Table 3.3 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

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.

The ePass function is an offline function. That is, it is not used over a network but is used in a situation where the cardholder presents the card for a particular service at that location, such as going through a passport control checkpoint.

The eID function can be used for both online and offline services. An example of an offline use is an inspection system. An inspection system is a terminal for law enforcement checks, for example, by police or border control officers. An inspection system can read identifying information of the cardholder as well as biometric information stored on the card, such as facial image and fingerprints. The biometric information can be used to verify that the individual in possession of the card is the actual cardholder.

User authentication is a good example of online use of the eID function. Figure 3.6 illustrates a Web-based scenario. To begin, an eID user visits a Web site and requests a service that requires authentication. The Web site sends back

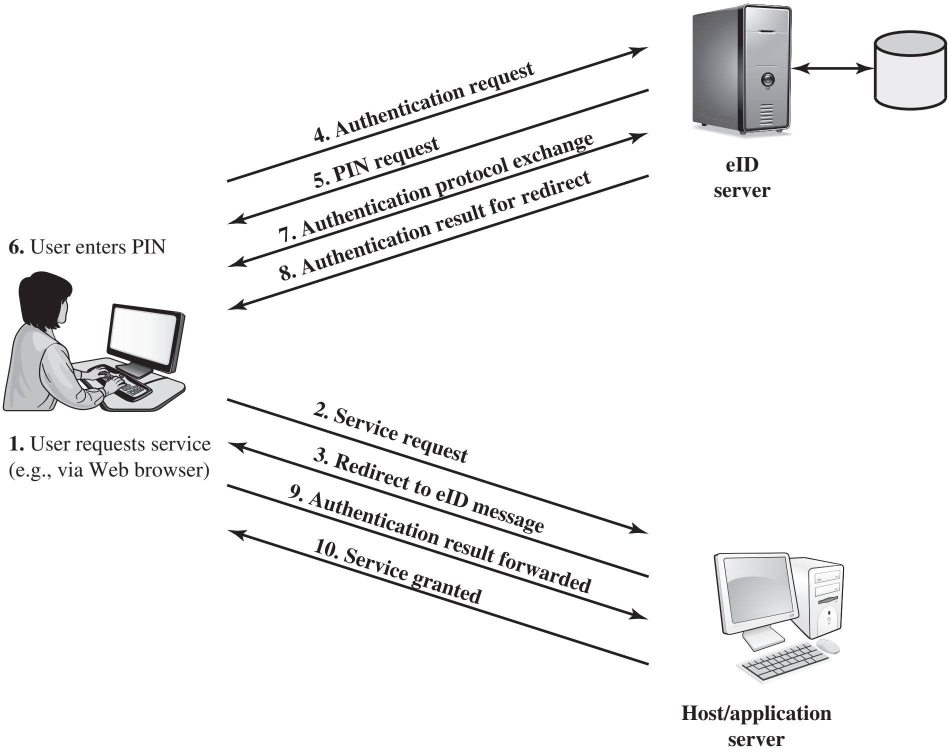

4. Authentication request
5. PIN request
7. Authentication protocol exchange
8. Authentication result for redirect
eID server
6. User enters PIN
1. User requests service (e.g., via Web browser)
2. Service request
3. Redirect to eID message
9. Authentication result forwarded
10. Service granted
Host/application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

Figure 3.6 **User Authentication with eID**

a redirect message that forwards an authentication request to an eID server. The eID server requests that the user enter the PIN number for the eID card. Once the user has correctly entered the PIN, data can be exchanged between the eID card and the terminal reader in encrypted form. The server then engages in an authentication protocol exchange with the microprocessor on the eID card. If the user is authenticated the results are sent back to the user system to be redirected to the Web server application.

For the preceding scenario, the appropriate software and hardware are required on the user system. Software on the main user system includes functionality for requesting and accepting the PIN number and for message redirection. The hardware required is an eID card reader. The card reader can be an external contact or contactless reader or a contactless reader internal to the user system.

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.

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

A number of different types of physical characteristics are either in use or under study for user authentication. The most common are the following:

- **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 characteris-

tics 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 3.7 gives a rough indication of the relative cost and accuracy of these biometric measures. The concept of accuracy does not apply to user authentication schemes using smart cards or passwords. For example, if a user enters a password, it either matches exactly the password expected for that user or not. In the case of biometric parameters, the system instead must determine how closely a presented biometric characteristic matches a stored characteristic. Before elaborating on the concept of biometric accuracy, we need to have a general idea of how biometric systems work.

Operation of a Biometric Authentication System

Figure 3.8 illustrates the operation of a biometric system. Each individual who is to be included in the database of authorized users must first be **enrolled** in the system. This is analogous to assigning a password to a user. For a biometric system, the user presents a name and, typically, some type of password or PIN to the system. At the same time the system senses some biometric characteristic of this user (e.g., fingerprint of right index finger). The system digitizes the input and then extracts a set of features that can be stored as a number or set of numbers representing this unique biometric characteristic; this set of numbers is referred to as the user's template. The user is now enrolled in the system, which maintains for the user a name (ID), perhaps a PIN or password, and the biometric value.

Depending on application, user authentication on a biometric system involves either **verification** or **identification**. Verification is analogous to a user logging on to a system by using a memory card or smart card coupled with a password or PIN. For biometric verification, the user enters a PIN and also uses a biometric sensor. The system extracts the corresponding feature and compares that to the template stored for this user. If there is a match, then the system authenticates this user.

For an identification system, the individual uses the biometric sensor but presents no additional information. The system then compares the presented template with the set of stored templates. If there is a match, then this user is identified. Otherwise, the user is rejected.

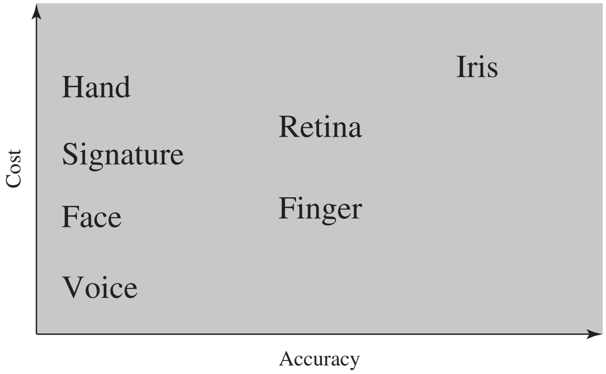
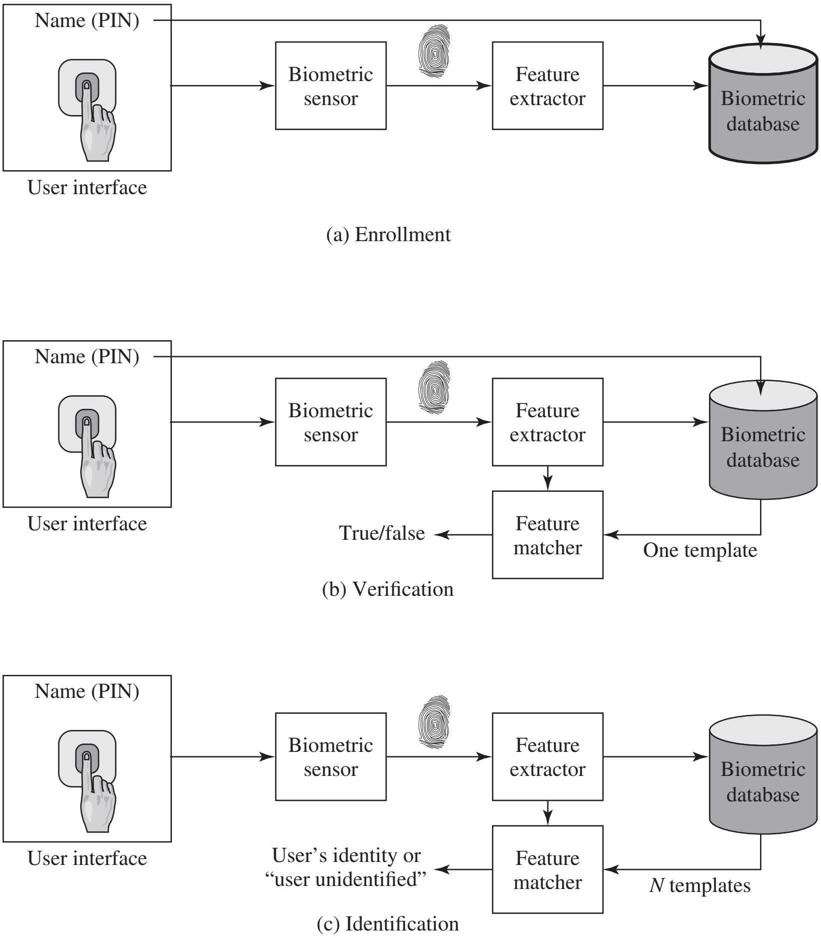


Figure 3.7 Cost versus Accuracy of Various Biometric Characteristics in User Authentication Schemes



(a) Enrollment

(b) Verification

(c) Identification

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.

Biometric Accuracy

In any biometric scheme, some physical characteristic of the individual is mapped into a digital representation. For each individual, a single digital representation, or template, is stored in the computer. When the user is to be authenticated, the system compares the stored template to the presented template. Given the complexities of physical characteristics, we cannot expect that there will be an exact match between the two templates. Rather, the system uses an algorithm to generate a matching score (typically a single number) that quantifies the similarity between the input and the stored template. To proceed with the discussion, we define the following terms. The false match rate is the frequency with which biometric samples from different sources are erroneously assessed to be from the same source. The false nonmatch rate is the frequency with which samples from the same source are erroneously assessed to be from different sources.

Figure 3.9 illustrates the dilemma posed to the system. If a single user is tested by the system numerous times, the matching score *s* will vary, with a probability density function typically forming a bell curve, as shown. For example, in the case of a fingerprint, results may vary due to sensor noise; changes in the print due to swelling or dryness; finger placement; and so on. On average, any other individual should have a much lower matching score but again will exhibit a bell-shaped probability density function. The difficulty is that the range of matching scores produced by two individuals, one genuine and one an imposter, compared to a given reference template, are likely to overlap. In Figure 3.9 a threshold value is selected thus that if the presented value s ≥ t a match is assumed, and for $s < t$, a mismatch is assumed. The shaded part to the right of *t* indicates a range of values for which a false match is possible, and the shaded part to the left indicates a range of values for which a false nonmatch is possible. A false match results in the acceptance of a user who should not be accepted, and a false mismatch triggers the rejection of a valid user. The area of each shaded part represents the probability of a false match or nonmatch, respectively. By moving the threshold, left or right, the probabilities can be altered, but note that a decrease in false match rate results in an increase in false nonmatch rate, and vice versa.

For a given biometric scheme, we can plot the false match versus false nonmatch rate, called the operating characteristic curve. Figure 3.10 shows idealized curves for two different systems. The curve that is lower and to the left performs better. The dot on the curve corresponds to a specific threshold for biometric testing. Shifting the threshold along the curve up and to the left provides greater security and the cost of decreased convenience. The inconvenience comes from a valid user being denied access

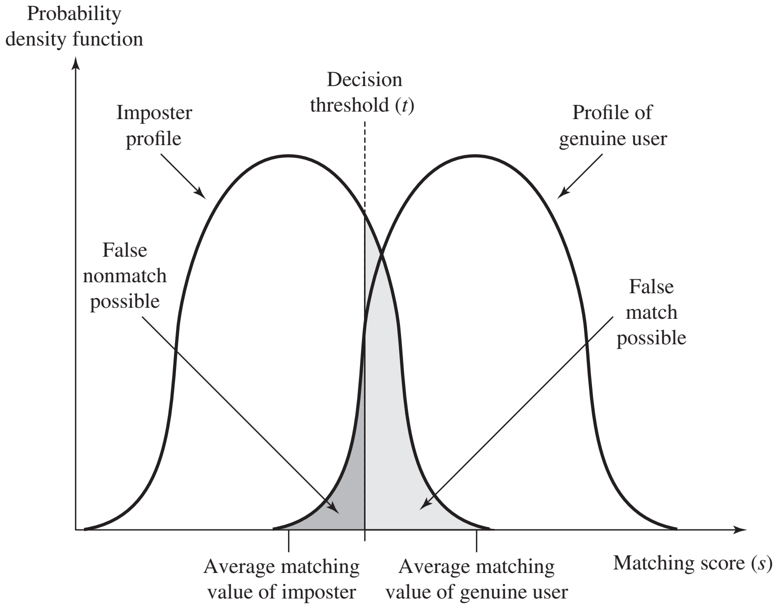


Figure 3.9 Profiles of a Biometric Characteristic of an Imposter and an Authorized User In this depiction, the comparison between the presented feature and a reference feature is reduced to a single numeric value. If the input value (*s*) is greater than a preassigned threshold (*t*), a match is declared.

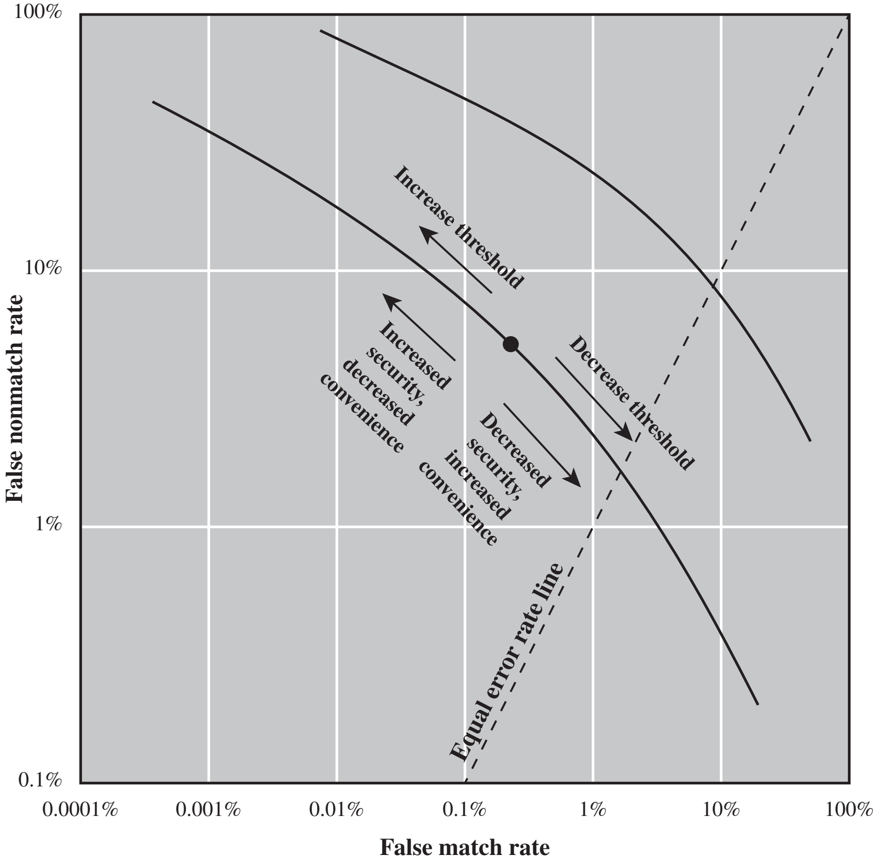


Figure 3.10 Idealized Biometric Measurement Operating Characteristic Curves (log-log scale)

and being required to take further steps. A plausible tradeoff is to pick a threshold that corresponds to a point on the curve where the rates are equal. A high-security application may require a very low false match rate, resulting in a point farther to the left on the curve. For a forensic application, in which the system is looking for possible candidates, to be checked further, the requirement may be for a low false nonmatch rate.

Figure 3.11 shows characteristic curves developed from actual product testing. The iris system had no false matches in over 2 million cross-comparisons. Note that over a broad range of false match rates, the face biometric is the worst performer.

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,

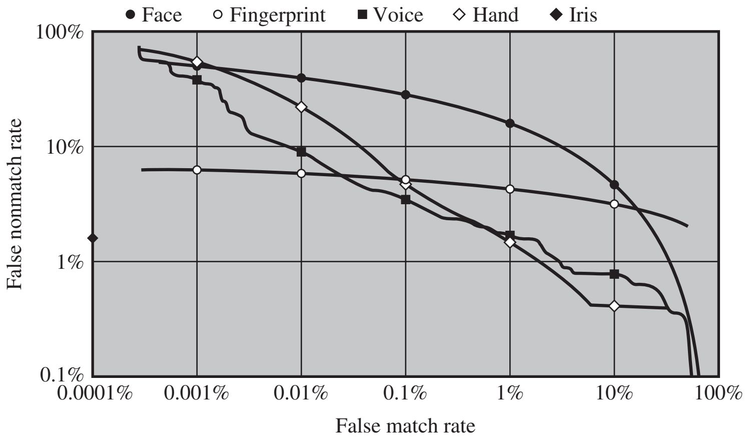


Figure 3.11 Actual Biometric Measurement Operating Characteristic Curves, Reported in [MANSO1] To clarify differences among systems, a log-log scale is used.

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.

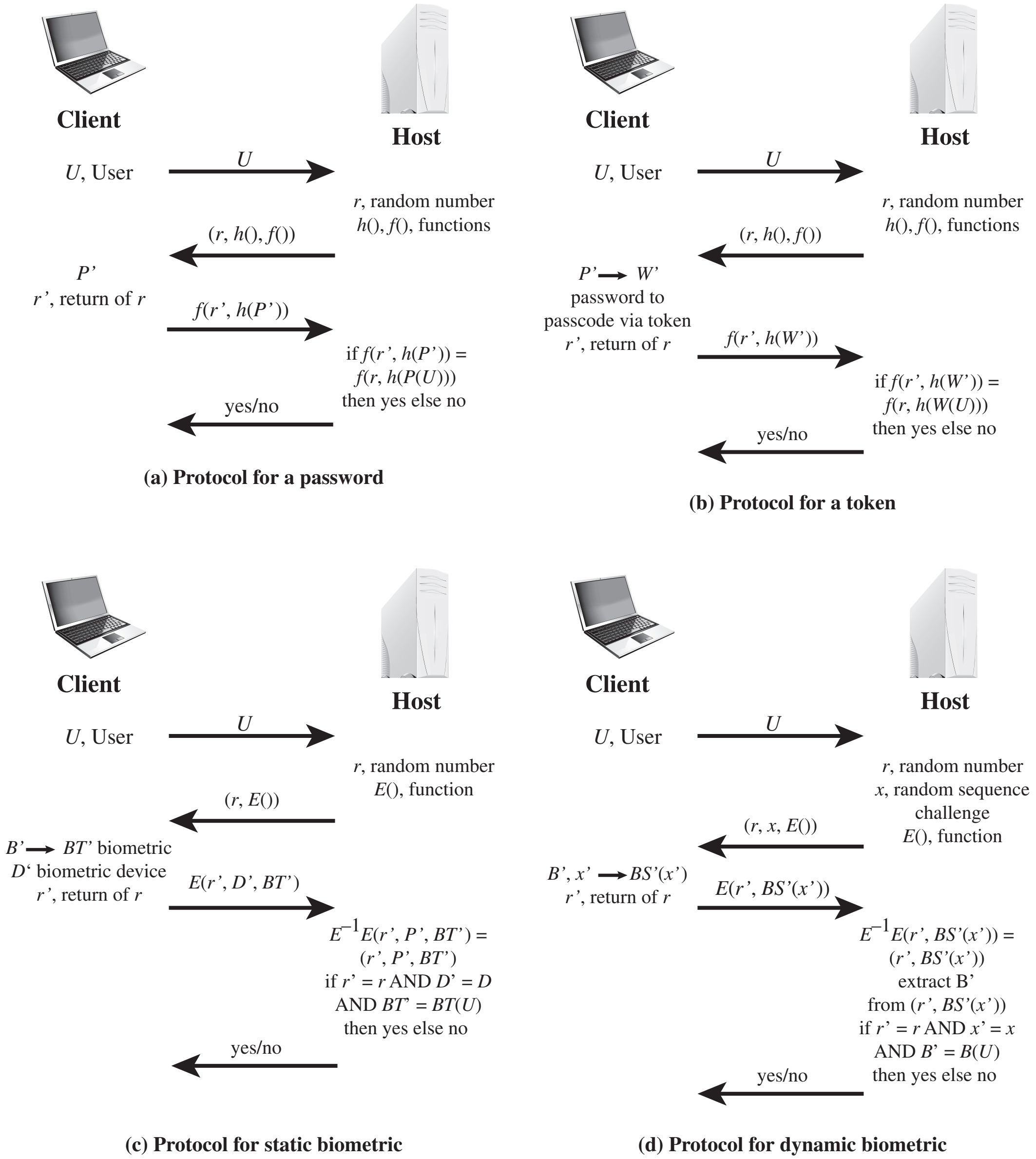
To counter threats to remote user authentication, systems generally rely on some form of challenge-response protocol. In this section, we present the basic elements of such protocols for each of the types of authenticators discussed in this chapter.

Password Protocol

Figure 3.12a provides a simple example of a challenge-response protocol for authentication via password. Actual protocols are more complex, such as Kerberos, 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 that 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, h($P(U)$)). If the quantities match, the user is authenticated.

This scheme defends against several forms of attack. The host stores not the password but a hash code of the password. As discussed in Section 3.2, this secures 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



(a) Protocol for a password

(b) Protocol for a token

(c) Protocol for static biometric

(d) Protocol for dynamic biometric

Figure 3.12 **Basic Challenge-Response Protocols for Remote User Authentication**
Source: Based on [OGOR03].

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.

Token Protocol

Figure 3.12b provides a simple example of a token protocol for authentication. As before, a user first transmits his or her identity to the remote host. The host returns a random number and the identifiers of functions f() and h() to be used in the response. At the user end, the token provides a passcode W'. The token either stores a static passcode or generates a one-time random passcode. For a one-time random passcode, the token must be synchronized in some fashion with the host. In either case, the user activates the passcode by entering a password P'. This password is shared only between the user and the token and does not involve the remote host. The token responds to the host with the quantity f(r', h(W')). For a static passcode, the host stores the hashed value h($W(U)$); for a dynamic passcode, the host generates a one-time passcode (synchronized to that generated by the token) and takes its hash. Authentication then proceeds in the same fashion as for the password protocol.

Static Biometric Protocol

Figure 3.12c is an example of a user authentication protocol using a static biometric. As before, the user transmits an ID to the host, which responds with a random number r and, in this case, the identifier for an encryption E(). On the user side is a client system that controls a biometric device. The system generates a biometric template BT' from the user's biometric B' and returns the ciphertext E(r', D', BT'), where D' identifies this particular biometric device. The host decrypts the incoming message to recover the three transmitted parameters and compares these to locally stored values. For a match, the host must find $r' = r$. Also, the matching score between BT' and the stored template must exceed a predefined threshold. Finally, the host provides a simple authentication of the biometric capture device by comparing the incoming device ID to a list of registered devices at the host database.

Dynamic Biometric Protocol

Figure 3.12d is an example of a user authentication protocol using a dynamic biometric. The principal difference from the case of a stable biometric is that the host provides a random sequence as well as a random number as a challenge. The sequence challenge is a sequence of numbers, characters, or words. The human user at the client end must then vocalize (speaker verification), type (keyboard dynamics verification), or write (handwriting verification) the sequence to generate a biometric signal $BS'(x')$. The client side encrypts the biometric signal and the random number. At the host side, the incoming message is decrypted. The incoming random number r' must be an exact match to the random number that was originally used as a challenge (r). In addition, the host generates a comparison based on the incoming biometric signal $BS'(x')$, the stored template $BT(U)$ for this user and the original signal x. If the comparison value exceeds a predefined threshold, the user is authenticated.

3.6 SECURITY ISSUES FOR USER AUTHENTICATION

As with any security service, user authentication, particularly remote user authentication, is subject to a variety of attacks. Table 3.4, from [OGOR03], summarizes the principal attacks on user authentication, broken down by type of authenticator. Much of the table is self-explanatory. In this section, we expand on some of the table's entries.

Table 3.4 **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
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
Denial of service	Password, token, biometric	Lockout by multiple failed authentications	Multifactor with token

Client attacks are those in which an adversary attempts to achieve user authentication without access to the remote host or to the intervening communications path. The adversary attempts to masquerade as a legitimate user. For a password-based system, the adversary may attempt to guess the likely user password. Multiple guesses may be made. At the extreme, the adversary sequences through all possible passwords in an exhaustive attempt to succeed. One way to thwart such an attack is to select a password that is both lengthy and unpredictable. In effect, such a password has large entropy; that is, many bits are required to represent the password. Another countermeasure is to limit the number of attempts that can be made in a given time period from a given source.

A token can generate a high-entropy passcode from a low-entropy PIN or password, thwarting exhaustive searches. The adversary may be able to guess or acquire the PIN or password but must additionally acquire the physical token to succeed.

Host attacks are directed at the user file at the host where passwords, token passcodes, or biometric templates are stored. Section 3.2 discusses the security considerations with respect to passwords. For tokens, there is the additional defense of using one-time passcodes, so that passcodes are not stored in a host passcode file. Biometric features of a user are difficult to secure because they are physical features of the user. For a static feature, biometric device authentication adds a measure of protection. For a dynamic feature, a challenge-response protocol enhances security.

Eavesdropping in the context of passwords refers to an adversary's attempt to learn the password by observing the user, finding a written copy of the password, or some similar attack that involves the physical proximity of user and adversary. Another form of eavesdropping is keystroke logging (keylogging), in which malicious hardware or software is installed so that the attacker can capture the user's keystrokes for later analysis. A system that relies on multiple factors (e.g., password plus token or password plus biometric) is resistant to this type of attack. For a token, an analogous threat is **theft** of the token or physical copying of the token. Again, a multifactor protocol resists this type of attack better than a pure token protocol. The analogous threat for a biometric protocol is **copying** or imitating the biometric parameter so as to generate the desired template. Dynamic biometrics are less susceptible to such attacks. For static biometrics, device authentication is a useful countermeasure.

Replay attacks involve an adversary repeating a previously captured user response. The most common countermeasure to such attacks is the challenge-response protocol.

In a **Trojan horse** attack, an application or physical device masquerades as an authentic application or device for the purpose of capturing a user password, passcode, or biometric. The adversary can then use the captured information to masquerade as a legitimate user. A simple example of this is a rogue bank machine used to capture user ID/password combinations.

A **denial-of-service** attack attempts to disable a user authentication service by flooding the service with numerous authentication attempts. A more selective attack denies service to a specific user by attempting logon until the threshold is reached that causes lockout to this user because of too many logon attempts. A multifactor authentication protocol that includes a token thwarts this attack, because the adversary must first acquire the token.

3.7 PRACTICAL APPLICATION: AN IRIS BIOMETRIC SYSTEM

As an example of a biometric user authentication system, we look at an iris biometric system that was developed for use by the United Arab Emirates (UAE) at border control points [DAUG04, TIRO05, NBSP08]. The UAE relies heavily on an outside workforce, and has increasingly become a tourist attraction. Accordingly, relative to its size, the UAE has a very substantial volume of incoming visitors. On a typical day, more than 6,500 passengers enter the UAE via seven international airports, three land ports, and seven sea ports. Handling a large volume of incoming visitors in an efficient and timely manner thus poses a significant security challenge. Of particular concern to the UAE are attempts by expelled persons to re-enter the country. Traditional means of preventing reentry involve identifying individuals by name, date of birth, and other text-based data. The risk is that this information can be changed after expulsion. An individual can arrive with a different passport with a different nationality and changes to other identifying information.

To counter such attempts, the UAE decided on using a biometric identification system and identified the following requirements:

- Identify a single person from a large population of people
- Rely on a biometric feature that does not change over time
- Use biometric features that can be acquired quickly
- Be easy to use
- Respond in real-time for mass transit applications
- Be safe and non-invasive
- Scale into the billions of comparisons and maintain top performance
- Be affordable

And chose iris recognition as the most efficient and foolproof method. No two irises are alike. There is no correlation between the iris patterns of even identical twins, or the right and left eye of an individual.

System implementation involves enrollment and identity checking. All expelled foreigners are subjected to an iris scan at one of the multiple enrollment centers. This information is merged into one central database. Iris scanners are installed at all 17 air, land, and sea ports into the UAE. An iris-recognition camera takes a black-and-white picture 5 to 24 inches from the eye, depending on the camera. The camera uses non-invasive, near-infrared illumination that is similar to a TV remote control, barely visible and considered extremely safe. The picture first is processed by software that localizes the inner and outer boundaries of the iris, and the eyelid contours, in order to extract just the iris portion. The software creates a so-called phase code for the texture of the iris, similar to a DNA sequence code. The unique features of the iris are captured by this code and can be compared against a large database of scanned irises to make a match. Over a distributed network (Figure 3.13) the iris codes of all arriving passengers are compared in real-time exhaustively against an enrolled central database.

Note that this is computationally a more demanding task than verifying an identity. In this case, the iris pattern of each incoming passenger is compared against the entire database of known patterns to determine if there is a match. Given the current volume of traffic and size of the database, the daily number of iris cross-comparisons is well over 9 billion.

As with any security system, adversaries are always looking for countermeasures. UAE officials had to adopt new security methods to detect if an iris has been dilated with eye drops before scanning. Expatriates who were banned from the

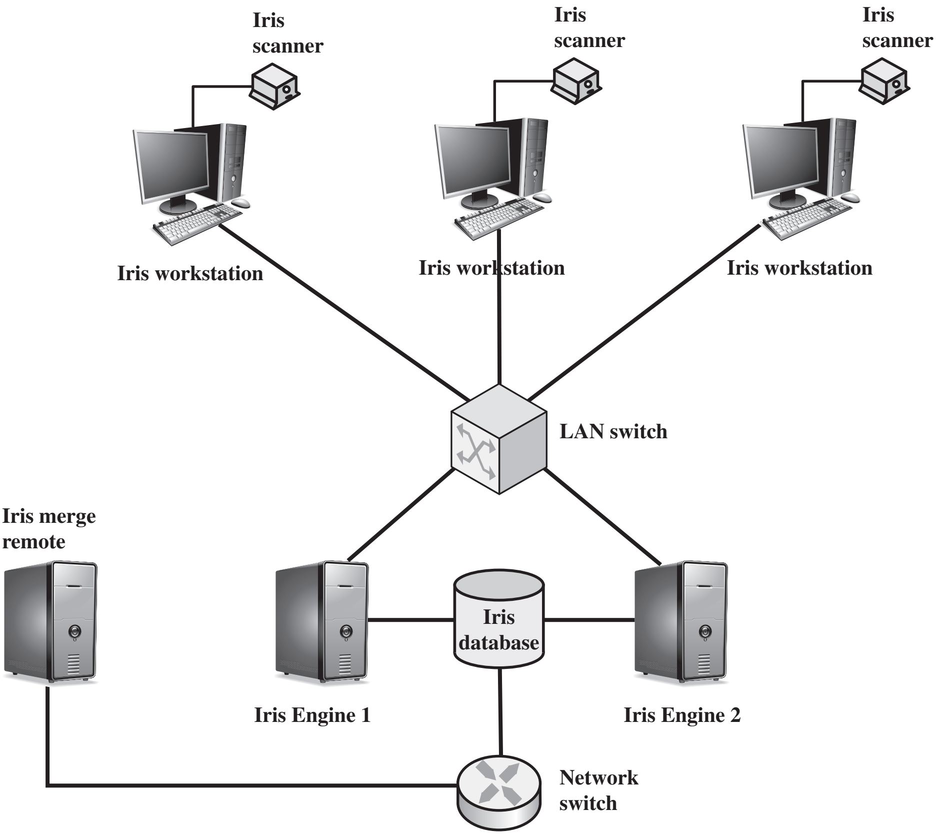


Figure 3.13 **General Iris Scan Site Architecture for UAE System**

UAE started using eye drops in an effort to fool the government's iris recognition system when they try to re-enter the country. A new algorithm and computerized step-by-step procedure has been adopted to help officials determine if an iris is in normal condition or an eye-dilating drop has been used.

3.8 CASE STUDY: SECURITY PROBLEMS FOR ATM SYSTEMS

Redspin, Inc., an independent auditor, recently released a report describing a security vulnerability in ATM (automated teller machine) usage that affects a number of small to mid-size ATM card issuers. This vulnerability provides a useful case study illustrating that cryptographic functions and services alone do not guarantee security; they must be properly implemented as part of a system.

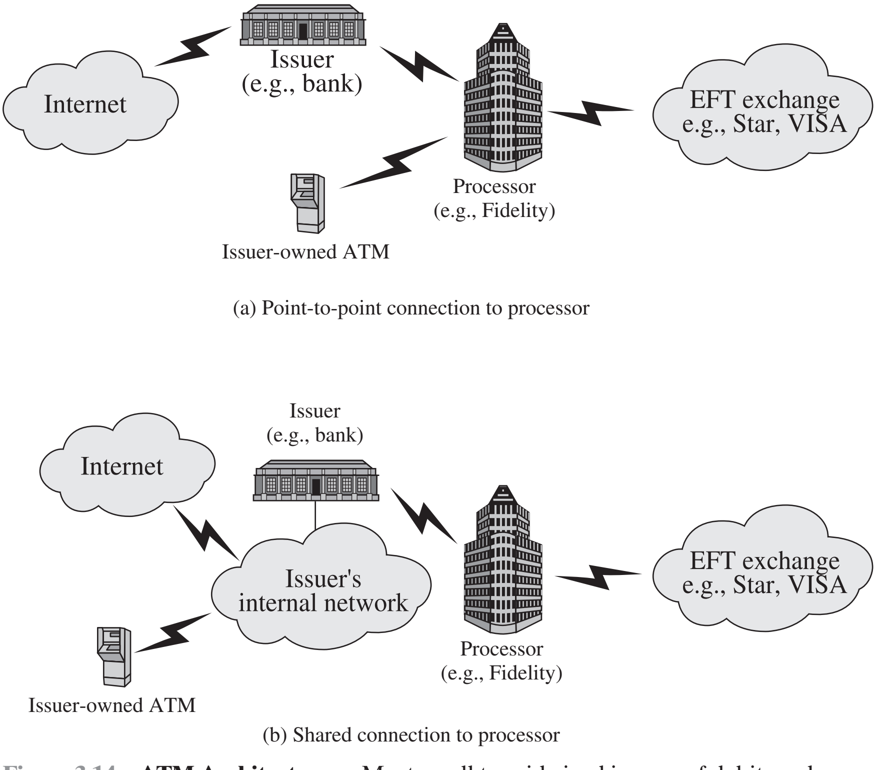
We begin by defining terms used in this section:

- **Cardholder:** An individual to whom a debit card is issued. Typically, this individual is also responsible for payment of all charges made to that card.

- **Issuer:** An institution that issues debit cards to cardholders. This institution is responsible for the cardholder's account and authorizes all transactions. Banks and credit unions are typical issuers.
- **Processor:** An organization that provides services such as core data processing (PIN recognition and account updating), electronic funds transfer (EFT), and so on to issuers. EFT allows an issuer to access regional and national networks that connect point of sale (POS) devices and ATMs worldwide. Examples of processing companies include Fidelity National Financial and Jack Henry & Associates.

Customers expect 24/7 service at ATM stations. For many small to mid-sized issuers, it is more cost-effective for contract processors to provide the required data processing and EFT/ATM services. Each service typically requires a dedicated data connection between the issuer and the processor, using a leased line or a virtual leased line.

Prior to about 2003, the typical configuration involving issuer, processor, and ATM machines could be characterized by Figure 3.14a. The ATM units linked directly to the processor rather than to the issuer that owned the ATM, via leased or virtual leased line. The use of a dedicated link made it difficult to maliciously



(a) Point-to-point connection to processor

(b) Shared connection to processor

Figure 3.14 ATM Architectures Most small to mid-sized issuers of debit cards contract processors to provide core data processing and electronic funds transfer (EFT) services. The bank's ATM machine may link directly to the processor or to the bank.

intercept transferred data. To add to the security, the PIN portion of messages transmitted from ATM to processor was encrypted using DES (Data Encryption Standard). Processors have connections to EFT (electronic funds transfer) exchange networks to allow cardholders access to accounts from any ATM. With the configuration of Figure 3.14a, a transaction proceeds as follows. A user swipes her card and enters her PIN. The ATM encrypts the PIN and transmits it to the processor as part of an authorization request. The processor updates the customer's information and sends a reply.

In the early 2000s, banks worldwide began the process of migrating from an older generation of ATMs using IBM's OS/2 operating system to new systems running Windows. The mass migration to Windows has been spurred by a number of factors, including IBM's decision to stop supporting OS/2 by 2006, market pressure from creditors such as MasterCard International and Visa International to introduce stronger Triple DES, and pressure from U.S. regulators to introduce new features for disabled users. Many banks, such as those audited by Redspin, included a number of other enhancements at the same time as the introduction of Windows and triple DES, especially the use of TCP/IP as a network transport.

Because issuers typically run their own Internet-connected local area networks (LANs) and intranets using TCP/IP, it was attractive to connect ATMs to these issuer networks and maintain only a single dedicated line to the processor, leading to the configuration illustrated in Figure 3.14b. This configuration saves the issuer expensive monthly circuit fees and enables easier management of ATMs by the issuer. In this configuration, the information sent from the ATM to the processor traverses the issuer's network before being sent to the processor. It is during this time on the issuer's network that the customer information is vulnerable.

The security problem was that with the upgrade to a new ATM OS and a new communications configuration, the only security enhancement was the use of triple DES rather than DES to encrypt the PIN. The rest of the information in the ATM request message is sent in the clear. This includes the card number, expiration date, account balances, and withdrawal amounts. A hacker tapping into the bank's network, either from an internal location or from across the Internet potentially would have complete access to every single ATM transaction.

The situation just described leads to two principal vulnerabilities:

- **Confidentiality:** The card number, expiration date, and account balance can be used for online purchases or to create a duplicate card for signature-based transactions.
- **Integrity:** There is no protection to prevent an attacker from injecting or altering data in transit. If an adversary is able to capture messages en route, the adversary can masquerade as either the processor or the ATM. Acting as the processor, the adversary may be able to direct the ATM to dispense money without the processor ever knowing that a transaction has occurred. If an adversary captures a user's account information and encrypted PIN, the account is compromised until the ATM encryption key is changed, enabling the adversary to modify account balances or effect transfers.

Redspin recommended a number of measures that banks can take to counter these threats. Short-term fixes include segmenting ATM traffic from the rest of the network either by implementing strict firewall rule sets or physically dividing the networks altogether. An additional short-term fix is to implement network-level encryption between routers that the ATM traffic traverses.

Long-term fixes involve changes in the application-level software. Protecting confidentiality requires encrypting all customer-related information that traverses the network. Ensuring data integrity requires better machine-to-machine authentication between the ATM and processor and the use of challenge-response protocols to counter replay attacks.

3.9 RECOMMENDED READING

[OGOR03] is the paper to read for an authoritative survey of the topics of this chapter. [BURR13] is also a worthwhile survey. [SCAR09] is a comprehensive look at many issues related to password selection and management.

BURR13 Burr, W, et al. *Electronic Authentication Guideline.* Gaithersburg, MD: National Institute of Standards and Technology, Special Publication 800–63–2, August 2013.

OGOR03 O'Gorman, L. "Comparing Passwords, Tokens and Biometrics for User Authentication." *Proceedings of the IEEE*, December 2003.

SCAR09a Scarfone, K., and Souppaya, M. *Guide to Enterprise Password Management (Draft)*. NIST Special Publication SP 800-118 (Draft), April 2009.

3.10 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

biometric
challenge-response protocol
claimant
credential
credential service provider (CSP)
dynamic biometric
enroll
hashed password
identification
memory card
nonce
password
rainbow table
registration authority (RA)
relying party (RP)
salt
shadow password file
smart card
static biometric
subscriber
token
user authentication
verification
verifier

Review Questions

3.1 In general terms, what are four means of authenticating a user's identity?

3.2 List and briefly describe the principal threats to the secrecy of passwords.

3.3 What are two common techniques used to protect a password file?

3.4 List and briefly describe four common techniques for selecting or assigning passwords.

3.5 Explain the difference between a simple memory card and a smart card.

3.6 List and briefly describe the principal physical characteristics used for biometric identification.

3.7 In the context of biometric user authentication, explain the terms, enrollment, verification, and identification.

3.8 Define the terms *false match rate* and *false nonmatch rate*, and explain the use of a threshold in relationship to these two rates.

3.9 Describe the general concept of a challenge-response protocol.

Problems

3.1 Explain the suitability or unsuitability of the following passwords:

a. YK 334 **b.** mfmitm (for "my favorite movie is tender mercies) **c.** Natalie1 **d.** Washington

e. Aristotle **f.** tv9stove **g.** 12345678 **h.** dribgib

3.2 An early attempt to force users to use less predictable passwords involved computer-supplied passwords. The passwords were eight characters long and were taken from the character set consisting of lowercase letters and digits. They were generated by a pseudorandom number generator with 2^{15} possible starting values. Using the technology of the time, the time required to search through all character strings of length 8 from a 36-character alphabet was 112 years. Unfortunately, this is not a true reflection of the actual security of the system. Explain the problem.

3.3 Assume that passwords are selected from four-character combinations of 26 alphabetic characters. Assume that an adversary is able to attempt passwords at a rate of one per second.

a. Assuming no feedback to the adversary until each attempt has been completed, what is the expected time to discover the correct password?

b. Assuming feedback to the adversary flagging an error as each incorrect character is entered, what is the expected time to discover the correct password?

3.4 Assume that source elements of length k are mapped in some uniform fashion into a target elements of length p. If each digit can take on one of r values, then the number of source elements is r^k and the number of target elements is the smaller number r^p. A particular source element x_i is mapped to a particular target element y_j.

a. What is the probability that the correct source element can be selected by an adversary on one try?

b. What is the probability that a different source element x_k ($x_i \neq x_k$) that results in the same target element, yj, could be produced by an adversary?

c. What is the probability that the correct target element can be produced by an adversary on one try?

3.5 A phonetic password generator picks two segments randomly for each six-letter password. The form of each segment is CVC (consonant, vowel, consonant), where $V = <$ a, e, i, o, u $>$ and $C = \overline{V}$.

a. What is the total password population?

b. What is the probability of an adversary guessing a password correctly?

3.6 Assume that passwords are limited to the use of the 95 printable ASCII characters and that all passwords are 10 characters in length. Assume a password cracker with an encryption rate of 6.4 million encryptions per second. How long will it take to test exhaustively all possible passwords on a UNIX system?

3.7 Because of the known risks of the UNIX password system, the SunOS-4.0 documentation recommends that the password file be removed and replaced with a publicly readable file called /etc/publickey. An entry in the file for user A consists of a user's identifier ID_A, the user's public key, PU_a, and the corresponding private key PR_a.

This private key is encrypted using DES with a key derived from the user's login password P_a. When A logs in, the system decrypts E(P_a, PR_a) to obtain PR_a.

a. The system then verifies that P_a was correctly supplied. How?

b. How can an opponent attack this system?

3.8 It was stated that the inclusion of the salt in the UNIX password scheme increases the difficulty of guessing by a factor of 4096. But the salt is stored in plaintext in the same entry as the corresponding ciphertext password. Therefore, those two characters are known to the attacker and need not be guessed. Why is it asserted that the salt increases security?

3.9 Assuming that you have successfully answered the preceding problem and understand the significance of the salt, here is another question. Wouldn't it be possible to thwart completely all password crackers by dramatically increasing the salt size to, say, 24 or 48 bits?

3.10 Consider the Bloom filter discussed in Section 3.3. Define k = number of hash functions; N = number of bits in hash table; and D = number of words in dictionary.

a. Show that the expected number of bits in the hash table that are equal to zero is expressed as

$$\phi = \left(1 - \frac{k}{N}\right)^D$$

b. Show that the probability that an input word, not in the dictionary, will be falsely accepted as being in the dictionary is

$$P = (1 - \phi)^k$$

c. Show that the preceding expression can be approximated as

$$P \approx \left(1 - \mathrm{e}^{-kD/N}\right)^k$$

3.11 For the biometric authentication protocols illustrated in Figure 3.12, note that the biometric capture device is authenticated in the case of a static biometric but not authenticated for a dynamic biometric. Explain why authentication is useful in the case of a stable biometric but not needed in the case of a dynamic biometric.

3.12 A relatively new authentication proposal is the Secure Quick Reliable Login (SQRL) described here: https://www.grc.com/sqrl/sqrl.htm. Write a brief summary of how SQRL works and indicate how it fits into the categories of types of user authentication listed in this chapter.