Unit-II User Authentication

Introduction:

User authentication verifies the identity of a user attempting to gain access to a network or computing resource by authorizing a human-to-machine transfer of credentials during interactions on a network to confirm a user's authenticity.

User authentication is the fundamental building block and the primary line of defense. User authentication is the basis for most types of access control and for user accountability.

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

An authentication process consists of two steps:

- **Identification step:** Presenting an identifier to the security system. (Identifiers should be assigned carefully, because authenticated identities are the basis for other security services, such as access control service.)
- **Verification step:** Presenting or generating authentication information that corroborates the binding between the entity and the identifier.

Identification is the means by which a user provides a claimed identity to the system; user authentication is the means of establishing the validity of the claim.

Definition: Authentication is the process of recognizing a user's identity. It is the mechanism of associating an incoming request with a set of identifying credentials.

Means of Authentication:

There are four general means of authenticating a user’s identity, which can be used alone or in combination:

- **Something the individual knows:** Examples includes a password, a personal identification number (PIN), or answers to a prearranged set of questions.
- **Something the individual possesses:** Examples include electronic key cards, smart cards, and physical keys. This type of authenticator is referred to as a token.
- **Something the individual is (static biometrics):** Examples include recognition by fingerprint, retina, and face.
- **Something the individual does (dynamic biometrics):** Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm.

All of these methods, properly implemented and used, can provide secure user authentication.

Password-Based Authentication:-

A widely used line of defense against intruders is the password system. 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 supervisory or "super user" status that enables them to read files and perform functions that are especially protected by the operating system.
- 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:

A system that uses password-based authentication maintains a password file indexed by user ID. One technique that is typically used is to store one-way hash function of the password.

Identification of attack strategies and countermeasures:

• Offline dictionary attack:

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.

• 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.

- **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.

- **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.

- **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.

The Use of Hashed Passwords

A widely used password security technique is the use of hashed passwords and salt value. To load a new password into the system, the user selects or is assigned a password. This password is combined with a fixed-length salt value. The password and salt value serve as inputs to a hashing algorithm to produce a fixed-length hash code. The hash algorithm is designed to be executing in order to thwart attacks. The hashed password is then stored, together with a plaintext copy of the salt value, in the password file for the corresponding user ID. When a user attempts to log on to a UNIX system, the user provides an ID and a password .The operating system uses the ID to index into the password file and retrieve the plaintext salt value and the encrypted password. The salt value and user-supplied passwords are used as input to the encryption routine. If the result matches the stored value, the password is accepted.

Password Cracking of User-Choosen Passwords

Password cracking is the process of using an application program to identify an unknown or forgotten password to a computer or network resource.

password cracking techniques

Password crackers use two primary methods to identify correct passwords: **brute-force** and **dictionary attacks**. However, there are plenty of other password cracking methods, including the following:

Brute force. This attack runs through combinations of characters of a predetermined length until it finds the combination that matches the password.
Dictionary search. Here, a password cracker searches each word in the dictionary for the correct password. Password dictionaries exist for a variety of topics and combinations of topics, including politics, movies and music groups.
Phishing. These attacks are used to gain access to user passwords without the use of a password cracking tool. Instead, a user is fooled into clicking on an email attachment. From here, the attachment could install malware or prompt the user to use their email to sign into a false version of a website, revealing their password.
Malware. Similar to phishing, using malware is another method of gaining unauthored access to passwords without the use of a password cracking tool. Malware such as keyloggers, which track keystrokes, or screen scrapers, which take screenshots, are used instead.
Rainbow attack. This approach involves using different words from the original password in order to generate other possible passwords. Malicious actors can keep a list called a rainbow table with them. This list contains leaked and previously cracked passwords, which will make the overall password cracking method more effective.
Guessing. An attacker may be able to guess a password without the use of tools. If the threat actor has enough information about the victim or the victim is using a common enough password, they may be able to come up with the correct characters.

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**. Special attention is paid to making the shadow password file protected from unauthorized access.

Instead of capturing the system password file, another approach to collecting user IDs and passwords is through sniffing network traffic.

Thus, a password protection policy must complement access control measures with techniques to force users to select passwords that are difficult to guess.

Password Selection Strategies:
If users are assigned passwords consisting of eight randomly selected printable characters, password cracking is effectively impossible.
Four basic techniques are in use:

- **User education**

user education strategy is unlikely to succeed at most installations, particularly where there is a large user population or a lot of turnover. Many users will simply ignore the guidelines. Others may not be good judges of what is a strong password. For example, many users (mistakenly) believe that reversing a word or capitalizing the last letter makes a password unguessable.

- **Computer-generated passwords**

Computer-generated passwords are quite random in nature. Computer-generated passwords also have problems. If the passwords are quite random in nature, users will not be able to remember them. Even if the password is pronounceable, the user may have difficulty remembering it and so be tempted to write it down. In general, computer-generated password schemes have a history of poor acceptance by users. FIPS PUB 181 defines one of the best-designed automated password generators. The standard includes not only a description of the approach but also a complete listing of the C source code of the algorithm. The algorithm generates words by forming pronounceable syllables and concatenating them to form a word. A random number generator produces a random stream of characters used to construct the syllables and words.

- **Reactive password checking**

A reactive password checking strategy is one in which the system periodically runs its own password cracker to find guessable passwords. The system cancels any passwords that are guessed and notifies the user. This tactic has a number of drawbacks. First it is resource intensive, if the job is done right. Because a determined opponent who is able to steal a password file can devote full CPU time to the task for hours or even days an effective reactive password checker is at a distinct disadvantage. Furthermore, any existing passwords remain vulnerable until the reactive password checker finds them.

• Complex password policy or Proactive password checker

The most promising approach to improved password security is a proactive password checker. In this scheme, a user is allowed to select his or her own password. However, at the time of selection, the system checks to see if the password is allowable and if not, rejects it. Such checkers are based on the philosophy that with sufficient guidance from the system, users can select memorable passwords from a fairly large password space that are not likely to be guessed in a dictionary attack. The trick with a proactive password checker is to strike a balance between user acceptability and strength. If the system rejects too many passwords, users will complain that it is too hard to select a password. If the system uses some simple algorithm to define what is acceptable, this provides guidance to password crackers to refine their guessing technique. In the remainder of this subsection, we look at possible approaches to proactive password checking.

Token-Based Authentication:

Objects that a user possesses for the purpose of user authentication are called tokens. we examine two types of tokens that are widely used; these are cards that have the appearance and size of bank cards.

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

Memory Cards

Memory cards can store data 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 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 PIN. Among the potential drawbacks are the following.

- **Requires special reader**: This increases the cost of using the token and creates the requirement to maintain the security of the reader's hardware and software.
- **Token loss:** A lost token temporarily prevents its owner from gaining system access. Thus there is an administrative cost in replacing the lost token. In addition, if the token is found, stolen, or forged, then an adversary now need only determine the PIN to gain unauthorized access.
- **User dissatisfaction**: Although users may have no difficulty in accepting the use of a memory card for ATM access, its use for computer access may be deemed inconvenient.

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 and 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 that 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.

A smart card contains within it an entire microprocessor, including processor, memory, and I/O ports. Some versions incorporate a special co-processing circuit for cryptographic operation to speed the task of encoding and decoding messages or generating digital signatures to validate the information transferred.

A typical smart card includes three types of memory. Read-only memory (ROM) stores data that does not change during the card's life, such as the card number and the cardholder's name. Electrically erasable programmable ROM (EEPROM) holds application data and programs, such as the protocols that the card can execute. It also holds data that may vary with time. For example, in a
Telephone card, the EEPROM holds the talk time remaining. Random access memory (RAM) holds temporary data generated when applications are executed.

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.

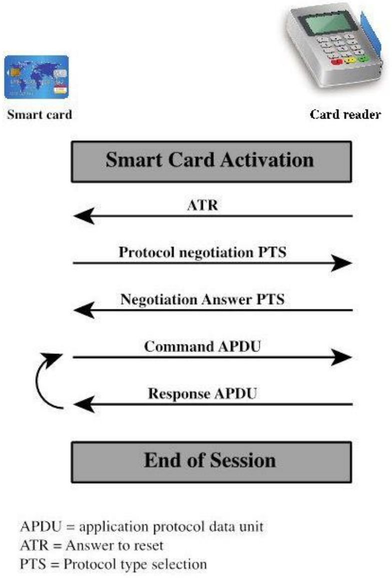

Figure 3.5 Smart Card/Reader Exchange

One of the most recent and most advanced eID deployments is the German eID card neuer Personalausweis . The card has human-readable data printed on its surface, including the following:

Personal data: Such as name, date of birth, and address; this is the type of printed information found on passports and driver's licenses.

Document number: An alphanumerical nine-character unique identifier of each card.

Card access number (CAN): A six-digit decimal random number printed on the face of the card. This is used as a password, as explained subsequently.

Machine readable zone (MRZ): Three lines of human- and machine-readable text on the back of the card. This may also be used as a password.

Eid Functions:

The card has the following three separate electronic functions, each with its own protected dataset.

- **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.

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.

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.

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.

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.

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.

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.

Password Protocol:

It provides a simple example of a challenge-response protocol for authentication via password. Actual protocols are more complex, 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.

Token Protocol:

It 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:

This 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:

This 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.

Security Issues for User Authentication:

As with any security service, user authentication, particularly remote user authentication, is subject to a variety of attacks. Summarizes the principal attacks on user authentication, broken down by type of authenticator. Much of the table is self-explanatory.

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. 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. 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.

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.

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. 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.

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

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.

Case Study: Security Problems for ATM Systems:
This vulnerability provides a useful case study illustrating those cryptographic functions and services alone do not guarantee security; they must be properly implemented as part of a system.

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.

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.

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:

- **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.