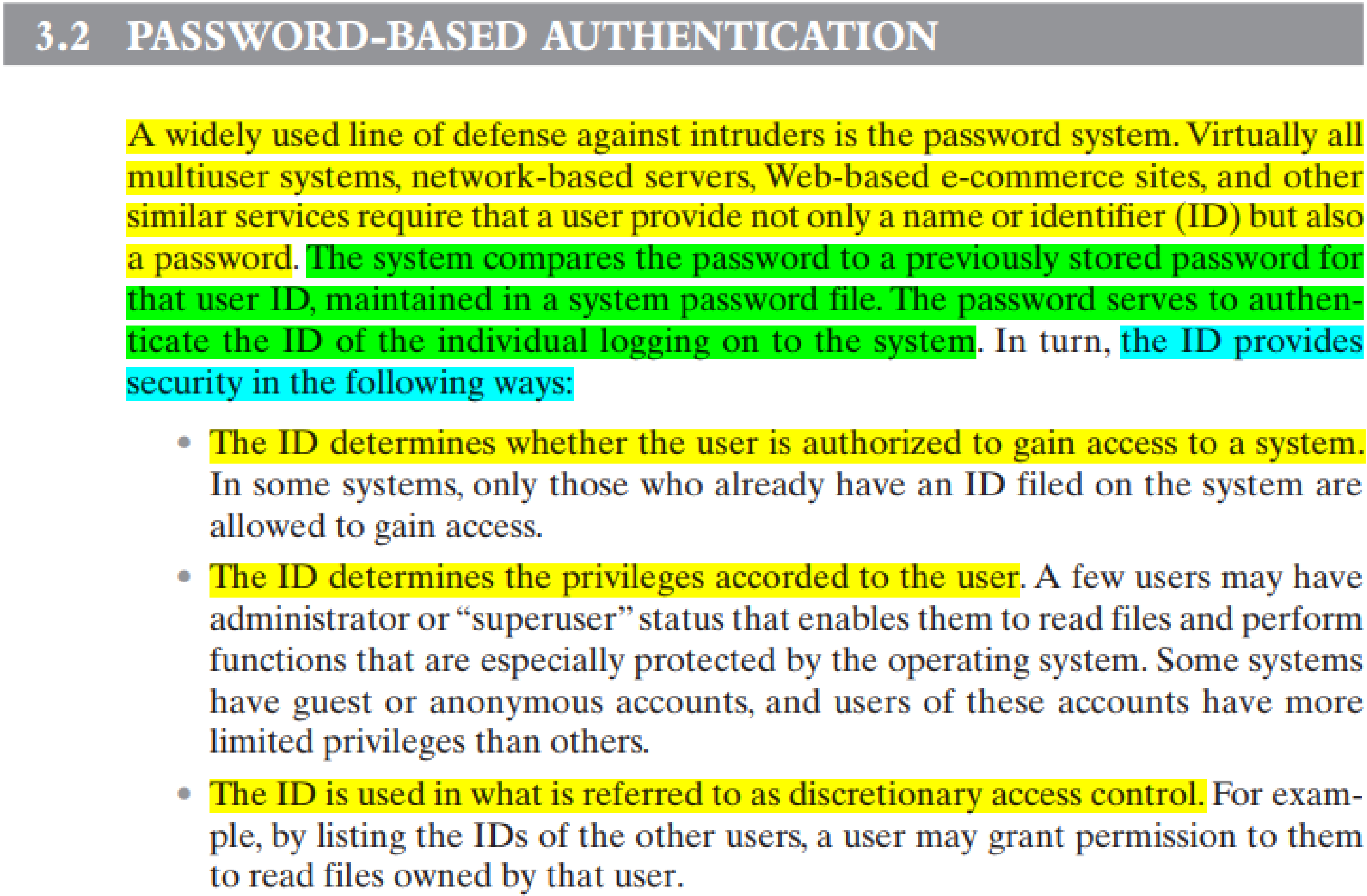
Table 3.2 Maximum Potential Impacts for Each Assurance Level



	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

This table suggests a technique for doing risk assessment. For a given information system or service asset of an organization, the organization needs to determine the level of impact if an authentication failure occurs, using the categories of impact, or risk areas, that are of concern.

Read Assurance levels, Potential risk, Risk from the book

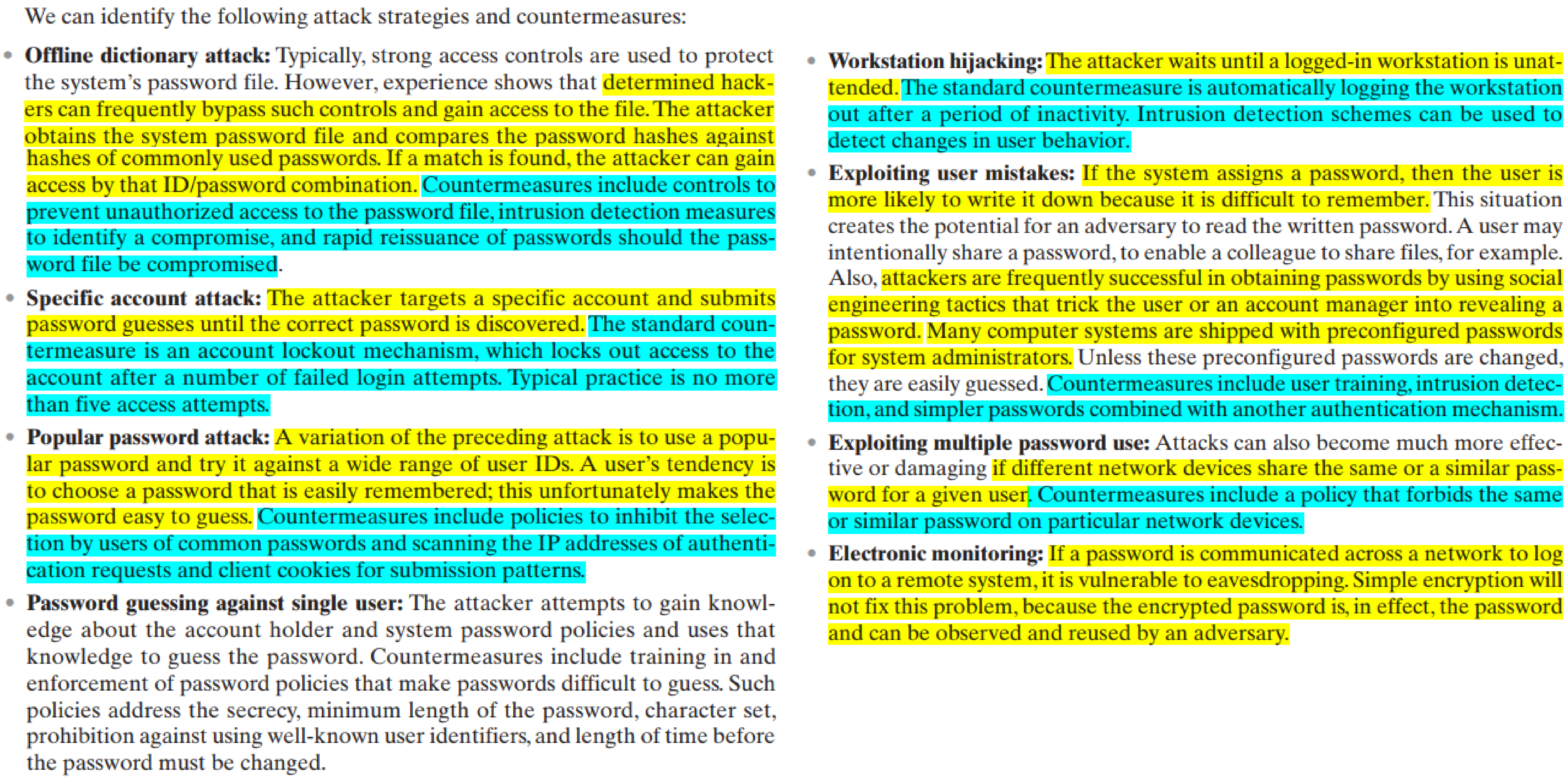


3.2 PASSWORD-BASED AUTHENTICATION

A widely used line of defense against intruders is the password system. Virtually all multiuser systems, network-based servers, Web-based e-commerce sites, and other similar services require that a user provide not only a name or identifier (ID) but also a password. The system compares the password to a previously stored password for that user ID, maintained in a system password file. The password serves to authenticate the ID of the individual logging on to the system. In turn, the ID provides security in the following ways:

- The ID determines whether the user is authorized to gain access to a system. In some systems, only those who already have an ID filed on the system are allowed to gain access.
- The ID determines the privileges accorded to the user. A few users may have administrator or "superuser" status that enables them to read files and perform functions that are especially protected by the operating system. Some systems have guest or anonymous accounts, and users of these accounts have more limited privileges than others.
- The ID is used in what is referred to as discretionary access control. For example, by listing the IDs of the other users, a user may grant permission to them to read files owned by that user.

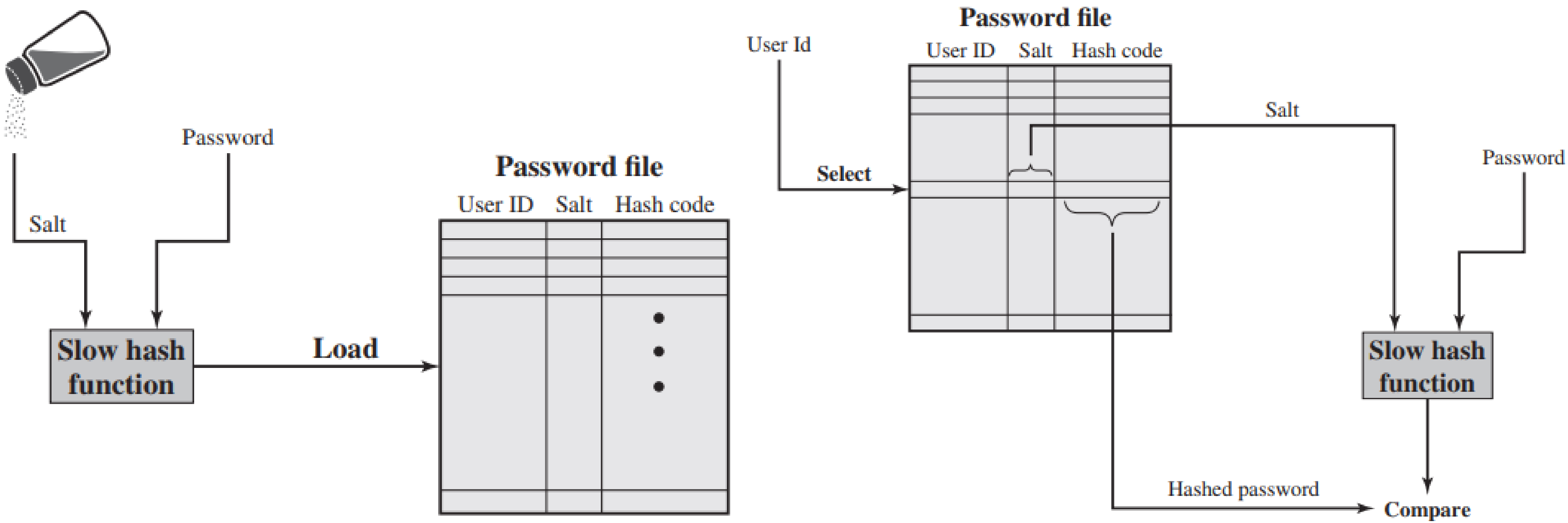
The Vulnerability of Passwords



We can identify the following attack strategies and countermeasures:

- **Offline dictionary attack:** Typically, strong access controls are used to protect the system's password file. However, experience shows that determined hackers can frequently bypass such controls and gain access to the file. The attacker obtains the system password file and compares the password hashes against hashes of commonly used passwords. If a match is found, the attacker can gain access by that ID/password combination. Countermeasures include controls to prevent unauthorized access to the password file, intrusion detection measures to identify a compromise, and rapid reissuance of passwords should the password file be compromised.
- **Specific account attack:** The attacker targets a specific account and submits password guesses until the correct password is discovered. The standard countermeasure is an account lockout mechanism, which locks out access to the account after a number of failed login attempts. Typical practice is no more than five access attempts.
- **Popular password attack:** A variation of the preceding attack is to use a popular password and try it against a wide range of user IDs. A user's tendency is to choose a password that is easily remembered; this unfortunately makes the password easy to guess. Countermeasures include policies to inhibit the selection by users of common passwords and scanning the IP addresses of authentication requests and client cookies for submission patterns.
- **Password guessing against single user:** The attacker attempts to gain knowledge about the account holder and system password policies and uses that knowledge to guess the password. Countermeasures include training in and enforcement of password policies that make passwords difficult to guess. Such policies address the secrecy, minimum length of the password, character set, prohibition against using well-known user identifiers, and length of time before the password must be changed.
- **Workstation hijacking:** The attacker waits until a logged-in workstation is unattended. The standard countermeasure is automatically logging the workstation out after a period of inactivity. Intrusion detection schemes can be used to detect changes in user behavior.
- **Exploiting user mistakes:** If the system assigns a password, then the user is more likely to write it down because it is difficult to remember. This situation creates the potential for an adversary to read the written password. A user may intentionally share a password, to enable a colleague to share files, for example. Also, attackers are frequently successful in obtaining passwords by using social engineering tactics that trick the user or an account manager into revealing a password. Many computer systems are shipped with preconfigured passwords for system administrators. Unless these preconfigured passwords are changed, they are easily guessed. Countermeasures include user training, intrusion detection, and simpler passwords combined with another authentication mechanism.
- **Exploiting multiple password use:** Attacks can also become much more effective or damaging if different network devices share the same or a similar password for a given user. Countermeasures include a policy that forbids the same or similar password on particular network devices.
- **Electronic monitoring:** If a password is communicated across a network to log on to a remote system, it is vulnerable to eavesdropping. Simple encryption will not fix this problem, because the encrypted password is, in effect, the password and can be observed and reused by an adversary.

Figure 3.3 UNIX Password Scheme

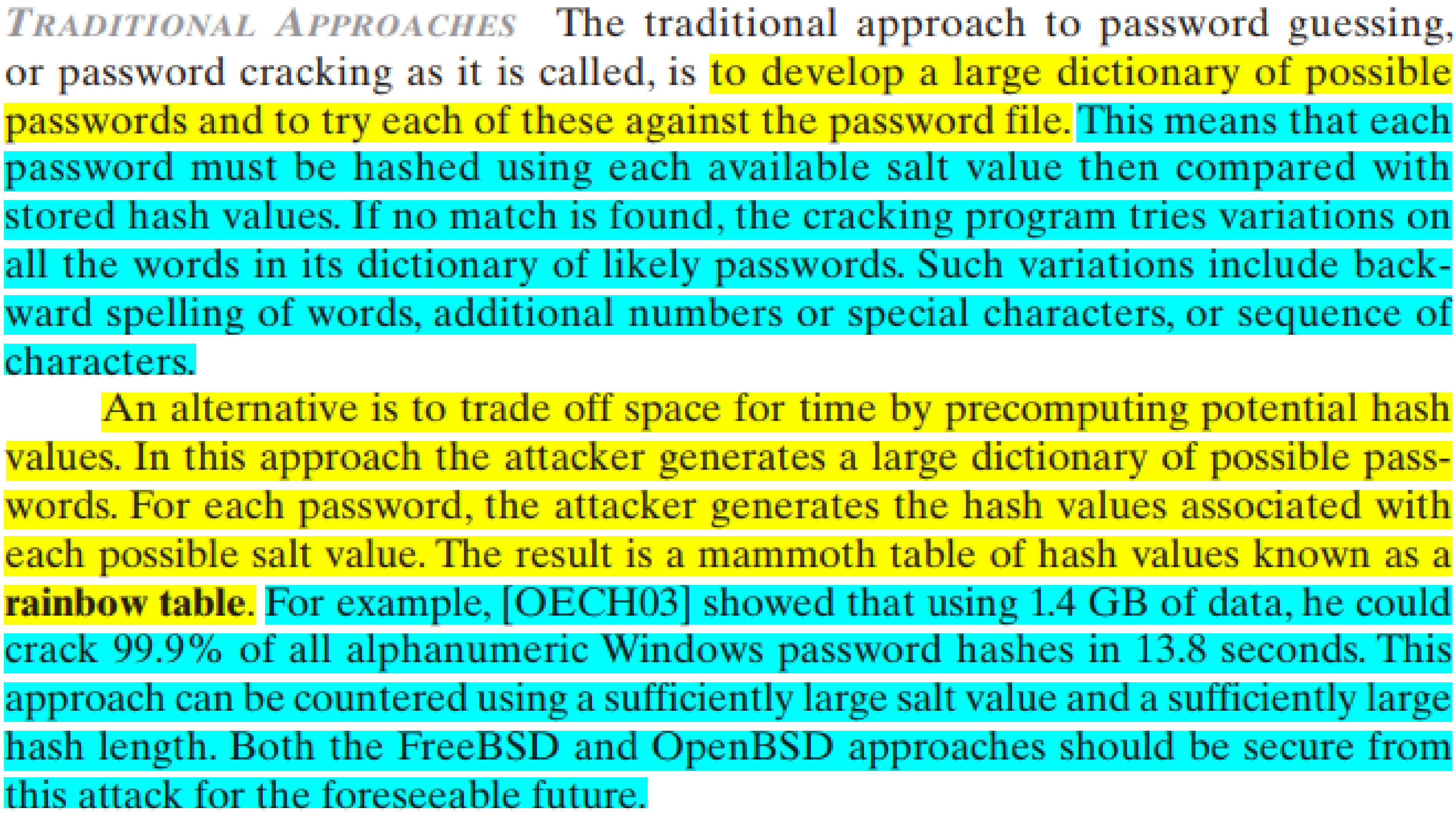


(a) Loading a new password

(b) Verifying a password

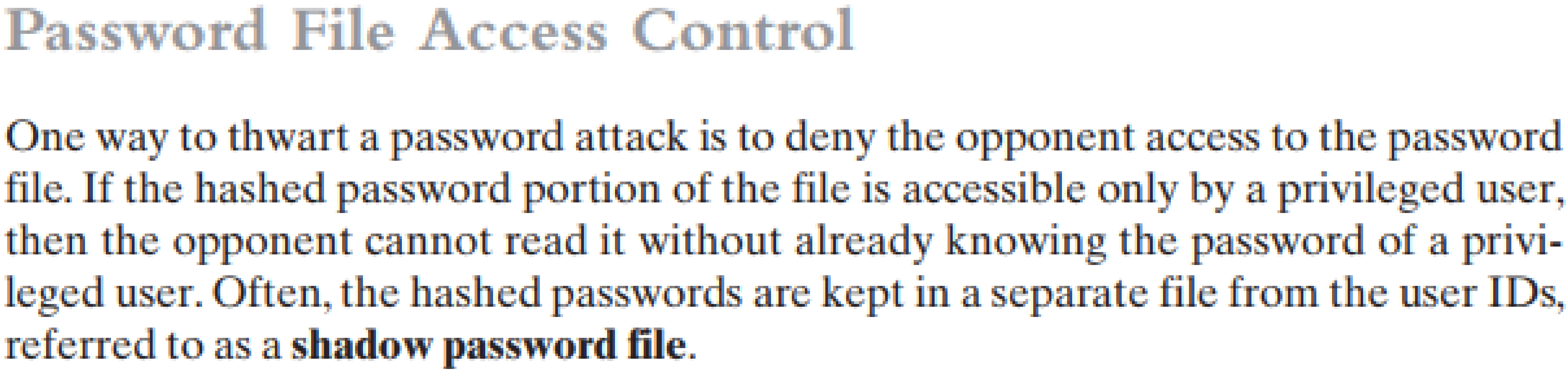
The password and salt serve as inputs to a hashing algorithm to produce a fixed-length hash code. The hash algorithm is designed to be slow to execute in order to thwart attacks. The hashed password is then stored, together with a plaintext copy of the salt, in the password file for the corresponding user ID. The hashed password method has been shown to be secure against a variety of cryptanalytic attacks

Password Cracking of User-Chosen Passwords



TRADITIONAL APPROACHES The traditional approach to password guessing, or password cracking as it is called, is to develop a large dictionary of possible passwords and to try each of these against the password file. This means that each password must be hashed using each available salt value then compared with stored hash values. If no match is found, the cracking program tries variations on all the words in its dictionary of likely passwords. Such variations include backward spelling of words, additional numbers or special characters, or sequence of characters.

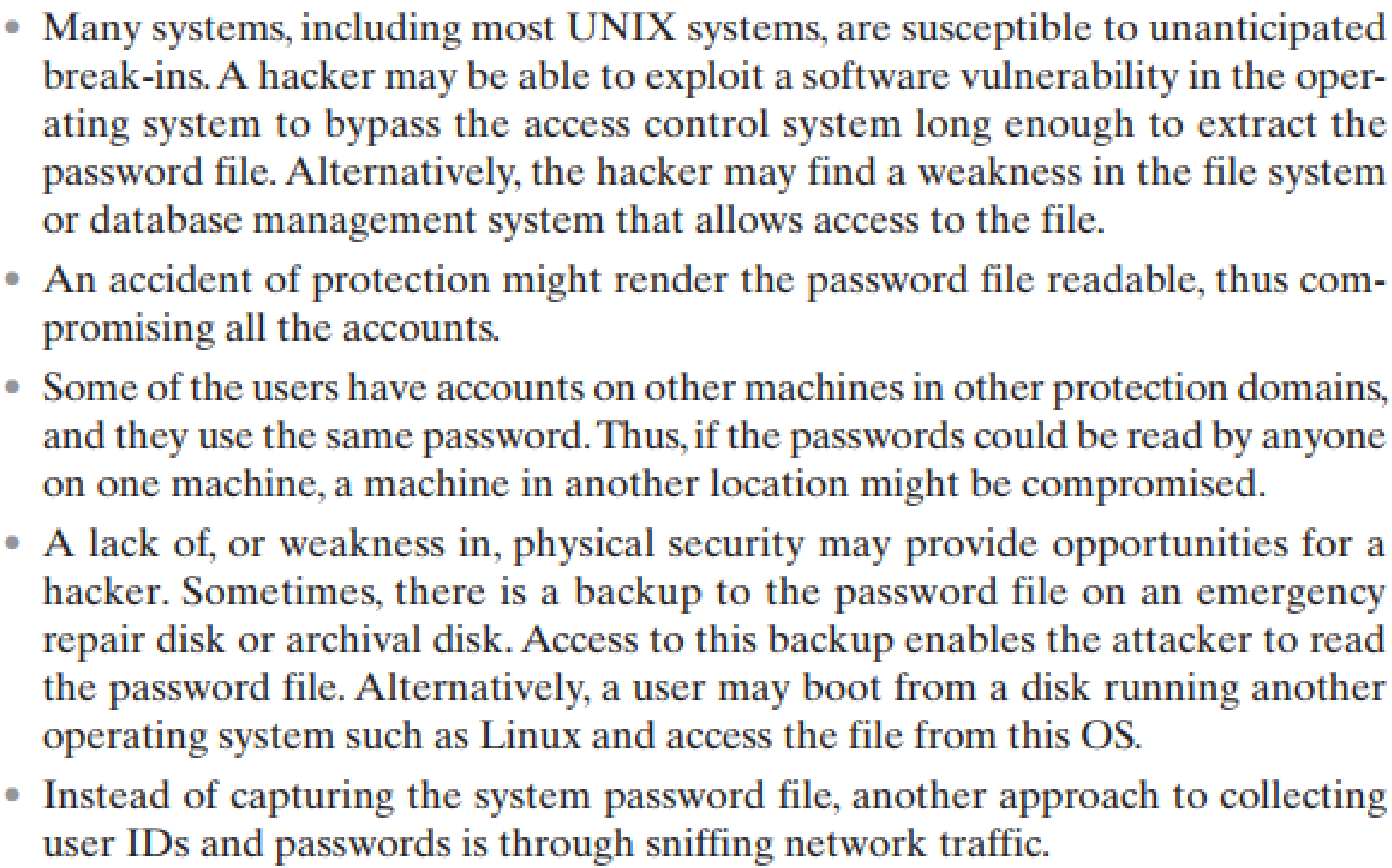
An alternative is to trade off space for time by precomputing potential hash values. In this approach the attacker generates a large dictionary of possible passwords. For each password, the attacker generates the hash values associated with each possible salt value. The result is a mammoth table of hash values known as a **rainbow table**. For example, [OECH03] showed that using 1.4 GB of data, he could crack 99.9% of all alphanumeric Windows password hashes in 13.8 seconds. This approach can be countered using a sufficiently large salt value and a sufficiently large hash length. Both the FreeBSD and OpenBSD approaches should be secure from this attack for the foreseeable future.



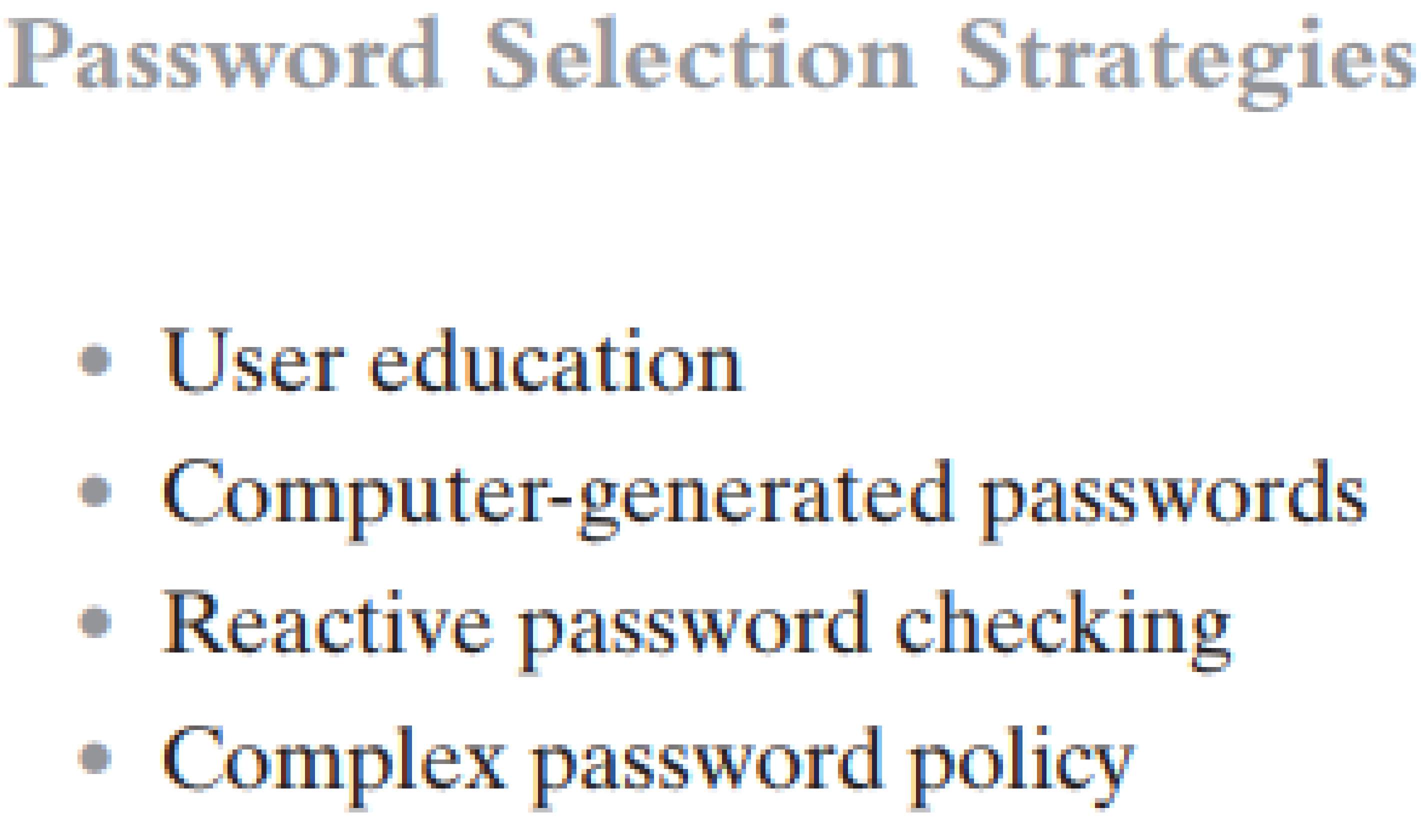
Password File Access Control

One way to thwart a password attack is to deny the opponent access to the password file. If the hashed password portion of the file is accessible only by a privileged user, then the opponent cannot read it without already knowing the password of a privileged user. Often, the hashed passwords are kept in a separate file from the user IDs, referred to as a **shadow password file**.

Password File Access Control vulnerabilities

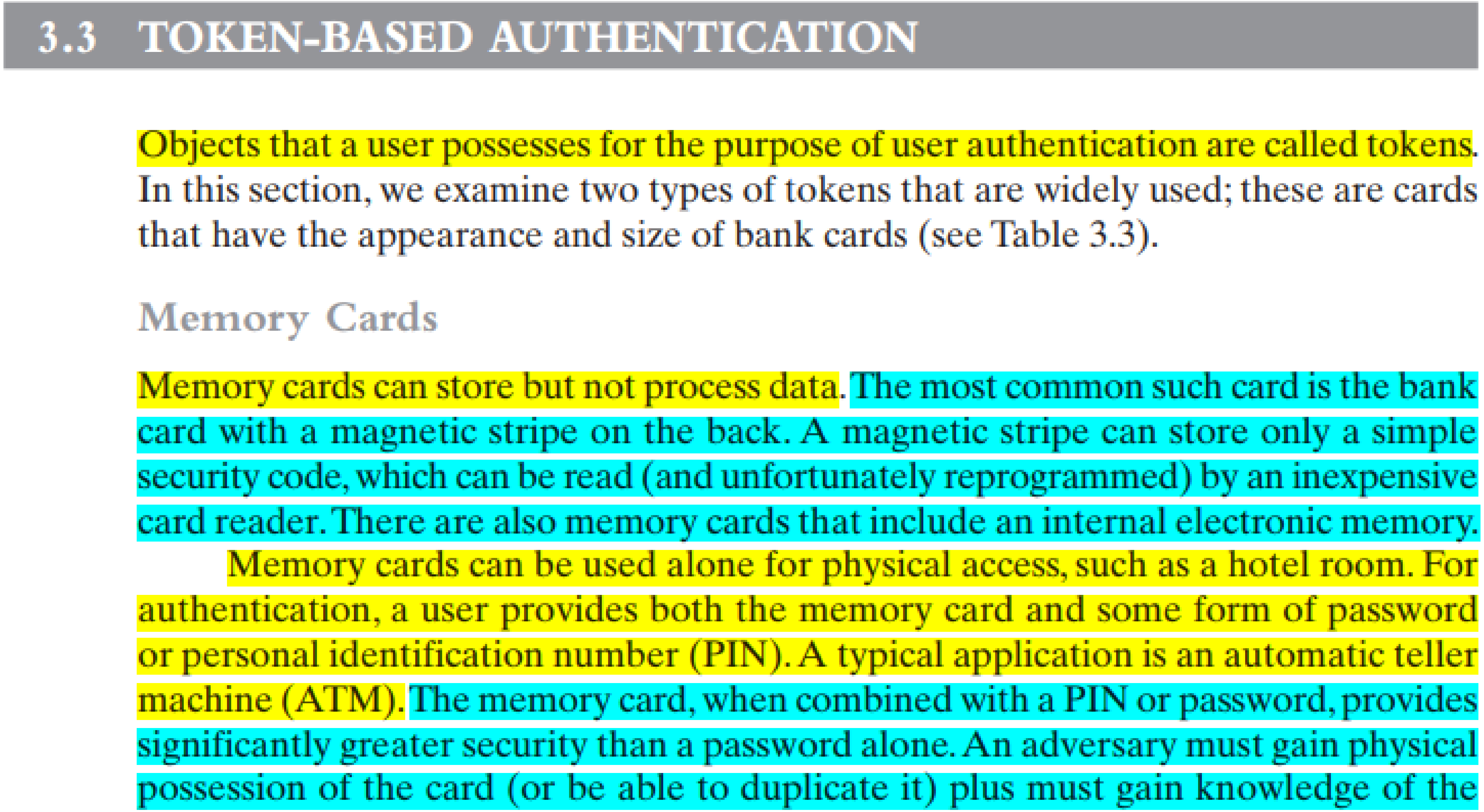


- Many systems, including most UNIX systems, are susceptible to unanticipated break-ins. A hacker may be able to exploit a software vulnerability in the operating system to bypass the access control system long enough to extract the password file. Alternatively, the hacker may find a weakness in the file system or database management system that allows access to the file.
- An accident of protection might render the password file readable, thus compromising all the accounts.
- Some of the users have accounts on other machines in other protection domains, and they use the same password. Thus, if the passwords could be read by anyone on one machine, a machine in another location might be compromised.
- A lack of, or weakness in, physical security may provide opportunities for a hacker. Sometimes, there is a backup to the password file on an emergency repair disk or archival disk. Access to this backup enables the attacker to read the password file. Alternatively, a user may boot from a disk running another operating system such as Linux and access the file from this OS.
- Instead of capturing the system password file, another approach to collecting user IDs and passwords is through sniffing network traffic.



Password Selection Strategies

- User education
- Computer-generated passwords
- Reactive password checking
- Complex password policy



3.3 TOKEN-BASED AUTHENTICATION

Objects that a user possesses for the purpose of user authentication are called tokens. In this section, we examine two types of tokens that are widely used; these are cards that have the appearance and size of bank cards (see Table 3.3).

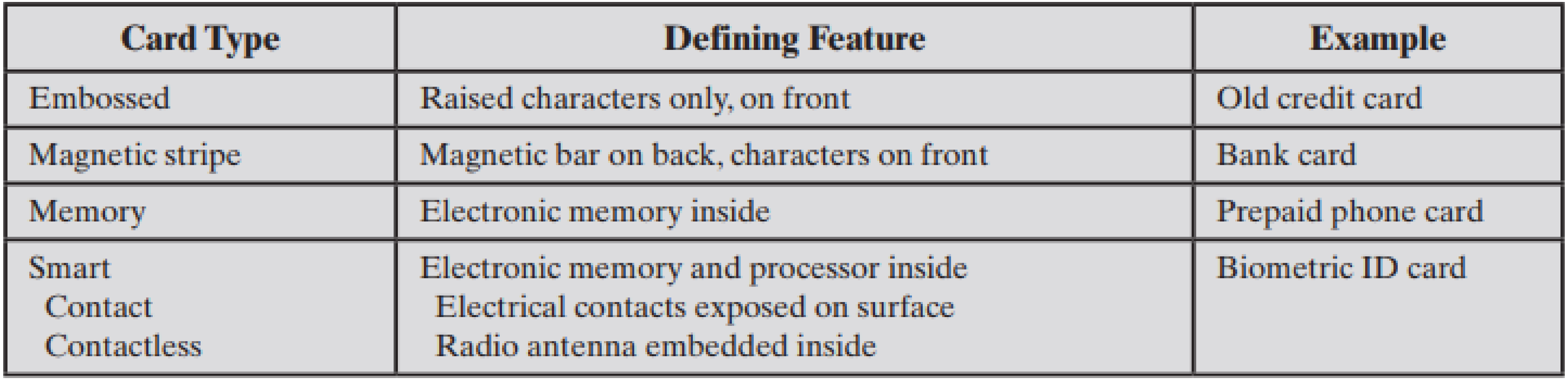
Memory Cards

Memory cards can store but not process data. The most common such card is the bank card with a magnetic stripe on the back. A magnetic stripe can store only a simple security code, which can be read (and unfortunately reprogrammed) by an inexpensive card reader. There are also memory cards that include an internal electronic memory.

Memory cards can be used alone for physical access, such as a hotel room. For authentication, a user provides both the memory card and some form of password or personal identification number (PIN). A typical application is an automatic teller machine (ATM). The memory card, when combined with a PIN or password, provides significantly greater security than a password alone. An adversary must gain physical possession of the card (or be able to duplicate it) plus must gain knowledge of the

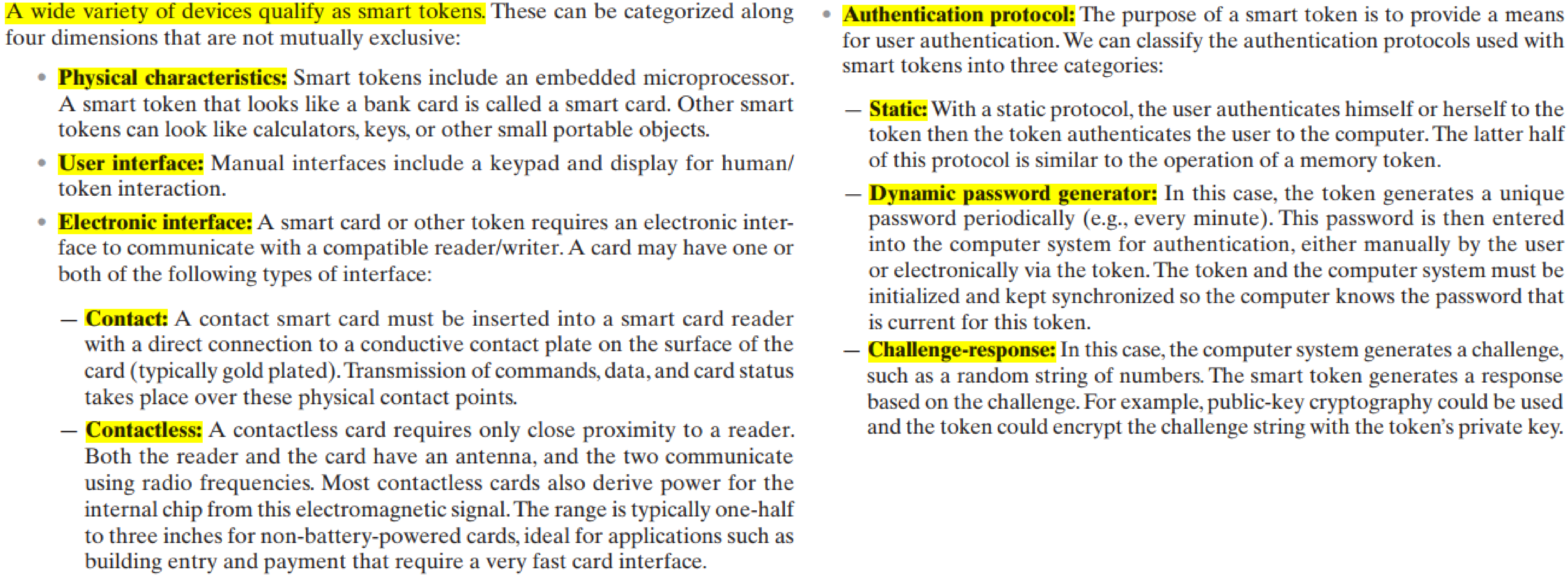
Read about potential drawbacks from the book

Table 3.3 Types of Cards Used as Tokens



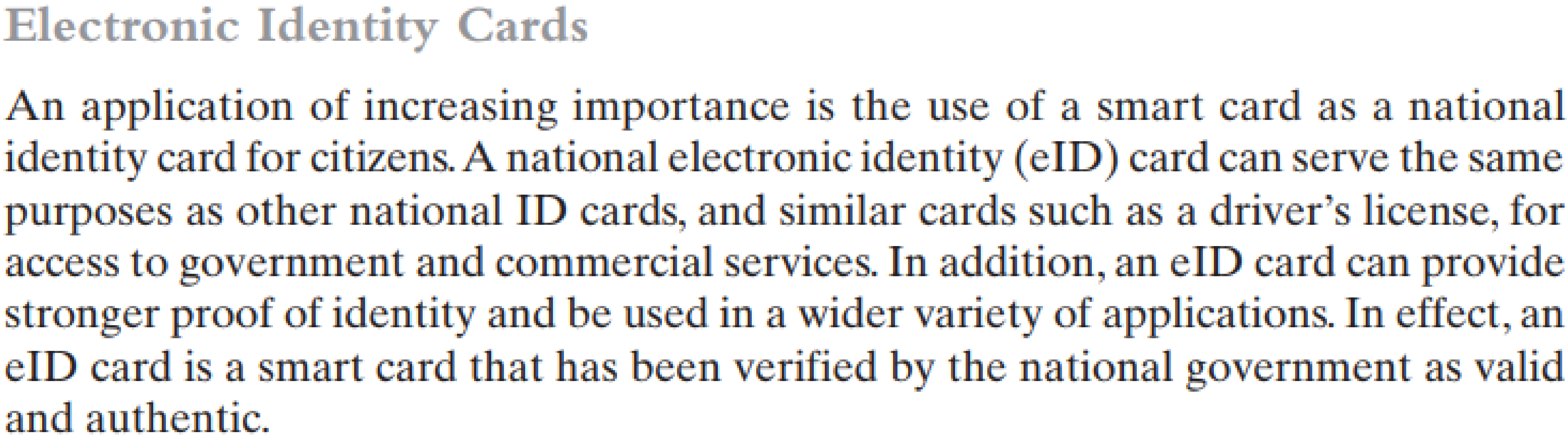
Card Type	Defining Feature	Example
Embossed	Raised characters only, on front	Old credit card
Magnetic stripe	Magnetic bar on back, characters on front	Bank card
Memory	Electronic memory inside	Prepaid phone card
Smart Contact Contactless	Electronic memory and processor inside Electrical contacts exposed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

Smart Cards



A wide variety of devices qualify as smart tokens. These can be categorized along four dimensions that are not mutually exclusive:

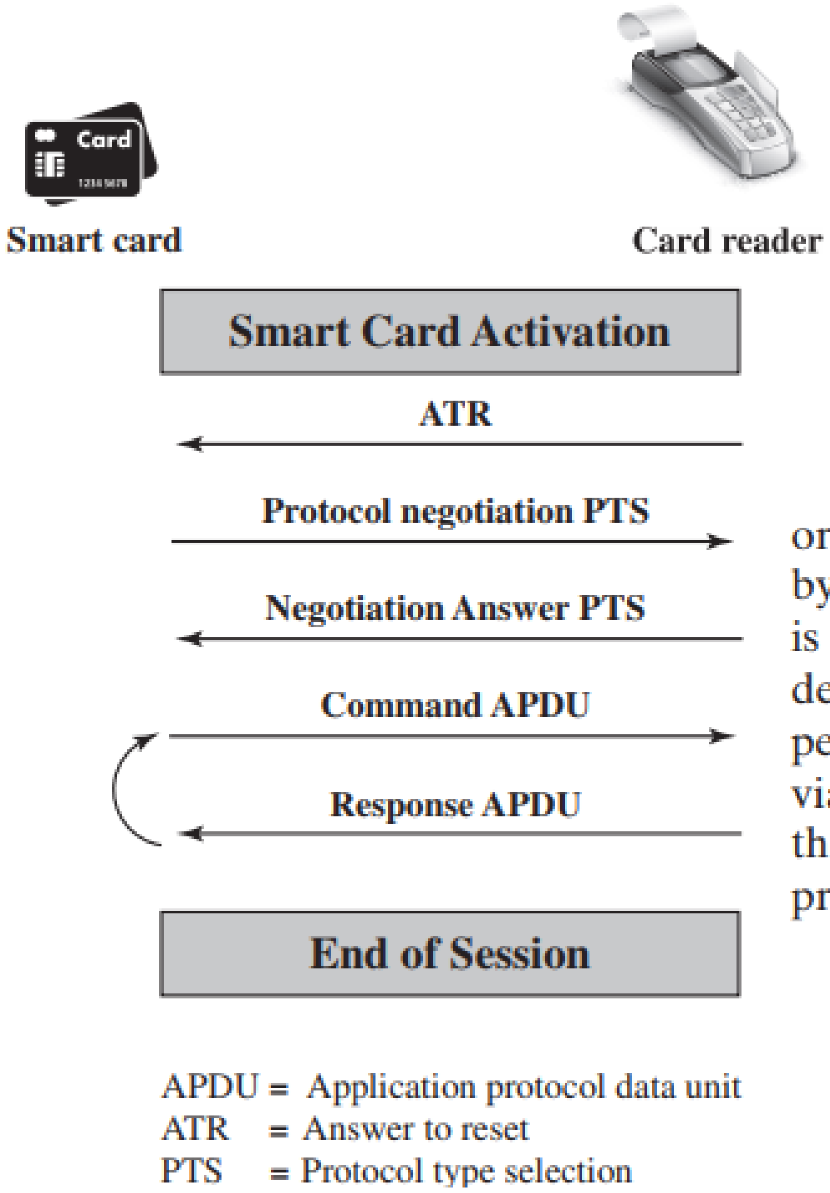
- **Physical characteristics:** Smart tokens include an embedded microprocessor. A smart token that looks like a bank card is called a smart card. Other smart tokens can look like calculators, keys, or other small portable objects.
- **User interface:** Manual interfaces include a keypad and display for human/token interaction.
- **Electronic interface:** A smart card or other token requires an electronic interface to communicate with a compatible reader/writer. A card may have one or both of the following types of interface:
  - **Contact:** A contact smart card must be inserted into a smart card reader with a direct connection to a conductive contact plate on the surface of the card (typically gold plated). Transmission of commands, data, and card status takes place over these physical contact points.
  - **Contactless:** A contactless card requires only close proximity to a reader. Both the reader and the card have an antenna, and the two communicate using radio frequencies. Most contactless cards also derive power for the internal chip from this electromagnetic signal. The range is typically one-half to three inches for non-battery-powered cards, ideal for applications such as building entry and payment that require a very fast card interface.
- **Authentication protocol:** The purpose of a smart token is to provide a means for user authentication. We can classify the authentication protocols used with smart tokens into three categories:
  - **Static:** With a static protocol, the user authenticates himself or herself to the token then the token authenticates the user to the computer. The latter half of this protocol is similar to the operation of a memory token.
  - **Dynamic password generator:** In this case, the token generates a unique password periodically (e.g., every minute). This password is then entered into the computer system for authentication, either manually by the user or electronically via the token. The token and the computer system must be initialized and kept synchronized so the computer knows the password that is current for this token.
  - **Challenge-response:** In this case, the computer system generates a challenge, such as a random string of numbers. The smart token generates a response based on the challenge. For example, public-key cryptography could be used and the token could encrypt the challenge string with the token's private key.



Electronic Identity Cards

An application of increasing importance is the use of a smart card as a national identity card for citizens. A national electronic identity (eID) card can serve the same purposes as other national ID cards, and similar cards such as a driver's license, for access to government and commercial services. In addition, an eID card can provide stronger proof of identity and be used in a wider variety of applications. In effect, an eID card is a smart card that has been verified by the national government as valid and authentic.

Smart Cards



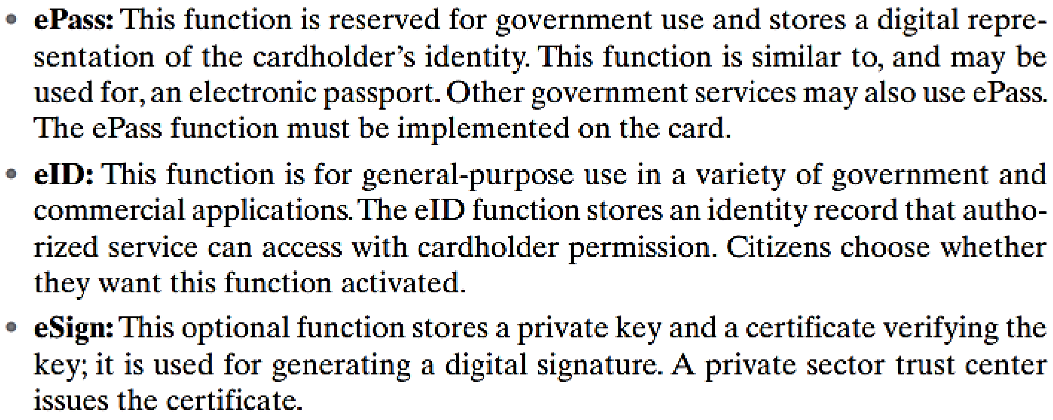
APDU = Application protocol data unit
ATR = Answer to reset
PTS = Protocol type selection

Figure 3.6 Smart Card/Reader Exchange

Figure 3.6 illustrates the typical interaction between a smart card and a reader or computer system. Each time the card is inserted into a reader, a reset is initiated by the reader to initialize parameters such as clock value. After the reset function is performed, the card responds with answer to reset (ATR) message. This message defines the parameters and protocols that the card can use and the functions it can perform. The terminal may be able to change the protocol used and other parameters via a protocol type selection (PTS) command. The card's PTS response confirms the protocols and parameters to be used. The terminal and card can now execute the protocol to perform the desired application.

Electronic Identity Cards

The card has the following three separate electronic functions, each with its own protected dataset (see Table 3.4):



- **ePass:** This function is reserved for government use and stores a digital representation of the cardholder's identity. This function is similar to, and may be used for, an electronic passport. Other government services may also use ePass. The ePass function must be implemented on the card.
- **eID:** This function is for general-purpose use in a variety of government and commercial applications. The eID function stores an identity record that authorized service can access with cardholder permission. Citizens choose whether they want this function activated.
- **eSign:** This optional function stores a private key and a certificate verifying the key; it is used for generating a digital signature. A private sector trust center issues the certificate.

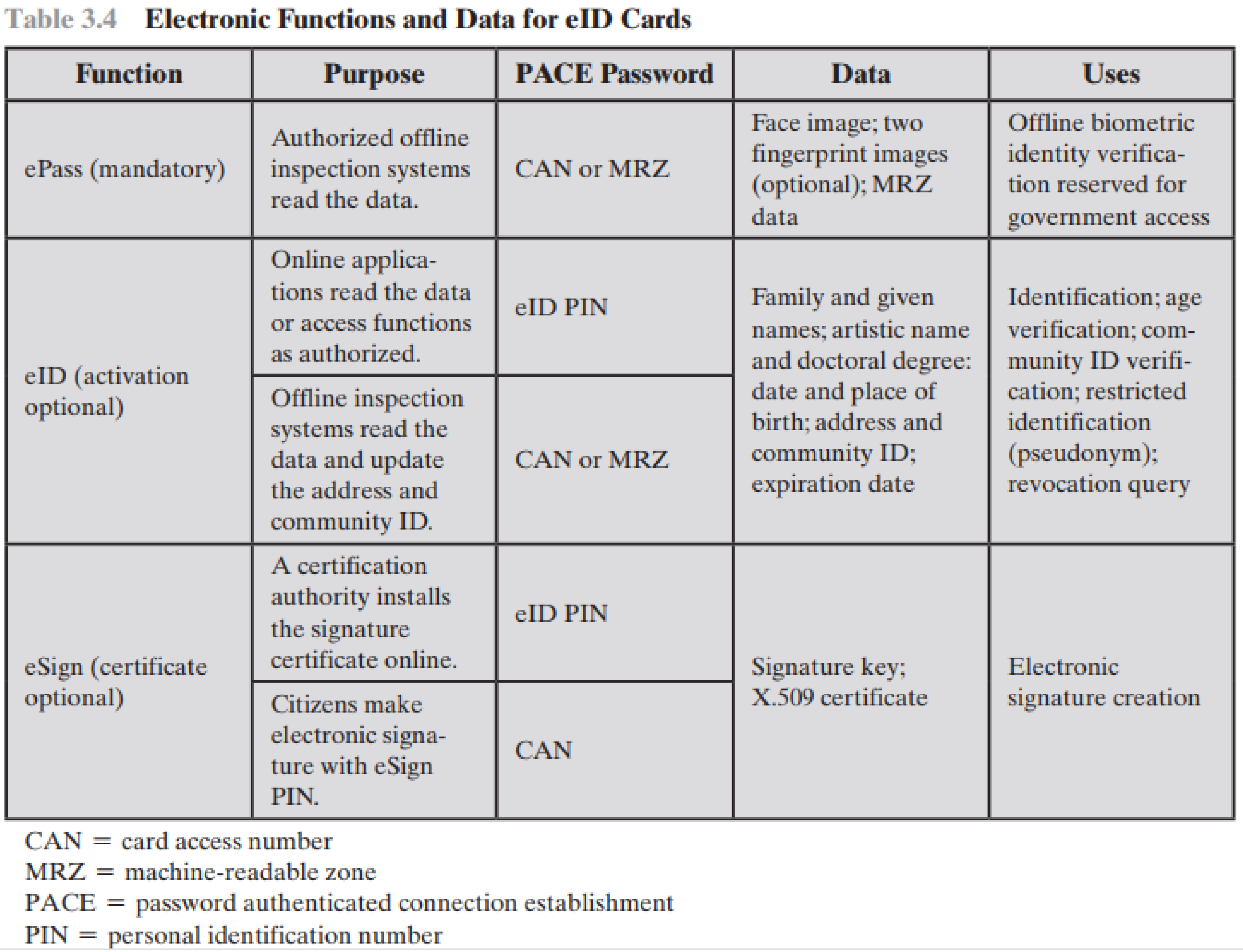
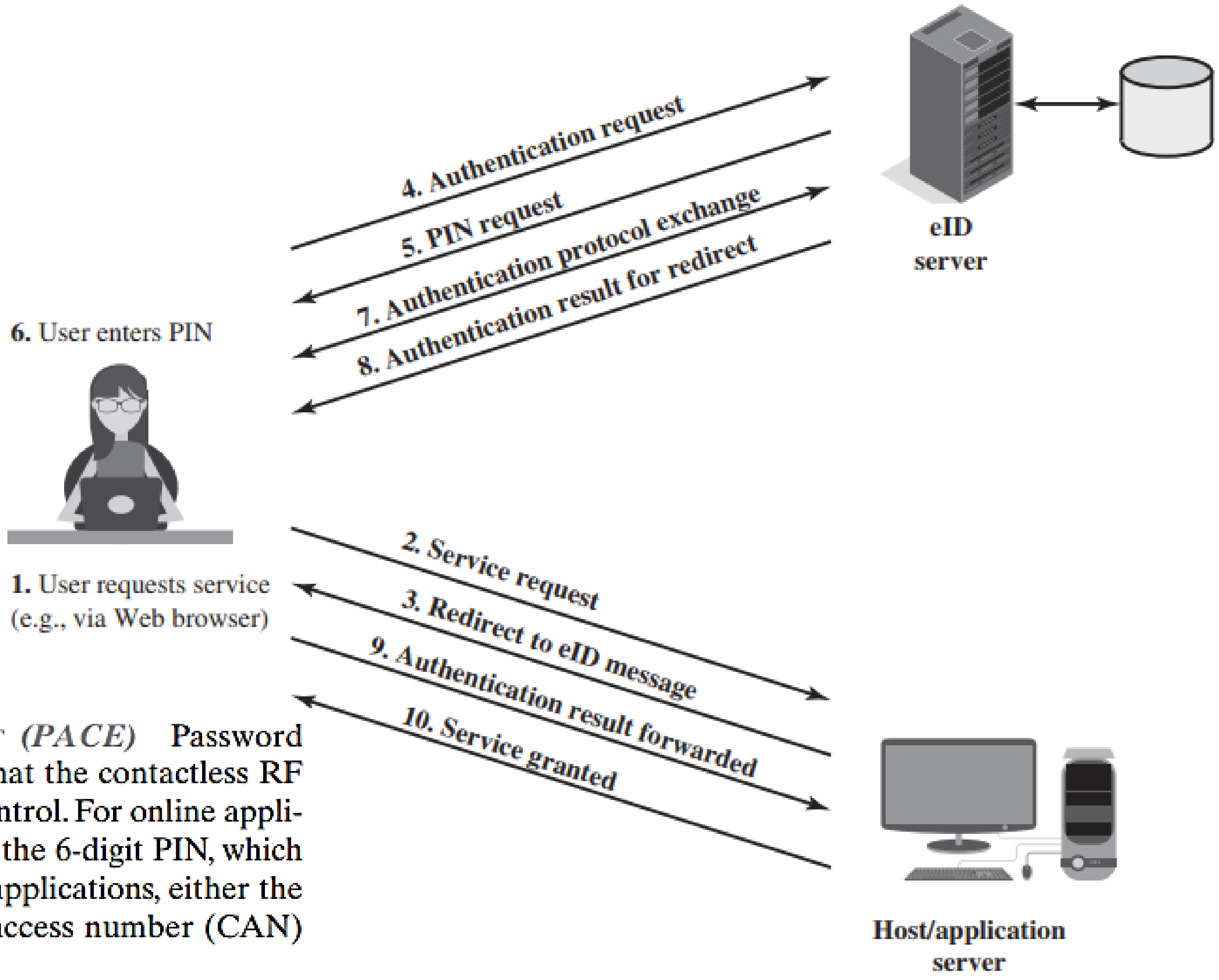


Table 3.4 **Electronic Functions and Data for eID Cards**

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		

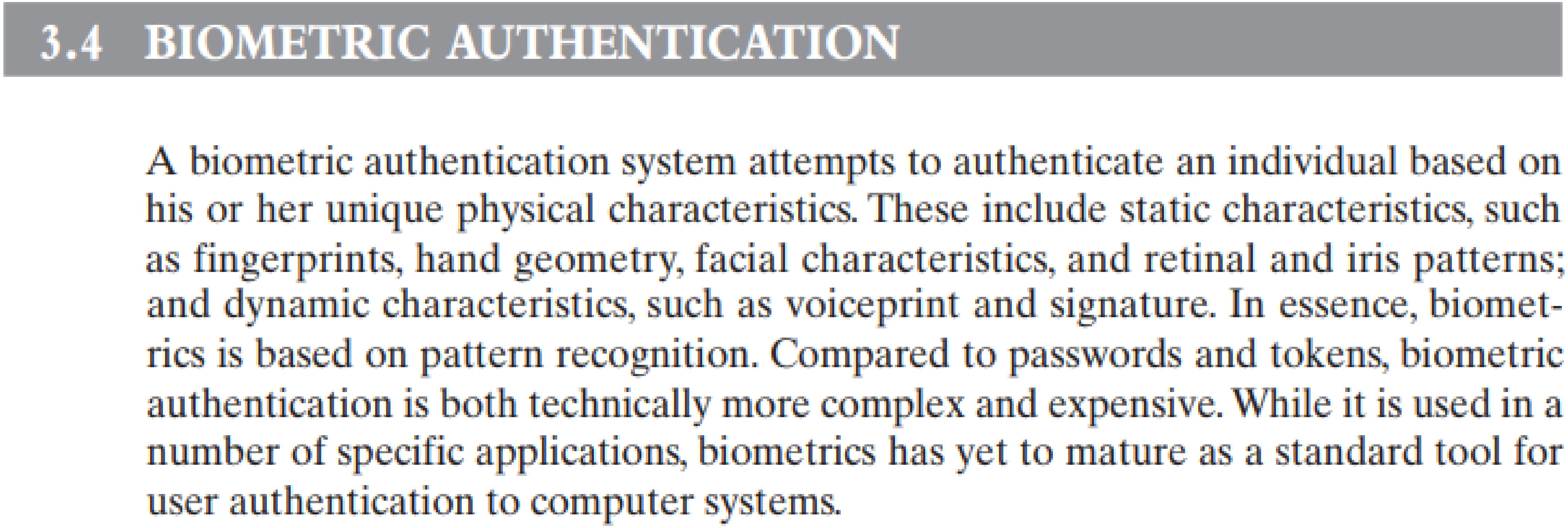
CAN = card access number
MRZ = machine-readable zone
PACE = password authenticated connection establishment
PIN = personal identification number

Electronic Identity Cards


eID server
4. Authentication request
5. PIN request
7. Authentication protocol exchange
8. Authentication result for redirect
6. User enters PIN
1. User requests service (e.g., via Web browser)
2. Service request
3. Redirect to eID message
9. Authentication result forwarded
10. Service granted
Host/application server

Password Authenticated Connection Establishment (PACE) Password Authenticated Connection Establishment (PACE) ensures that the contactless RF chip in the eID card cannot be read without explicit access control. For online applications, access to the card is established by the user entering the 6-digit PIN, which should only be known to the holder of the card. For offline applications, either the MRZ printed on the back of the card or the six-digit card access number (CAN) printed on the front is used.

Figure 3.7 **User Authentication with eID**



3.4 BIOMETRIC AUTHENTICATION

A biometric authentication system attempts to authenticate an individual based on his or her unique physical characteristics. These include static characteristics, such as fingerprints, hand geometry, facial characteristics, and retinal and iris patterns; and dynamic characteristics, such as voiceprint and signature. In essence, biometrics is based on pattern recognition. Compared to passwords and tokens, biometric authentication is both technically more complex and expensive. While it is used in a number of specific applications, biometrics has yet to mature as a standard tool for user authentication to computer systems.

Physical Characteristics Used in Biometric Applications

- **Facial characteristics:** Facial characteristics are the most common means of human-to-human identification; thus it is natural to consider them for identification by computer. The most common approach is to define characteristics based on relative location and shape of key facial features, such as eyes, eyebrows, nose, lips, and chin shape. An alternative approach is to use an infrared camera to produce a face thermogram that correlates with the underlying vascular system in the human face.
- **Fingerprints:** Fingerprints have been used as a means of identification for centuries, and the process has been systematized and automated particularly for law enforcement purposes. A fingerprint is the pattern of ridges and furrows on the surface of the fingertip. Fingerprints are believed to be unique across the entire human population. In practice, automated fingerprint recognition and matching system extract a number of features from the fingerprint for storage as a numerical surrogate for the full fingerprint pattern.
- **Hand geometry:** Hand geometry systems identify features of the hand, including shape, and lengths and widths of fingers.
- **Retinal pattern:** The pattern formed by veins beneath the retinal surface is unique and therefore suitable for identification. A retinal biometric system obtains a digital image of the retinal pattern by projecting a low-intensity beam of visual or infrared light into the eye.
- **Iris:** Another unique physical characteristic is the detailed structure of the iris.
- **Signature:** Each individual has a unique style of handwriting and this is reflected especially in the signature, which is typically a frequently written sequence. However, multiple signature samples from a single individual will not be identical. This complicates the task of developing a computer representation of the signature that can be matched to future samples.
- **Voice:** Whereas the signature style of an individual reflects not only the unique physical attributes of the writer but also the writing habit that has developed, voice patterns are more closely tied to the physical and anatomical characteristics of the speaker. Nevertheless, there is still a variation from sample to sample over time from the same speaker, complicating the biometric recognition task.

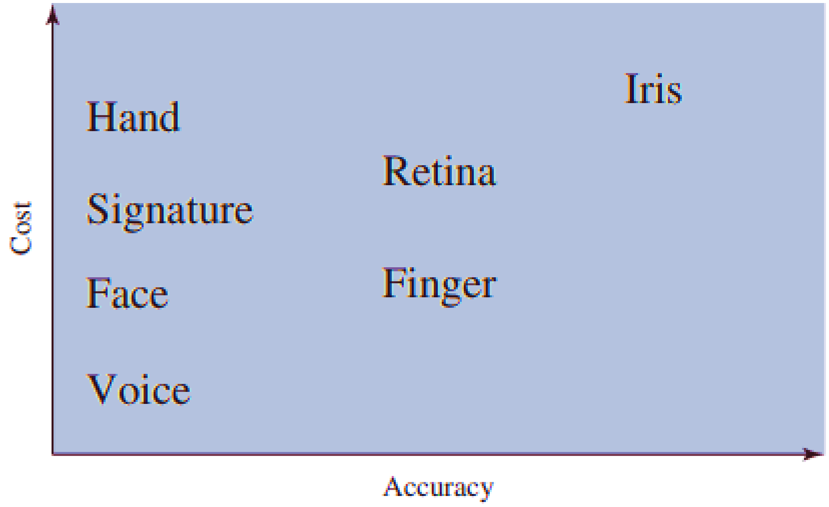
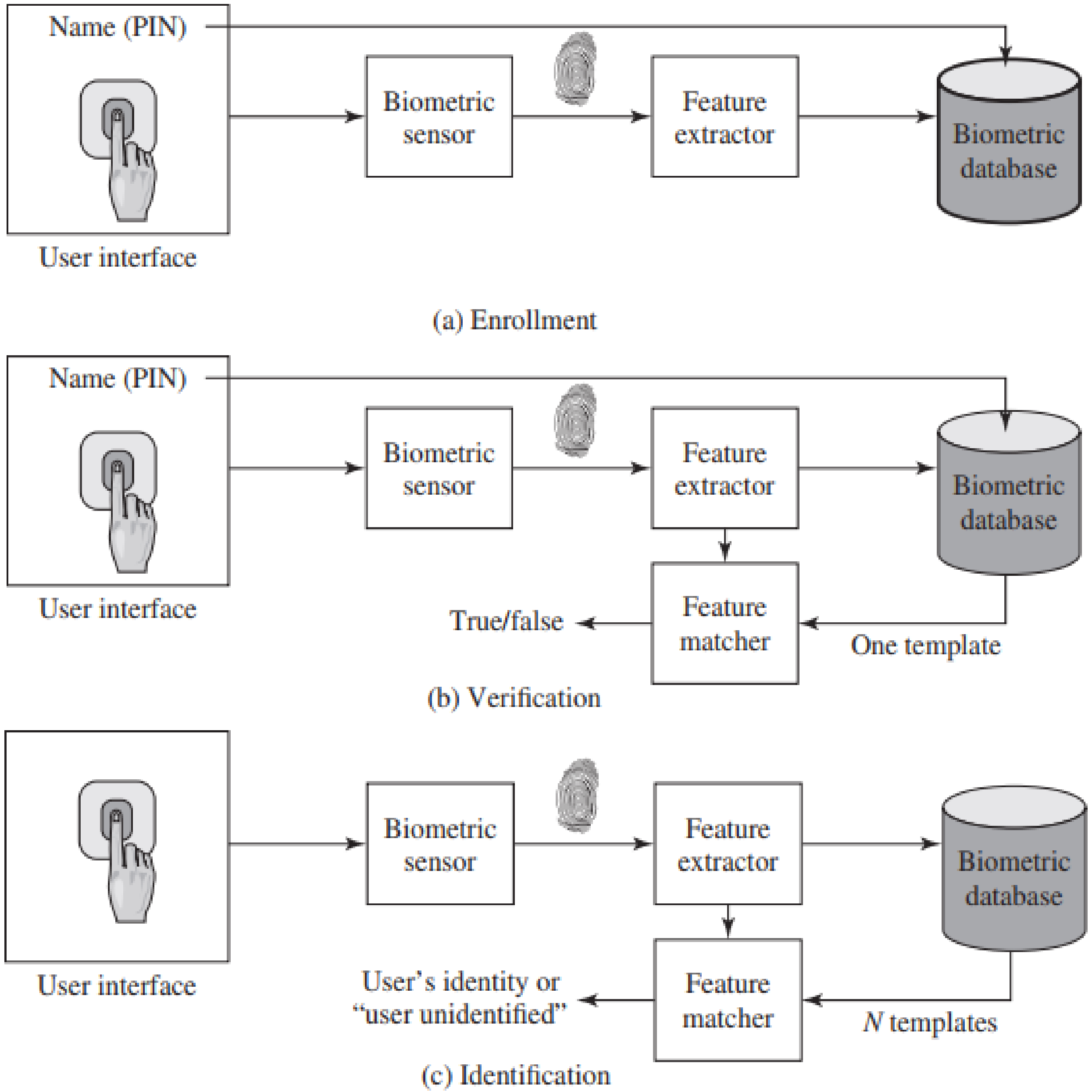
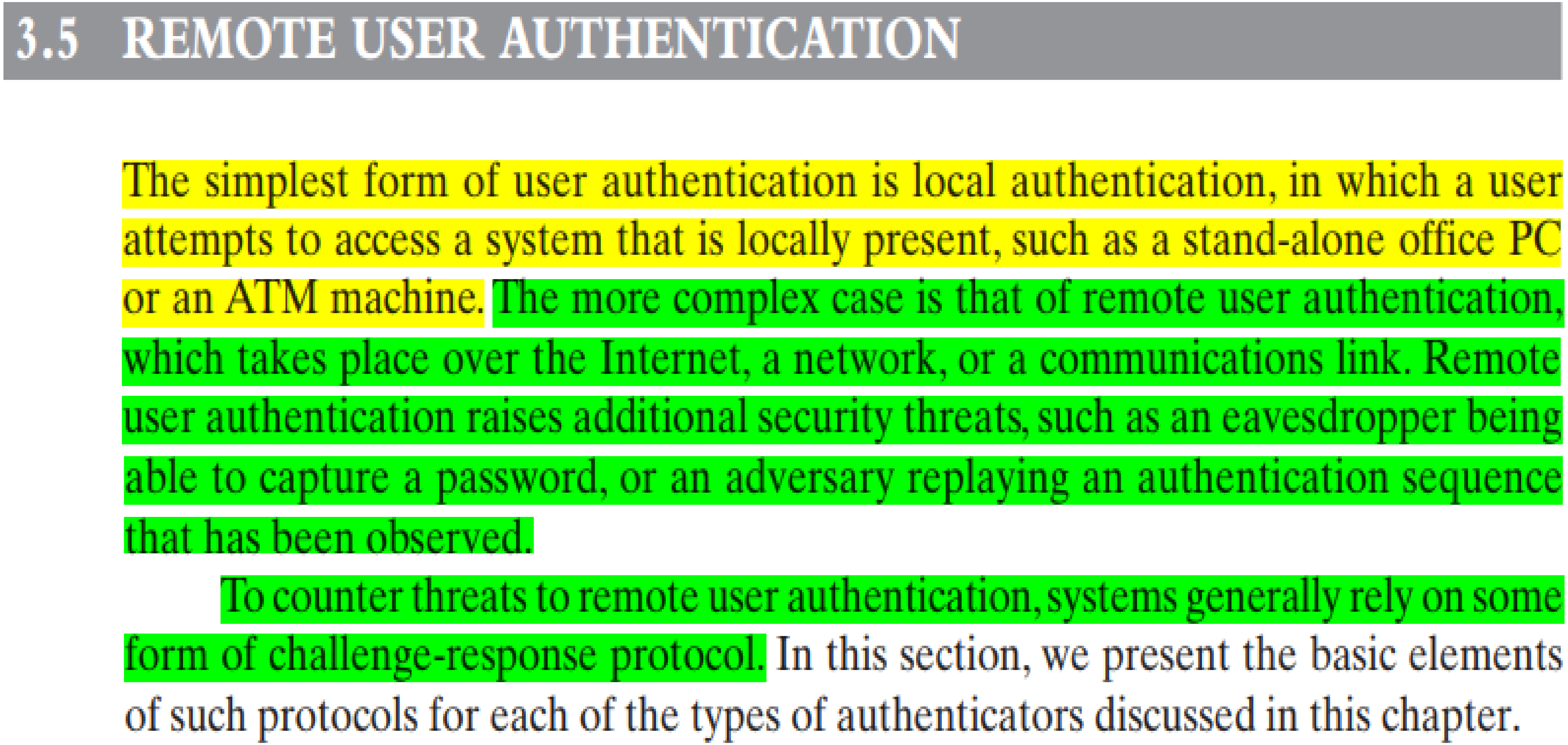


Figure 3.8 **Cost Versus Accuracy of Various Biometric Characteristics in User Authentication Schemes**

Figure 3.9 **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.

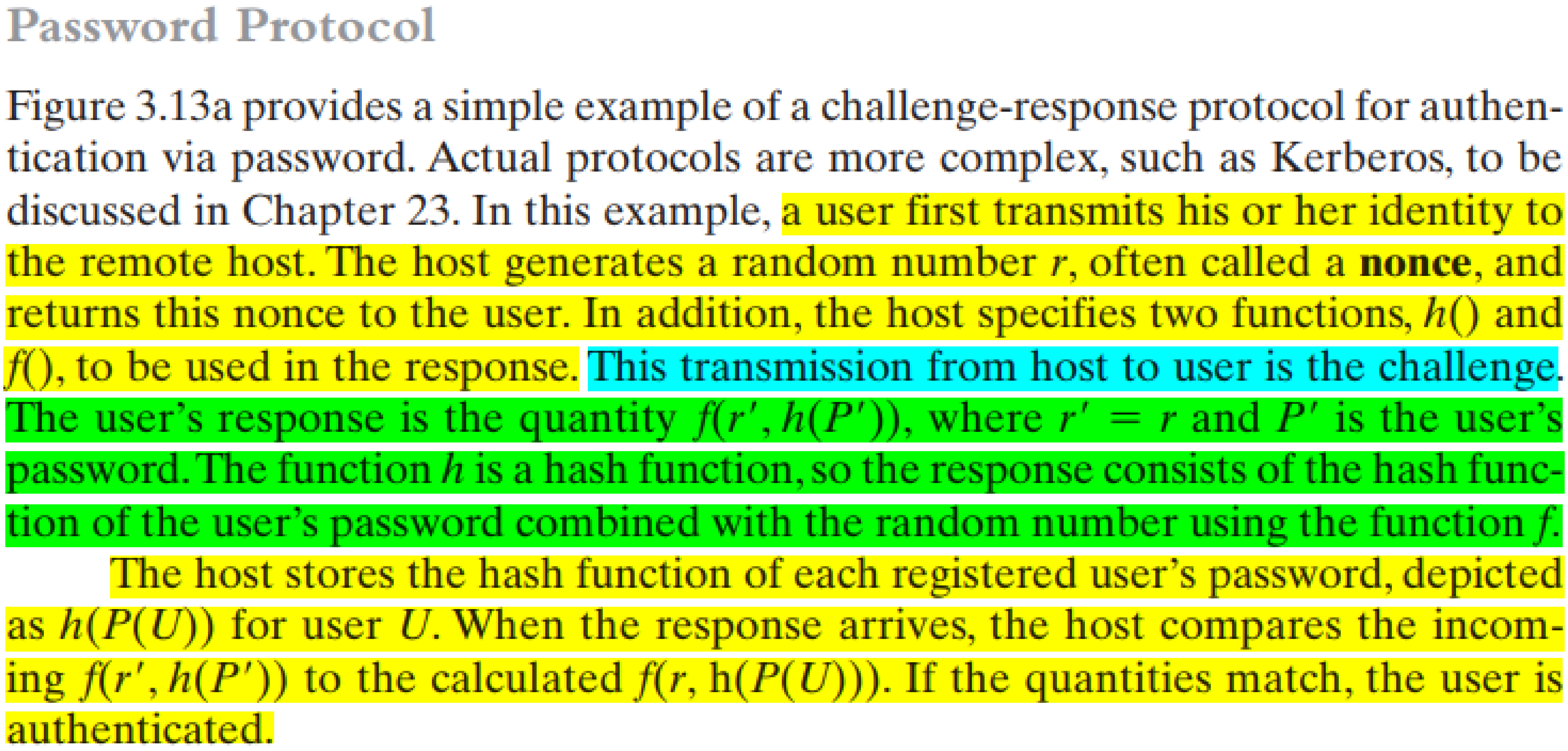




3.5 REMOTE USER AUTHENTICATION

The simplest form of user authentication is local authentication, in which a user attempts to access a system that is locally present, such as a stand-alone office PC or an ATM machine. The more complex case is that of remote user authentication, which takes place over the Internet, a network, or a communications link. Remote user authentication raises additional security threats, such as an eavesdropper being able to capture a password, or an adversary replaying an authentication sequence that has been observed.

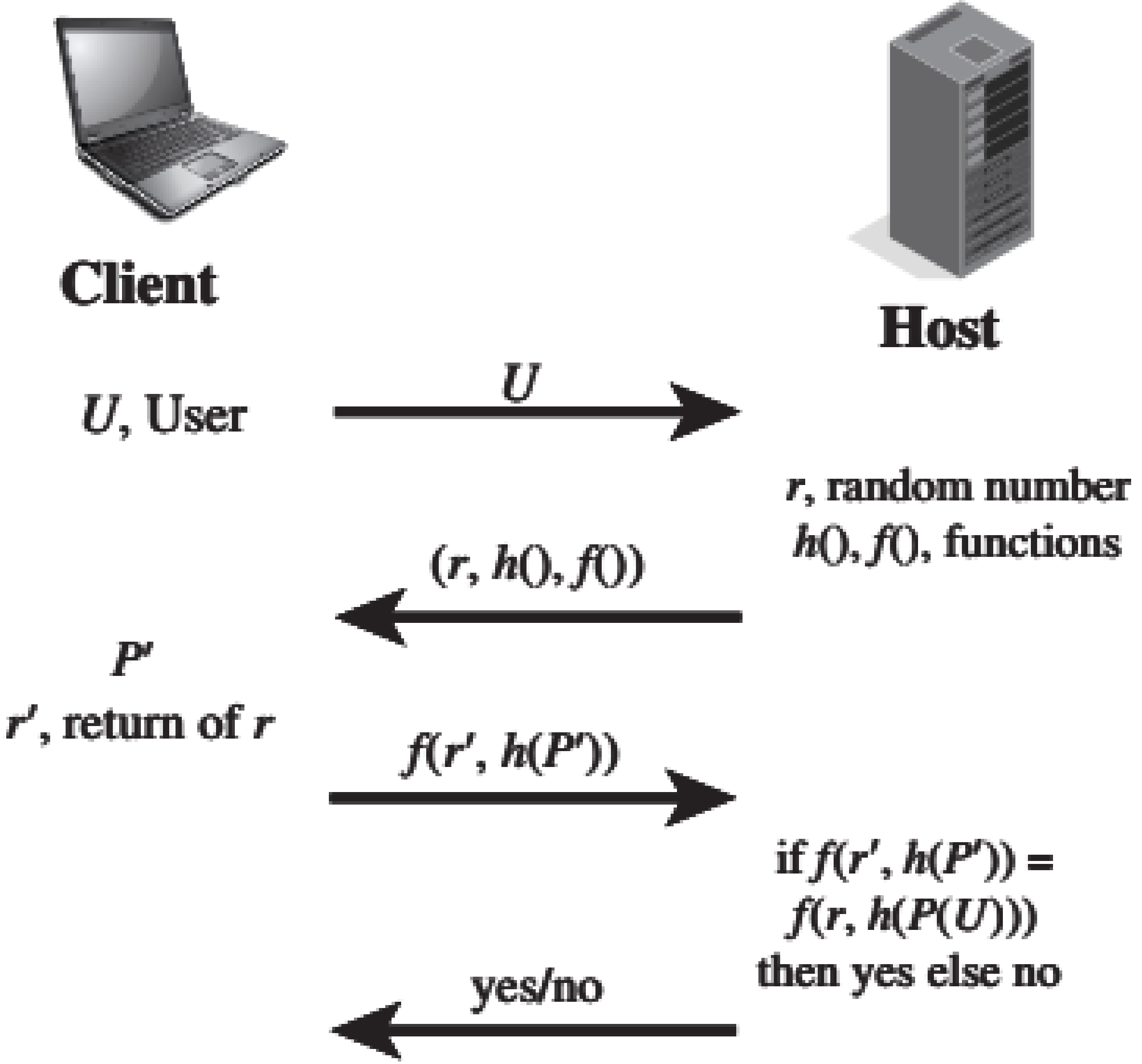
To counter threats to remote user authentication, systems generally rely on some form of challenge-response protocol. In this section, we present the basic elements of such protocols for each of the types of authenticators discussed in this chapter.



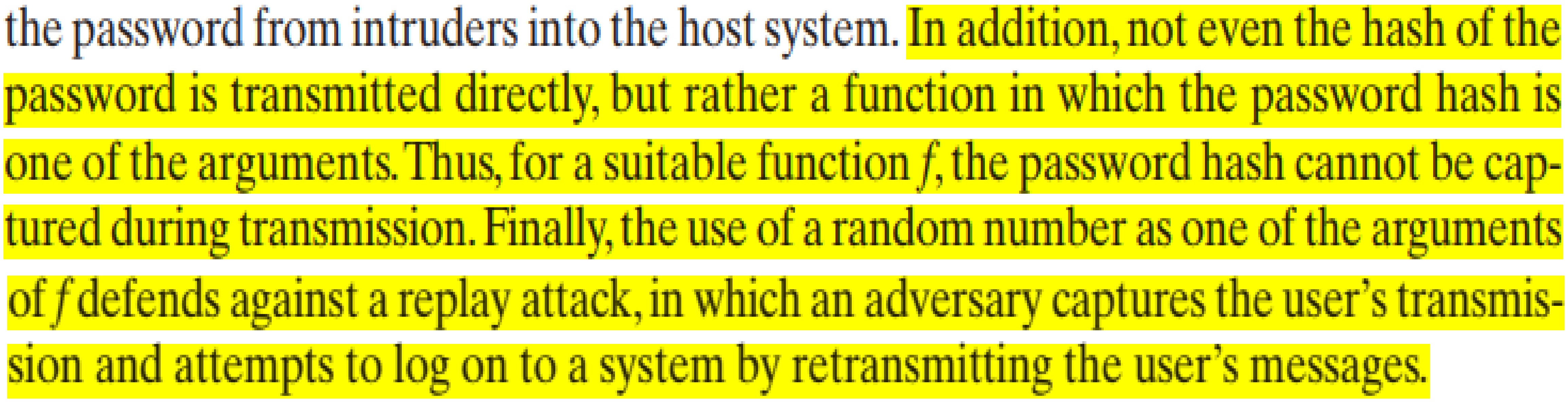
Password Protocol

Figure 3.13a provides a simple example of a challenge-response protocol for authentication via password. Actual protocols are more complex, such as Kerberos, to be discussed in Chapter 23. In this example, a user first transmits his or her identity to the remote host. The host generates a random number r, often called a **nonce**, and returns this nonce to the user. In addition, the host specifies two functions, $h()$ and $f()$, to be used in the response. This transmission from host to user is the challenge. The user's response is the quantity $f(r', h(P'))$, where $r' = r$ and P' is the user's password. The function h is a hash function, so the response consists of the hash function of the user's password combined with the random number using the function f.

The host stores the hash function of each registered user's password, depicted as $h(P(U))$ for user U. When the response arrives, the host compares the incoming $f(r', h(P'))$ to the calculated $f(r, \mathrm{h}(P(U)))$. If the quantities match, the user is authenticated.

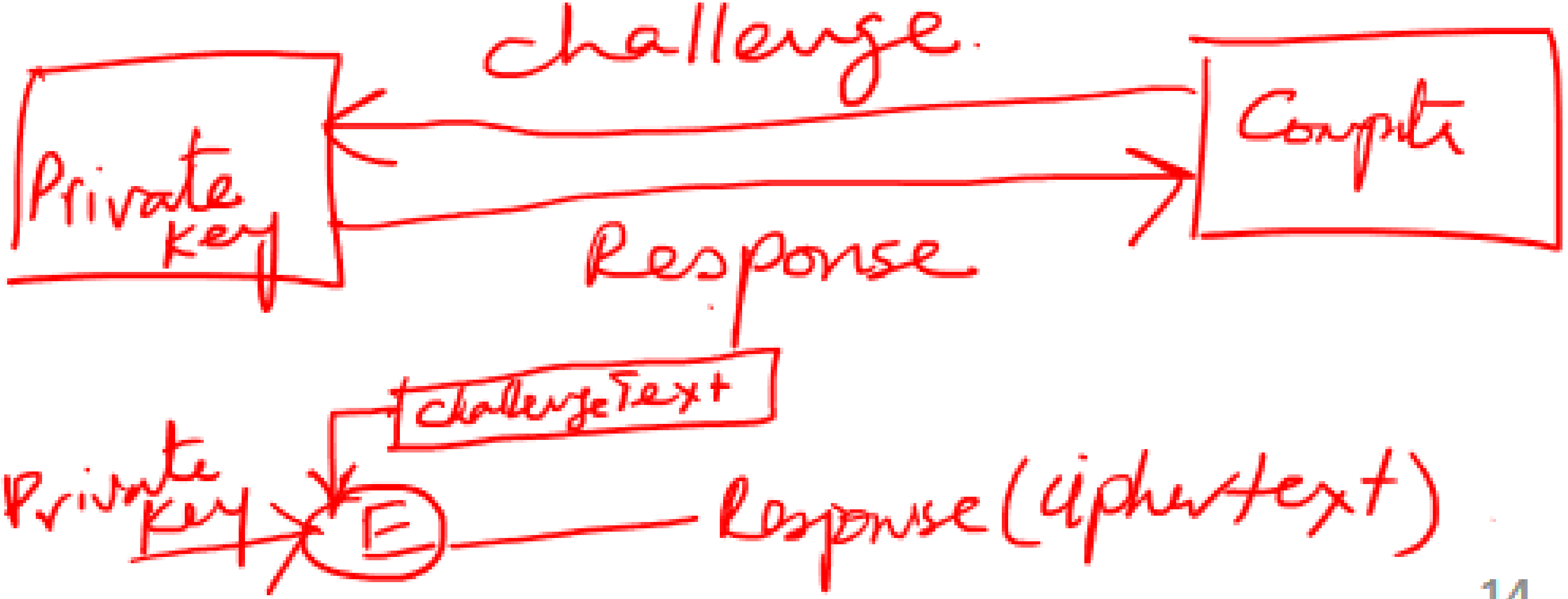


(a) Protocol for a password



the password from intruders into the host system. In addition, not even the hash of the password is transmitted directly, but rather a function in which the password hash is one of the arguments. Thus, for a suitable function *f*, the password hash cannot be captured during transmission. Finally, the use of a random number as one of the arguments of *f* defends against a replay attack, in which an adversary captures the user’s transmission and attempts to log on to a system by retransmitting the user’s messages.

Challenge-response: In this case, the computer system generates a challenge, such as a random string of numbers. The smart token generates a response based on the challenge. For example, public-key cryptography could be used and the token could encrypt the challenge string with the token's private key.



3.6 SECURITY ISSUES FOR USER AUTHENTICATION

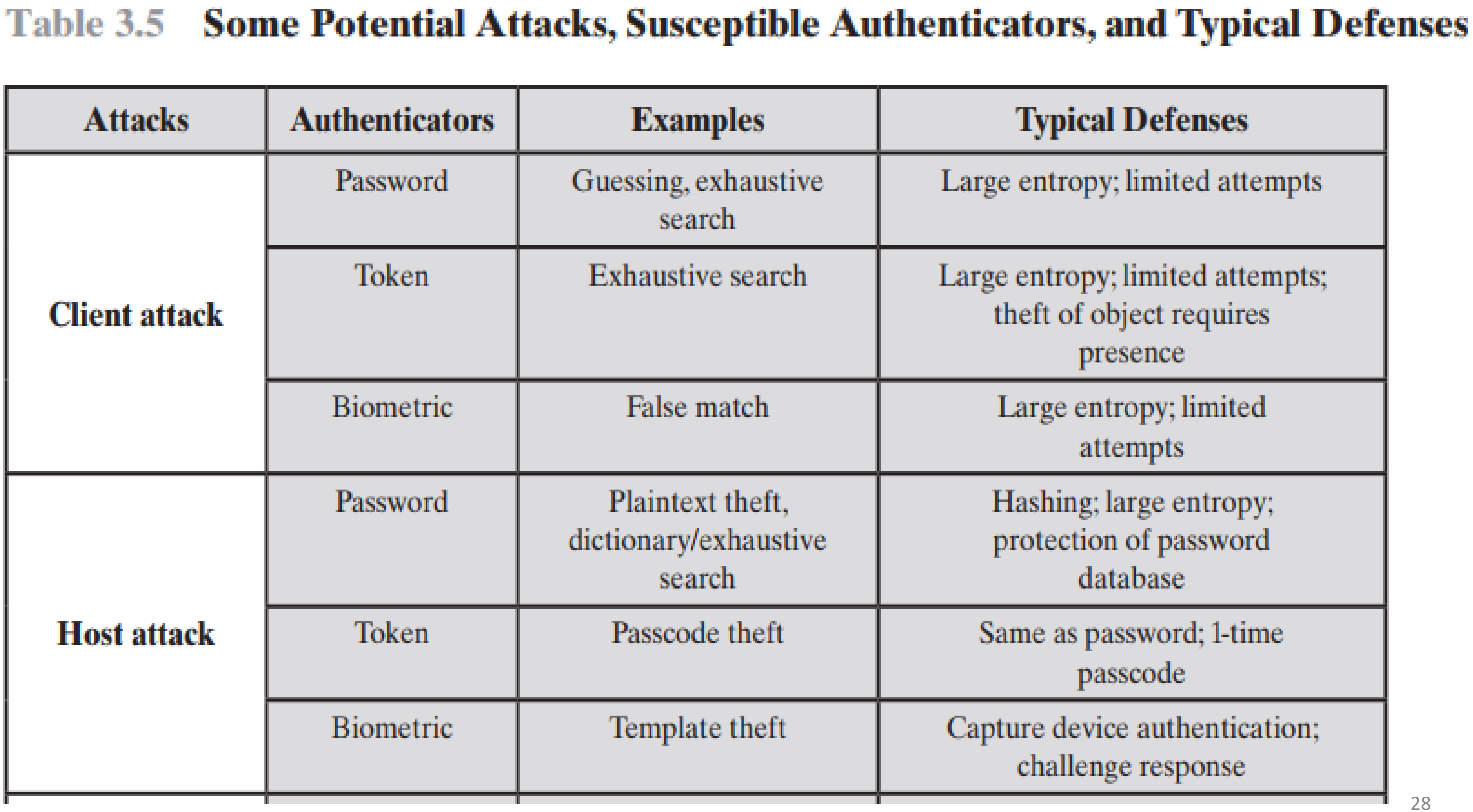
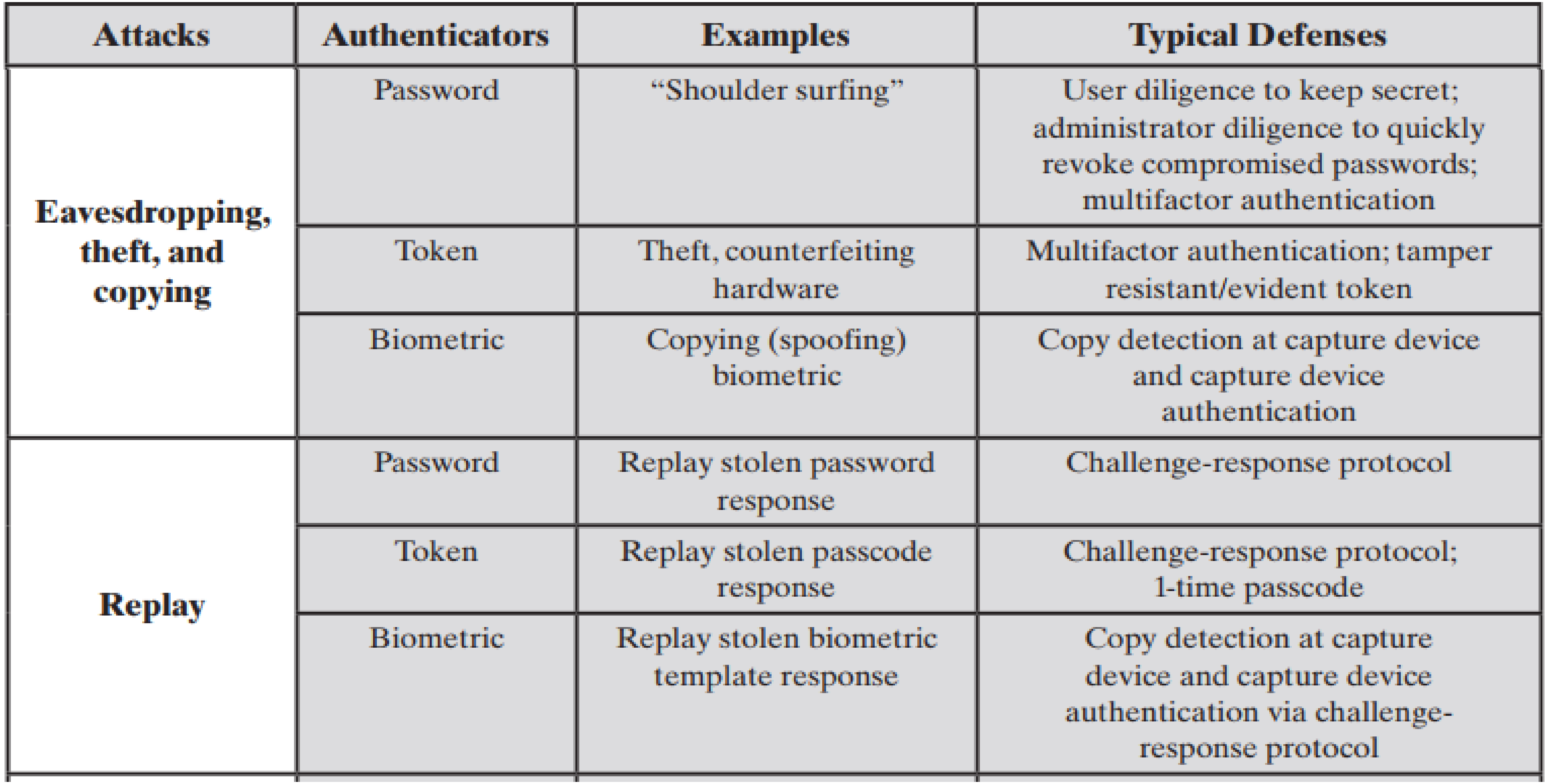


Table 3.5 **Some Potential Attacks, Susceptible Authenticators, and Typical Defenses**

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

Table 3.5 Some Potential Attacks, Susceptible Authenticators, and Typical Defenses



Attacks	Authenticators	Examples	Typical Defenses
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

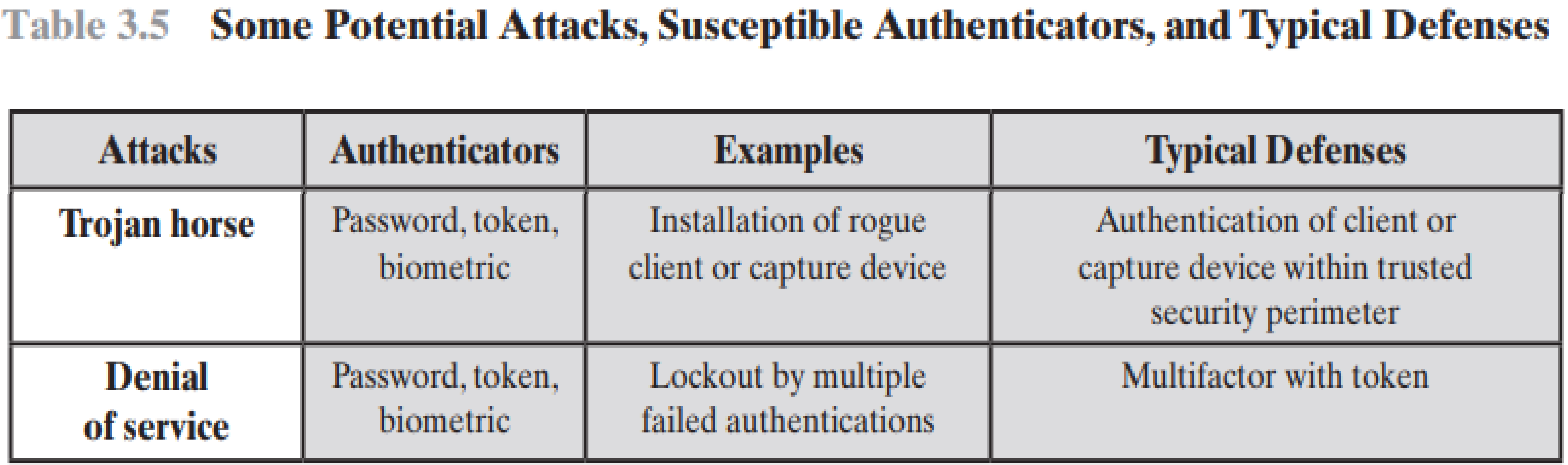


Table 3.5 **Some Potential Attacks, Susceptible Authenticators, and Typical Defenses**

Attacks	Authenticators	Examples	Typical Defenses
Trojan horse	Password, token, biometric	Installation of rogue client or capture device	Authentication of client or capture device within trusted security perimeter
Denial of service	Password, token, biometric	Lockout by multiple failed authentications	Multifactor with token