Example of Cyber attacks

Spanning 2015 and 2016, the SWIFT global messaging system which is used by banks to move money around the world, was used by hackers from North Korea. These successful attacks results in millions of dollars being stolen. The attack was linked back to a group called Lazarus who had links to North Korea. If the blame is rightly on North Korea, this instance would mark the first example of a cyberattack from a nation-state that targeted funds; still a very damaging attack.

Attackers were able to find vulnerabilities in the defences of banks and use them to access their systems and ultimately gain access to their legitimate SWIFT credentials.

The Fast Facts: Capital One determined that a hacker broke into a server by exploiting a configuration vulnerability in a web application firewall on March 22 and 23, 2019. The person accessed personal information for more than 100 million Capital One customers in the U.S. and 6 million in Canada. The outcome makes this hack one of the biggest ever. Then, according to the criminal complaint, the person tried to share the stolen information with other people online.

Dunkin' Donuts first reported a credential stuffing attack at the end of November 2018, and has notified users of more account breaches following a 2019 attack.

The Oregon DHS notified about 645,000 clients that their personal data was potentially breached during a spear-phishing attack. Nine employees fell for the email campaign providing their user credentials, giving hackers full access to more than 2 million emails.

In May 2019, a surveillance contractor for US Customs and Border Protection suffered a breach, and hackers stole photos of travelers and license plates related to about 100,000 people. The Tennessee-based contractor, a longtime CBP affiliate known as Perceptics, also lost detailed information about its surveillance hardware and how CBP implements it at multiple US ports of entry. The Perceptics breach was first reported by *The Register*, and CBP officials later disclosed the incident to *The Washington Post*. Though CBP was hesitant at first to admit that Perceptics was the contractor that had suffered the breach, the agency sent a Microsoft Word document to the *Post* titled "CBP Perceptics Public Statement" in its initial response. Days later, hackers posted the stolen Perceptics data to the dark web. On Tuesday, CBP suspended Perceptics from federal contracting, though it did not say why.

In March 2019, following a research report from the threat intelligence firm Kaspersky, computer maker Asus disclosed a supply chain attack sometime in the second half of 2018 that had compromised the company's Live Update tool to push malware to almost 1 million customers. Victim devices accepted the tainted software because the attackers signed it with a real Asus certificate

The combination of space, time, and strength that must be considered as the basic elements of this theory of defense makes this a fairly complicated matter. Consequently, it is not easy to find a fixed point of departure.

—*On War*, Carl Von Clausewitz

The art of war teaches us to rely not on the likelihood of the enemy's not coming, but on our own readiness to receive him; not on the chance of his not attacking, but rather on the fact that we have made our position unassailable.

—*The Art of War*, Sun Tzu

The requirements of **information security** within an organization have undergone two major changes in the last several decades. Before the widespread use of data processing equipment, the security of information felt to be valuable to an organization was provided primarily by physical and administrative means. An example of the former is the use of rugged filing cabinets with a combination lock for storing sensitive documents. An example of the latter is personnel screening procedures used during the hiring process.

With the introduction of the computer, the need for automated tools for protecting files and other information stored on the computer became evident. This is especially the case for a shared system, such as a time-sharing system, and the need is even more acute for systems that can be accessed over a public telephone network, data network, or the Internet. The generic name for the collection of tools designed to protect data and to thwart hackers is **computer security**.

The second major change that affected security is the introduction of distributed systems and the use of networks and communications facilities for carrying data between terminal user and computer and between computer and computer. Network security measures are needed to protect data during their transmission. In fact, the term **network security** is somewhat misleading, because virtually all business, government, and academic organizations interconnect their data processing equipment with a collection of interconnected networks. Such a collection is often referred to as an internet,[1] and the term **internet security** is used.

There are no clear boundaries between these two forms of security. For example, one of the most publicized types of attack on information systems is the computer virus. A virus may be introduced into a system physically when it arrives on a diskette and is subsequently loaded onto a computer. Viruses may also arrive over an internet. In either case, once the virus is resident on a computer system, internal computer security tools are needed to detect and recover from the virus.

---

[1]We use the term *internet*, with a lowercase "i," to refer to any interconnected collection of network. A corporate intranet is an example of an internet. The Internet with a capital "I" may be one of the facilities used by an organization to construct its internet.

This book focuses on internet security, which consists of measures to deter, prevent, detect, and correct security violations that involve the transmission of information. That is a broad statement that covers a host of possibilities. To give you a feel for the areas covered in this book, consider the following examples of security violations:

1. User A transmits a file to user B. The file contains sensitive information (e.g., payroll records) that is to be protected from disclosure. User C, who is not authorized to read the file, is able to monitor the transmission and capture a copy of the file during its transmission.
2. A network manager, D, transmits a message to a computer, E, under its management. The message instructs computer E to update an authorization file to include the identities of a number of new users who are to be given access to that computer. User F intercepts the message, alters its contents to add or delete entries, and then forwards the message to E, which accepts the message as coming from manager D and updates its authorization file accordingly.
3. Rather than intercept a message, user F constructs its own message with the desired entries and transmits that message to E as if it had come from manager D. Computer E accepts the message as coming from manager D and updates its authorization file accordingly.
4. An employee is fired without warning. The personnel manager sends a message to a server system to invalidate the employee's account. When the invalidation is accomplished, the server is to post a notice to the employee's file as confirmation of the action. The employee is able to intercept the message and delay it long enough to make a final access to the server to retrieve sensitive information. The message is then forwarded, the action taken, and the confirmation posted. The employee's action may go unnoticed for some considerable time.
5. A message is sent from a customer to a stockbroker with instructions for various transactions. Subsequently, the investments lose value and the customer denies sending the message.

Although this list by no means exhausts the possible types of security violations, it illustrates the range of concerns of network security.

Internetwork security is both fascinating and complex. Some of the reasons follow:

1. Security involving communications and networks is not as simple as it might first appear to the novice. The requirements seem to be straightforward; indeed, most of the major requirements for security services can be given self-explanatory one-word labels: confidentiality, authentication, nonrepudiation, integrity. But the mechanisms used to meet those requirements can be quite complex, and understanding them may involve rather subtle reasoning.
2. In developing a particular security mechanism or algorithm, one must always consider potential attacks on those security features. In many cases, successful attacks are designed by looking at the problem in a completely different way, therefore exploiting an unexpected weakness in the mechanism.

3. Because of point 2, the procedures used to provide particular services are often counterintuitive: It is not obvious from the statement of a particular requirement that such elaborate measures are needed. It is only when the various countermeasures are considered that the measures used make sense.
4. Having designed various security mechanisms, it is necessary to decide where to use them. This is true both in terms of physical placement (e.g., at what points in a network are certain security mechanisms needed) and in a logical sense [e.g., at what layer or layers of an architecture such as TCP/IP (Transmission Control Protocol/Internet Protocol) should mechanisms be placed].
5. Security mechanisms usually involve more than a particular algorithm or protocol. They usually also require that participants be in possession of some secret information (e.g., an encryption key), which raises questions about the creation, distribution, and protection of that secret information. There is also a reliance on communications protocols whose behavior may complicate the task of developing the security mechanism. For example, if the proper functioning of the security mechanism requires setting time limits on the transit time of a message from sender to receiver, then any protocol or network that introduces variable, unpredictable delays may render such time limits meaningless.

Thus, there is much to consider. This chapter provides a general overview of the subject matter that structures the material in the remainder of the book. We begin with a general discussion of network security services and mechanisms, and of the types of attacks they are designed for. Then we develop a general overall model within which the security services and mechanisms can be viewed.

1.1 SERVICES, MECHANISMS, AND ATTACKS

To assess the security needs of an organization effectively and to evaluate and choose various security products and policies, the manager responsible for security needs some systematic way of defining the requirements for security and characterizing the approaches to satisfying those requirements. One approach is to consider three aspects of information security:

- **Security attack:** Any action that compromises the security of information owned by an organization.
- **Security mechanism:** A mechanism that is designed to detect, prevent, or recover from a security attack.
- **Security service:** A service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.

Services

Let us consider these topics briefly, in reverse order. We can think of information security services as replicating the types of functions normally associated with phys-

ical documents. Much of the activity of humankind, in areas as diverse as commerce, foreign policy, military action, and personal interactions, depends on the use of documents and on both parties to a transaction having confidence in the integrity of those documents. Documents typically have signatures and dates; they may need to be protected from disclosure, tampering, or destruction; they may be notarized or witnessed; may be recorded or licensed, and so on.

As information systems become ever more pervasive and essential to the conduct of our affairs, electronic information takes on many of the roles traditionally performed by paper documents. Accordingly, the types of functions traditionally associated with paper documents must be performed on documents that exist in electronic form. Several aspects of electronic documents make the provision of such functions or services challenging:

1. It is usually possible to discriminate between an original paper document and a xerographic copy. However, an electronic document is merely a sequence of bits; there is no difference whatsoever between the "original" and any number of copies.
2. An alteration to a paper document may leave some sort of physical evidence of the alteration. For example, an erasure can result in a thin spot or a roughness in the surface. Altering bits in a computer memory or in a signal leaves no physical trace.
3. Any "proof" process associated with a physical document typically depends on the physical characteristics of that document (e.g., the shape of a handwritten signature or an embossed notary seal). Any such proof of authenticity of an electronic document must be based on internal evidence present in the information itself.

Table 1.1 lists some of the common functions traditionally associated with documents and for which analogous function for electronic documents and messages are required. We can think of these functions as requirements to be met by a security facility.

The list of Table 1.1 is lengthy and is not by itself a useful guide to organizing a security facility. Computer and network security research and development have instead focused on a few general security services that encompass the various functions required of an information security facility. We explore those in the next section.

Table 1.1 A Partial List of Common Information Integrity Functions [SIMM92]

- Identification
- Authorization
- License and/or certification
- Signature
- Witnessing (notarization)
- Concurrence
- Liability
- Receipts
- Certification of origination and/or receipt
- Endorsement
- Access (egress)
- Validation
- Time of occurrence
- Authenticity—software and/or files
- Vote
- Ownership
- Registration
- Approval/disapproval
- Privacy (secrecy)

Mechanisms

There is no single mechanism that will support all the functions listed in Table 1.1. As this book proceeds, we will see a variety of mechanisms that come into play. However, we can note at this point that there is one particular element that underlies many of the security mechanisms in use: cryptographic techniques. Encryption or encryption-like transformations of information (such as hash functions) are the most common mechanisms for providing security. Thus, this book focuses on the development, use, and management of such techniques.

Attacks

As G. J. Simmons points out, information security is about how to prevent attacks or, failing that, to detect attacks on information-based systems wherein the information itself has no meaningful physical existence and then to recover from the attacks [SIMM92].

Table 1.2 lists examples of attacks, each of which has arisen in a number of real-world cases. These are examples of specific attacks that an organization or an individual (or an organization on behalf of its employees) may need to counter. The nature of the attack that concerns an organization varies greatly from one set

Table 1.2 Examples of Security Attacks [SIMM92]

1. Gain unauthorized access to information (i.e., violate secrecy or privacy).
2. Impersonate another user either to shift responsibility (i.e., liability) or else to use the other's license for the purpose of
   a. originating fraudulent information,
   b. modifying legitimate information,
   c. using fraudulent identity to gain unauthorized access,
   d. fraudulently authorizing transactions or endorsing them.
3. Disavow responsibility or liability for information the cheater did originate.
4. Claim to have received from some other user information that the cheater created (i.e., fraudulent attribution of responsibility or liability).
5. Claim to have sent to a receiver (at a specified time) information that was not sent (or was sent at a different time).
6. Either disavow receipt of information that was in fact received, or claim a false time of receipt.
7. Enlarge cheater's legitimate license (for access, origination, distribution, etc.).
8. Modify (without authority to do so) the license of others (fraudulently enroll others, restrict or enlarge existing licenses, etc.).
9. Conceal the presence of some information (a covert communication) in other information (the overt communication).
10. Insert self into a communications link between other users as an active (undetected) relay point.
11. Learn who accesses which information (sources, files, etc.) and when the accesses are made even if the information itself remains concealed (e.g., a generalization of traffic analysis from communications channels to data bases, software, etc.).
12. Impeach an information integrity protocol by revealing information the cheater is supposed to (by the terms of the protocol) keep secret.
13. Pervert the function of software, typically by adding a covert function.
14. Cause others to violate a protocol by means of introducing incorrect information.
15. Undermine confidence in a protocol by causing apparent failures in the system.
16. Prevent communication among other users, in particular, surreptitious interference to cause authentic communication to be rejected as unauthentic.

Table 1.3 Threats and Attacks (RFC 2828)

Threat	A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.
Attack	An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

of circumstances to another. Fortunately, we can approach the problem from a different angle by looking at the generic types of attack that might be encountered. That is the subject of the next section.

You should note that, in the literature, the terms *threat* and *attack* are commonly used to mean more or less the same thing. Table 1.3 provides definitions taken from RFC 2828, *Internet Security Glossary*.

1.2 THE OSI SECURITY ARCHITECTURE

To assess effectively the security needs of an organization, and to evaluate and choose various security products and policies, the manager responsible for security needs some systematic way of defining the requirements for security and characterizing the approaches to satisfying those requirements. This is difficult enough in a centralized data processing environment; with the use of local area and wide area networks, the problems are compounded.

ITU-T[2] Recommendation X.800, *Security Architecture for OSI*, defines such a systematic approach. The OSI security architecture is useful to managers as a way of organizing the task of providing security. Furthermore, because this architecture was developed as an international standard, computer and communications vendors have developed security features for their products and services that relate to this structured definition of services and mechanisms.

For our purposes, the OSI security architecture provides a useful, if abstract, overview of many of the concepts that this book deals with. The OSI security architecture focuses on security services, mechanisms, and attacks.

Security Services

X.800 defines a security service as a service provided by a protocol layer of communicating open systems, which ensures adequate security of the systems or of data

[2]The International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T) is a United Nations (UN)-sponsored agency that develops standards, called Recommendations, relating to telecommunications and to open systems interconnection (OSI).

transfers. Perhaps a clearer definition is found in RFC 2828, which provides the following definition: a processing or communication service that is provided by a system to give a specific kind of protection to system resources; security services implement security policies, and are implemented by security mechanisms.

X800 divides these services into five categories and fourteen specific services (Table 1.4). We look at each category in turn.[3]

Authentication

The authentication service is concerned with assuring that a communication is authentic. In the case of a single message, such as a warning or alarm signal, the function of the authentication service is to assure the recipient that the message is from the source that it claims to be from. In the case of an ongoing interaction, such as the connection of a terminal to a host, two aspects are involved. First, at the time of connection initiation, the service assures that the two entities are authentic, that is, that each is the entity that it claims to be. Second, the service must assure that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties for the purposes of unauthorized transmission or reception.

Two specific authentication services are defined in the standard:

- **Peer entity authentication:** Provides for the corroboration of the identity of a peer entity in an association. It is provided for use at the establishment of, or at times during the data transfer phase, of a connection. It attempts to provide confidence that an entity is not attempting either a masquerade or an unauthorized replay of a previous connection.
- **Data origin authentication:** Provides for the corroboration of the source of a data unit. It does not provide protection against the duplication or modification of data units. This type of service supports applications like electronic mail where there are no prior interactions between the communicating entities.

Access Control

In the context of network security, access control is the ability to limit and control the access to host systems and applications via communications links. To achieve this, each entity trying to gain access must first be identified, or authenticated, so that access rights can be tailored to the individual.

Data Confidentiality

Confidentiality is the protection of transmitted data from passive attacks (defined subsequently). With respect to the content of a data transmission, several levels of protection can be identified. The broadest service protects all user data transmitted between two users over a period of time. For example, if a TCP con-

---

[3]There is no universal agreement about many of the terms used in the security literature. For example, the term *integrity* is sometimes used to refer to all aspects of information security. The term *authentication* is sometimes used to refer both to verification of identity and to the various functions listed under integrity in the following list. Our usage here agrees with both X.800 and RFC 2828.

Table 1.4 Security Services (X.800)

AUTHENTICATION

The assurance that the communicating entity is the one that it claims to be.

Peer Entity Authentication
Used in association with a logical connection to provide confidence in the identity of the entities connected.

Data-Origin Authentication
In a connectionless transfer, provides assurance that the source of received data is as claimed.

ACCESS CONTROL

The prevention of unauthorized use of a resource (i.e., this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do).

DATA CONFIDENTIALITY

The protection of data from unauthorized disclosure.

Connection Confidentiality
The protection of all user data on a connection.

Connectionless Confidentiality
The protection of all user data in a single data block

Selective-Field Confidentiality
The confidentiality of selected fields within the user data on a connection or in a single data block.

Traffic-flow Confidentiality
The protection of the information that might be derived from observation of traffic flows.

DATA INTEGRITY

The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).

Connection Integrity with Recovery
Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.

Connection Integrity without Recovery
As above, but provides only detection without recovery.

Selective-Field Connection Integrity
Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed.

Connectionless Integrity
Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.

Selective-Field Connectionless Integrity
Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.

NONREPUDIATION

Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.

Nonrepudiation, Origin
Proof that the message was sent by the specified party.

Nonrepudiation, Destination
Proof that the message was received by the specified party.

nection is set up between two systems, this broad protection would prevent the release of any user data transmitted over the TCP connection. Narrower forms of this service can also be defined, including the protection of a single message or even specific fields within a message. These refinements are less useful than the broad approach and may even be more complex and expensive to implement.

The other aspect of confidentiality is the protection of traffic flow from analysis. This requires that an attacker not be able to observe the source and destination, frequency, length, or other characteristics of the traffic on a communications facility.

Data Integrity

As with confidentiality, integrity can apply to a stream of messages, a single message, or selected fields within a message. Again, the most useful and straightforward approach is total stream protection.

A connection-oriented integrity service, one that deals with a stream of messages, assures that messages are received as sent, with no duplication, insertion, modification, reordering, or replays. The destruction of data is also covered under this service. Thus, the connection-oriented integrity service addresses both message stream modification and denial of service. On the other hand, a connectionless integrity service, one that deals with individual messages only without regard to any larger context, generally provides protection against message modification only.

We can make a distinction between the service with and without recovery. Because the integrity service relates to active attacks, we are concerned with detection rather than prevention. If a violation of integrity is detected, then the service may simply report this violation, and some other portion of software or human intervention is required to recover from the violation. Alternatively, there are mechanisms available to recover from the loss of integrity of data, as we will review subsequently. The incorporation of automated recovery mechanisms is, in general, the more attractive alternative.

Nonrepudiation

Nonrepudiation prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the message was in fact sent by the alleged sender. Similarly, when a message is received, the sender can prove that the message was in fact received by the alleged receiver.

Availability Service

Both X.800 and RFC 2828 define availability to be the property of a system or a system resource being accessible and usable upon demand by an authorized system entity, according to performance specifications for the system (i.e., a system is available if it provides services according to the system design whenever users request them). A variety of attacks can result in the loss of or reduction in availability. Some of these attacks are amenable to automated countermeasures, such as authentication and encryption, whereas others require some sort of physical action to prevent or recover from loss of availability of elements of a distributed system.

X.800 treats availability as a property to be associated with various security services. However, it makes sense to call out specifically an availability service. An

availability service is one that protects a system to ensure its availability. This service addresses the security concerns raised by denial-of-service attacks. It depends on proper management and control of system resources and thus depends on access control service and other security services.

Security Mechanisms

Table 1.5 lists the security mechanisms defined in X.800. As can be seen the mechanisms are divided into those that are implemented in a specific protocol layer and those that are not specific to any particular protocol layer or security service. These mechanisms will be covered in the appropriate places in the book and so we do not elaborate now, except to comment on the definition of encipherment. X.800 distinguishes between reversible enchiperment mechanisms and irreversible enchiperment mechanisms. A reversible encipherment mechanism is simply an encryption algorithm that allows data to be encrypted and subsequently decrypted. Irreversible encihperment mechanisms include hash algorithms and message authentication codes, which are used in digital signature and message authentication applications.

Table 1.6, based on one in X.800, indicates the relationship between security services and security mechanisms.

Security Attacks

A useful means of classifying security attacks, used both in X.800 and RFC 2828, is in terms of *passive attacks* and *active attacks*. A passive attack attempts to learn or make use of information from the system but does not affect system resources. An active attack attempts to alter system resources or affect their operation.

Passive Attacks

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of passive attacks are release of message contents and traffic analysis.

The **release of message contents** is easily understood. A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. We would like to prevent the opponent from learning the contents of these transmissions.

A second type of passive attack, **traffic analysis**, is subtler. Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. The common technique for masking contents is encryption. If we had encryption protection in place, an opponent might still be able to observe the pattern of these messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

Passive attacks are very difficult to detect because they do not involve any alteration of the data. However, it is feasible to prevent the success of these attacks, usually by means of encryption. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

Table 1.5 Security Mechanisms (X.800)

SPECIFIC SECURITY MECHANISMS	PERVASIVE SECURITY MECHANISMS
May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.	Mechanisms that are not specific to any particular OSI security service or protocol layer.
Encipherment The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.	**Trusted Functionality** That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).
Digital Signature Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient).	**Security Label** The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.
Access Control A variety of mechanisms that enforce access rights to resources.	**Event Detection** Detection of security-relevant events.
Data Integrity A variety of mechanisms used to assure the integrity of a data unit or stream of data units.	**Security Audit Trail** Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.
Authentication Exchange A mechanism intended to ensure the identity of an entity by means of information exchange.	**Security Recovery** Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.
Traffic Padding The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.	
Routing Control Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.	
Notarization The use of a trusted third party to assure certain properties of a data exchange.	

Active Attacks

Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service.

Table 1.6 *Relationship between Security Services and Mechanisms*

Service	Mechanism							
	Enciph-erment	Digital signature	Access control	Data integrity	Authenti-cation exchange	Traffic padding	Routing control	Notari-zation
Peer entity authentication	Y	Y			Y			
Data origin authentication	Y	Y						
Access control			Y					
Confidentiality	Y						Y	
Traffic flow confidentiality	Y					Y	Y	
Data integrity	Y	Y		Y				
Nonrepudiation		Y		Y				Y
Availability				Y	Y			

A **masquerade** takes place when one entity pretends to be a different entity. A masquerade attack usually includes one of the other forms of active attack. For example, authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges.

Replay involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.

Modification of messages simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect. For example, a message meaning "Allow John Smith to read confidential file *accounts*" is modified to mean "Allow Fred Brown to read confidential file *accounts*."

The **denial of service** prevents or inhibits the normal use or management of communications facilities. This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service). Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.

Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect, measures are available to prevent their success. On the other hand, it is quite difficult to prevent active attacks absolutely, because to do so would require physical protection of all communications facilities and paths at all times. Instead, the goal is to detect them and to recover from any disruption or delays caused by them. Because the detection has a deterrent effect, it may also contribute to prevention.

1.3 A MODEL FOR NETWORK SECURITY

A model for much of what we will be discussing is captured, in very general terms, in Figure 1.1. A message is to be transferred from one party to another across some sort of internet. The two parties, who are the *principals* in this transaction, must cooperate for the exchange to take place. A logical information channel is established by defining a route through the internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) by the two principals.

Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on. All the techniques for providing security have two components:

- A security-related transformation on the information to be sent. Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent, and the addition of a code based on the contents of the message, which can be used to verify the identity of the sender

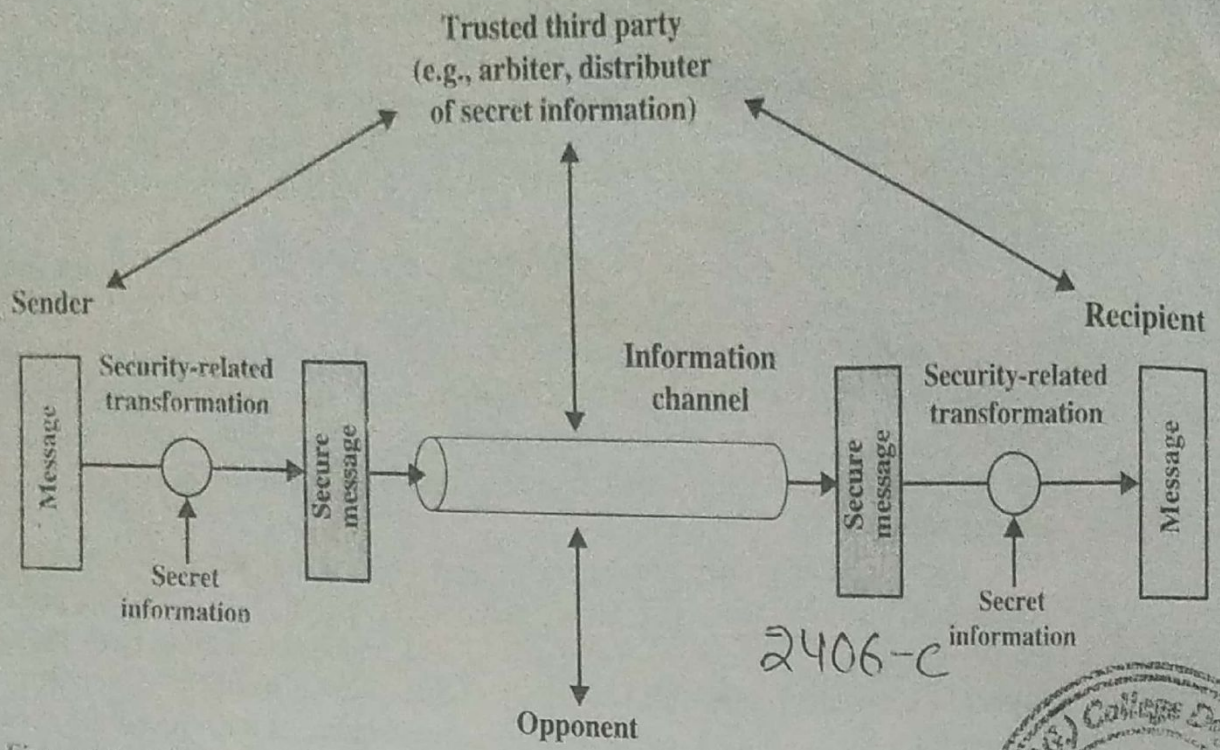


Figure 1.1 Model for Network Security

- Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception.[4]

A trusted third party may be needed to achieve secure transmission. For example, a third party may be responsible for distributing the secret information to the two principals while keeping it from any opponent. Or a third party may be needed to arbitrate disputes between the two principals concerning the authenticity of a message transmission.

This general model shows that there are four basic tasks in designing a particular security service:

1. Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose.
2. Generate the secret information to be used with the algorithm.
3. Develop methods for the distribution and sharing of the secret information.
4. Specify a protocol to be used by the two principals that makes use of the security algorithm and the secret information to achieve a particular security service.

Much of this book concentrates on the types of security mechanisms and services that fit into the model shown in Figure 1.1. However, there are other security-

[4]Part Two discusses a form of encryption, known as public-key encryption, in which only one of the two principals needs to have the secret information.

related situations of interest that do not neatly fit this model but that are considered in this book. A general model of these other situations is illustrated by Figure 1.2, which reflects a concern for protecting an information system from unwanted access. Most readers are familiar with the concerns caused by the existence of hackers, who attempt to penetrate systems that can be accessed over a network. The hacker can be someone who, with no malign intent, simply gets satisfaction from breaking and entering a computer system. Or, the intruder can be a disgruntled employee who wishes to do damage, or a criminal who seeks to exploit computer assets for financial gain (e.g., obtaining credit card numbers or performing illegal money transfers).

Another type of unwanted access is the placement in a computer system of logic that exploits vulnerabilities in the system and that can affect application programs as well as utility programs, such as editors and compilers. Two kinds of threats can be presented by programs:

- **Information access threats** intercept or modify data on behalf of users who should not have access to that data.
- **Service threats** exploit service flaws in computers to inhibit use by legitimate users.

Viruses and worms are two examples of software attacks. Such attacks can be introduced into a system by means of a disk that contains the unwanted logic concealed in otherwise useful software. They can also be inserted into a system across a network; this latter mechanism is of more concern in network security.

The security mechanisms needed to cope with unwanted access fall into two broad categories (see Figure 1.2). The first category might be termed a gatekeeper function. It includes password-based login procedures that are designed to deny access to all but authorized users and screening logic that is designed to detect and reject worms, viruses, and other similar attacks. Once access is gained, by either an unwanted user or unwanted software, the second line of defense consists of a variety of internal controls that monitor activity and analyze stored information in an attempt to detect the presence of unwanted intruders.

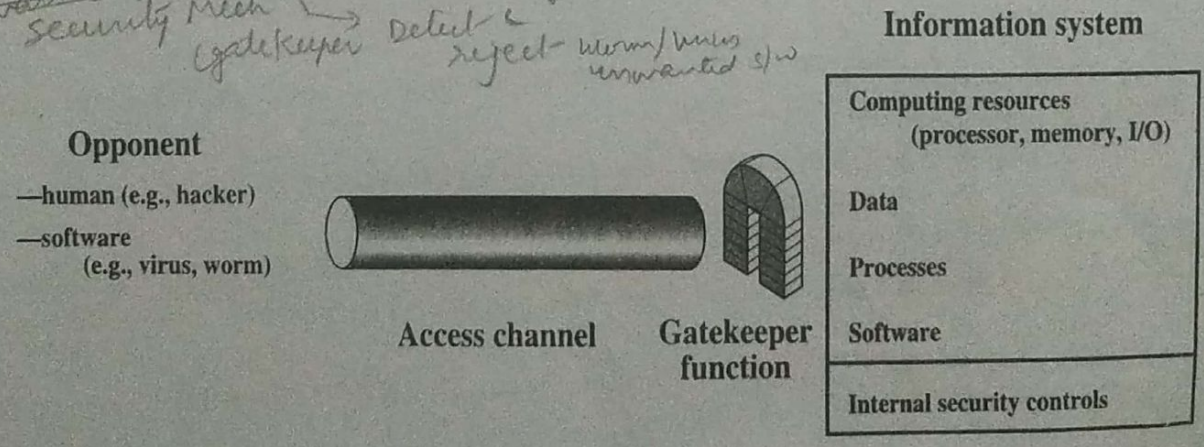


Figure 1.2 Network Access Security Model

Many savages at the present day regard their names as vital parts of themselves, and therefore take great pains to conceal their real names, lest these should give to evil-disposed persons a handle by which to injure their owners.

—The Golden Bough, **Sir James George Frazer**

Symmetric encryption, also referred to as conventional encryption or single-key encryption, was the only type of encryption in use prior to the development of public-key encryption.[1] It remains by far the most widely used of the two types of encryption. Part One examines a number of symmetric ciphers. In this chapter, we begin with a look at a general model for the symmetric encryption process; this will enable us to understand the context within which the algorithms are used. Next, we examine a variety of algorithms in use before the computer era. Finally, we look briefly at a different approach known as steganography. Chapter 3 examines the most widely used symmetric cipher: DES.

Before beginning, we define some terms. An original message is known as the **plaintext**, while the coded message is called the **ciphertext**. The process of converting from plaintext to ciphertext is known as **enciphering** or **encryption**; restoring the plaintext from the ciphertext is **deciphering** or **decryption**. The many schemes used for enciphering constitute the area of study known as **cryptography**. Such a scheme is known as a **cryptographic system** or a **cipher**. Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of **cryptanalysis**. Cryptanalysis is what the layperson calls "breaking the code." The areas of cryptography and cryptanalysis together are called **cryptology**.

2.1 SYMMETRIC CIPHER MODEL

A symmetric encryption scheme has five ingredients (Figure 2.1):

- **Plaintext:** This is the original intelligible message or data that is fed into the algorithm as input.
- **Encryption algorithm:** The encryption algorithm performs various substitutions and transformations on the plaintext.
- **Secret key:** The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.

[1] Public-key encryption was first described in the open literature in 1976; the National Security Agency (NSA) claims to have discovered it some years earlier.

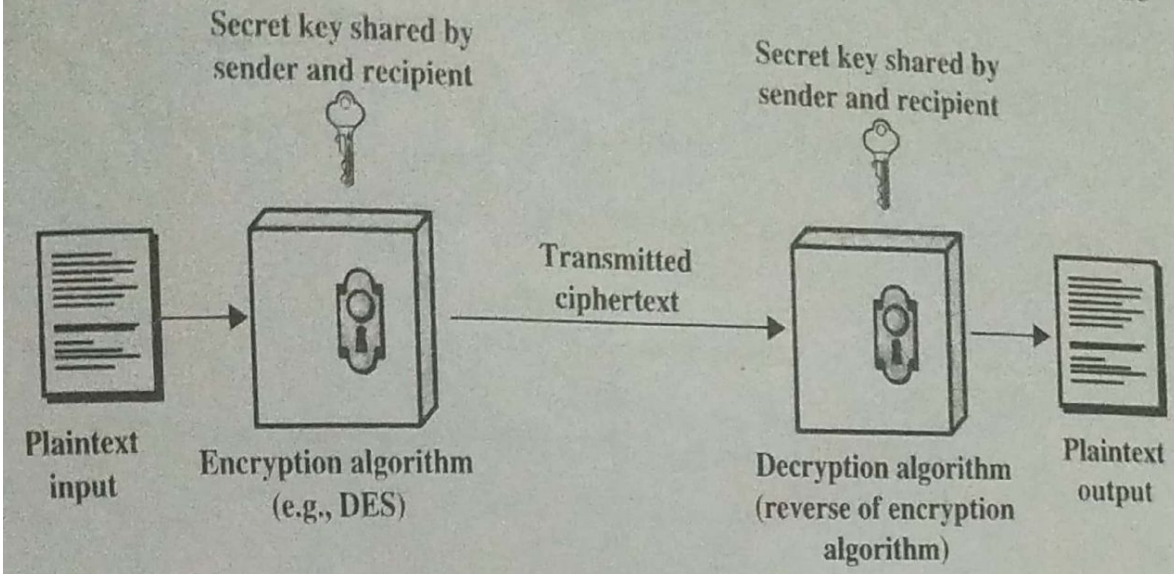


Figure 2.1 Simplified Model of Conventional Encryption

- **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.
- **Decryption algorithm:** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

There are two requirements for secure use of conventional encryption:

1. We need a strong encryption algorithm. At a minimum, we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key. This requirement is usually stated in a stronger form: The opponent should be unable to decrypt ciphertext or discover the key even if he or she is in possession of a number of ciphertexts together with the plaintext that produced each ciphertext.
2. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure. If someone can discover the key and knows the algorithm, all communication using this key is readable.

We assume that it is impractical to decrypt a message on the basis of the ciphertext *plus* knowledge of the encryption/decryption algorithm. In other words, we do not need to keep the algorithm secret; we need to keep only the key secret. This feature of symmetric encryption is what makes it feasible for widespread use. The fact that the algorithm need not be kept secret means that manufacturers can and have developed low-cost chip implementations of data encryption algorithms. These chips are widely available and incorporated into a number of products. With the use of symmetric encryption, the principal security problem is maintaining the secrecy of the key.

Let us take a closer look at the essential elements of a symmetric encryption scheme, using Figure 2.2. A source produces a message in plaintext, $X = [X_1, X_2, \ldots, X_M]$. The M elements of X are letters in some finite alphabet. Traditionally,

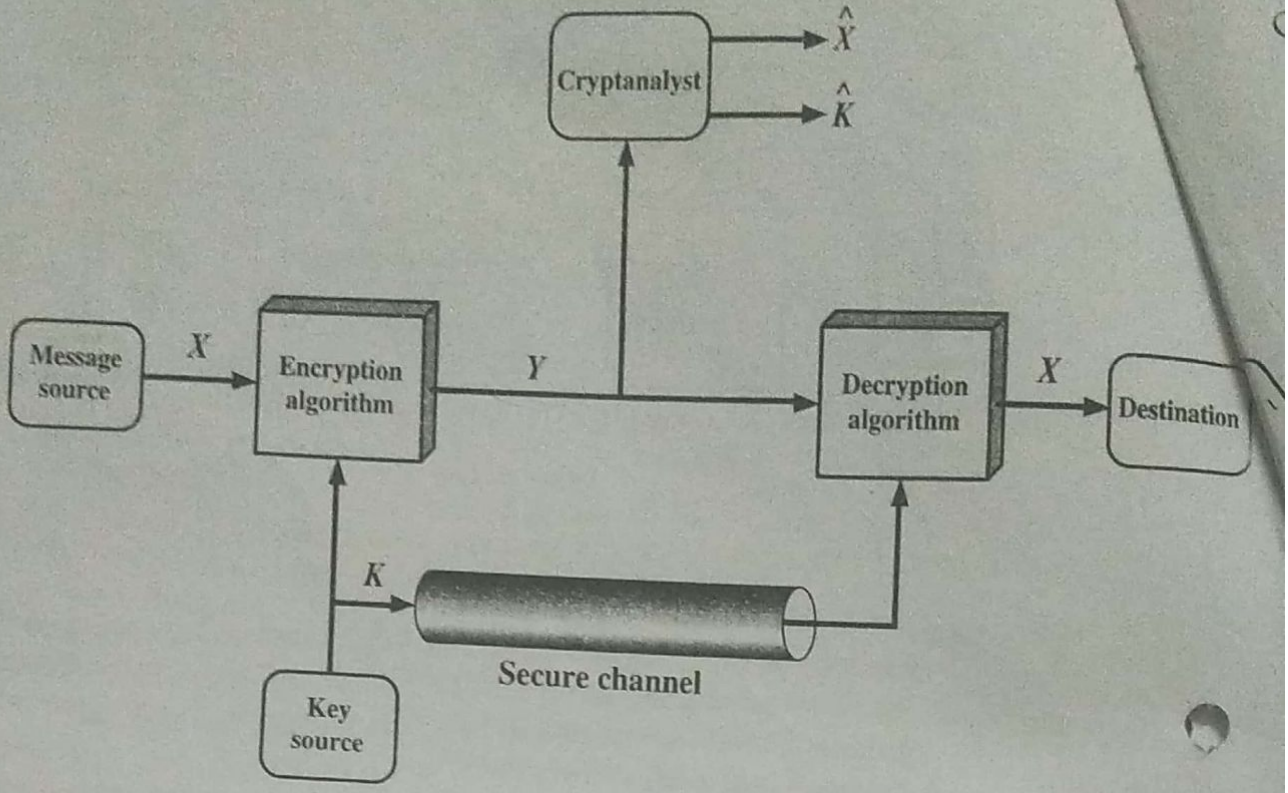


Figure 2.2 Model of Conventional Cryptosystem

the alphabet usually consisted of the 26 capital letters. Nowadays, the binary alphabet {0, 1} is typically used. For encryption, a key of the form $K = [K_1, K_2, \ldots, K_J]$ is generated. If the key is generated at the message source, then it must also be provided to the destination by means of some secure channel. Alternatively, a third party could generate the key and securely deliver it to both source and destination.

With the message X and the encryption key K as input, the encryption algorithm forms the ciphertext $Y = [Y_1, Y_2, \ldots, Y_N]$. We can write this as

$$Y = \mathrm{E}_K(X)$$

This notation indicates that Y is produced by using encryption algorithm E as a function of the plaintext X, with the specific function determined by the value of the key K.

The intended receiver, in possession of the key, is able to invert the transformation:

$$X = \mathrm{D}_K(Y)$$

An opponent, observing Y but not having access to K or X, may attempt to recover X or K or both X and K. It is assumed that the opponent knows the encryption (E) and decryption (D) algorithms. If the opponent is interested in only this particular message, then the focus of the effort is to recover X by generating a plaintext estimate $\hat{X}$. Often, however, the opponent is interested in being able to read future messages as well, in which case an attempt is made to recover K by generating an estimate $\hat{K}$.

Cryptography

Cryptographic systems are characterized along three independent dimensions:

1. **The type of operations used for transforming plaintext to ciphertext.** All encryption algorithms are based on two general principles: substitution, in which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element, and transposition, in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost (that is, that all operations are reversible). Most systems, referred to as product systems, involve multiple stages of substitutions and transpositions.
2. **The number of keys used.** If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption. If the sender and receiver each uses a different key, the system is referred to as asymmetric, two-key, or public-key encryption.
3. **The way in which the plaintext is processed.** A *block cipher* processes the input one block of elements at a time, producing an output block for each input block. A *stream cipher* processes the input elements continuously, producing output one element at a time, as it goes along.

Cryptanalysis

There are two general approaches to attacking a conventional encryption scheme:

- **Cryptanalysis:** Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext-ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used. If the attack succeeds in deducing the key, the effect is catastrophic: All future and past messages encrypted with that key are compromised.
- **Brute-force attack:** The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

We first consider cryptanalysis and then discuss brute-force attacks.

Table 2.1 summarizes the various types of cryptanalytic attacks, based on the amount of information known to the cryptanalyst. The most difficult problem is presented when all that is available is the *ciphertext only*. In some cases, not even the encryption algorithm is known, but in general we can assume that the opponent does know the algorithm used for encryption. One possible attack under these circumstances is the brute-force approach of trying all possible keys. If the key space is very large, this becomes impractical. Thus, the opponent must rely on an analysis of the ciphertext itself, generally applying various statistical tests to it. To use this approach, the opponent must have some general idea of the type of plaintext that is concealed, such as English or French text, a Windows EXE file, a Java source listing, an accounting file, and so on.

Table 2.1 Types of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst
Ciphertext only	• Encryption algorithm • Ciphertext to be decoded
Known plaintext	• Encryption algorithm • Ciphertext to be decoded • One or more plaintext-ciphertext pairs formed with the secret key
Chosen plaintext	• Encryption algorithm • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen ciphertext	• Encryption algorithm • Ciphertext to be decoded • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen text	• Encryption algorithm • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

The ciphertext-only attack is the easiest to defend against because the opponent has the least amount of information to work with. In many cases, however, the analyst has more information. The analyst may be able to capture one or more plaintext messages as well as their encryptions. Or the analyst may know that certain plaintext patterns will appear in a message. For example, a file that is encoded in the Postscript format always begins with the same pattern, or there may be a standardized header or banner to an electronic funds transfer message, and so on. All these are examples of *known plaintext*. With this knowledge, the analyst may be able to deduce the key on the basis of the way in which the known plaintext is transformed.

Closely related to the known-plaintext attack is what might be referred to as a probable-word attack. If the opponent is working with the encryption of some general prose message, he or she may have little knowledge of what is in the message. However, if the opponent is after some very specific information, then parts of the message may be known. For example, if an entire accounting file is being transmitted, the opponent may know the placement of certain key words in the header of the file. As another example, the source code for a program developed by Corporation X might include a copyright statement in some standardized position.

If the analyst is able somehow to get the source system to insert into the system a message chosen by the analyst, then a *chosen-plaintext* attack is possible. An example of this strategy is differential cryptanalysis, explored in Chapter 3. In general, if the analyst is able to choose the messages to encrypt, the analyst may deliberately pick patterns that can be expected to reveal the structure of the key.

algorithm that takes plaintext and an encryption key as input and produces a ciphertext.

- **Ciphertext**. It is the scrambled version of the plaintext produced by the encryption algorithm using a specific the encryption key. The ciphertext is not guarded. It flows on public channel. It can be intercepted or compromised by anyone who has access to the communication channel.

- **Decryption Algorithm**, It is a mathematical process, that produces a unique plaintext for any given ciphertext and decryption key. It is a cryptographic algorithm that takes a ciphertext and a decryption key as input, and outputs a plaintext. The decryption algorithm essentially reverses the encryption algorithm and is thus closely related to it.

- **Encryption Key**. It is a value that is known to the sender. The sender inputs the encryption key into the encryption algorithm along with the plaintext in order to compute the ciphertext.

- **Decryption Key**. It is a value that is known to the receiver. The decryption key is related to the encryption key, but is not always identical to it. The receiver inputs the decryption key into the decryption algorithm along with the ciphertext in order to compute the plaintext.

For a given cryptosystem, a collection of all possible decryption keys is called a **key space**.

An **interceptor** (an attacker) is an unauthorized entity who attempts to determine the plaintext. He can see the ciphertext and may know the decryption algorithm. He, however, must never know the decryption key.

Types of Cryptosystems

Fundamentally, there are two types of cryptosystems based on the manner in which encryption-decryption is carried out in the system:

- Symmetric Key Encryption
- Asymmetric Key Encryption

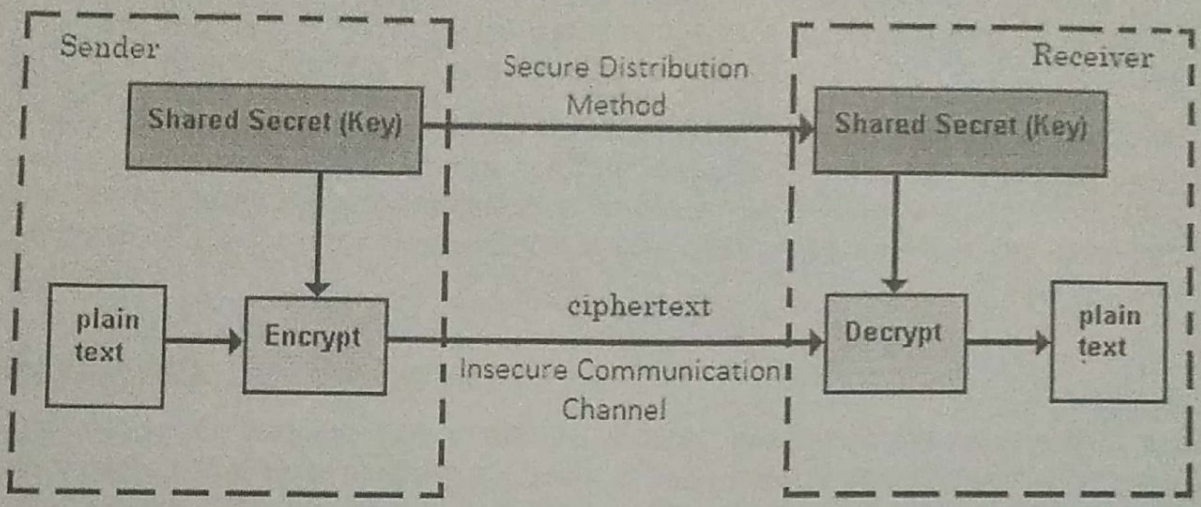
The main difference between these cryptosystems is the relationship between the encryption and the decryption key. Logically, in any cryptosystem, both the keys are closely associated. It is practically impossible to decrypt the ciphertext with the key that is unrelated to the encryption key.

Symmetric Key Encryption

The encryption process where **same keys are used for encrypting and decrypting** the information is known as Symmetric Key Encryption.

The study of symmetric cryptosystems is referred to as **symmetric cryptography**. Symmetric cryptosystems are also sometimes referred to as **secret key cryptosystems**.

A few well-known examples of symmetric key encryption methods are: Digital Encryption Standard (DES), Triple-DES (3DES), IDEA, and BLOWFISH.



Prior to 1970, all cryptosystems employed symmetric key encryption. Even today, its relevance is very high and it is being used extensively in many cryptosystems. It is very unlikely that this encryption will fade away, as it has certain advantages over asymmetric key encryption.

The salient features of cryptosystem based on symmetric key encryption are:

- Persons using symmetric key encryption must share a common key prior to exchange of information.
- Keys are recommended to be changed regularly to prevent any attack on the system.
- A robust mechanism needs to exist to exchange the key between the communicating parties. As keys are required to be changed regularly, this mechanism becomes expensive and cumbersome.
- In a group of **n** people, to enable two-party communication between any two persons, the number of keys required for group is **n × (n – 1)/2**.
- Length of Key (number of bits) in this encryption is smaller and hence, process of encryption-decryption is faster than asymmetric key encryption.
- Processing power of computer system required to run symmetric algorithm is less.

Challenge of Symmetric Key Cryptosystem

There are two restrictive challenges of employing symmetric key cryptography.

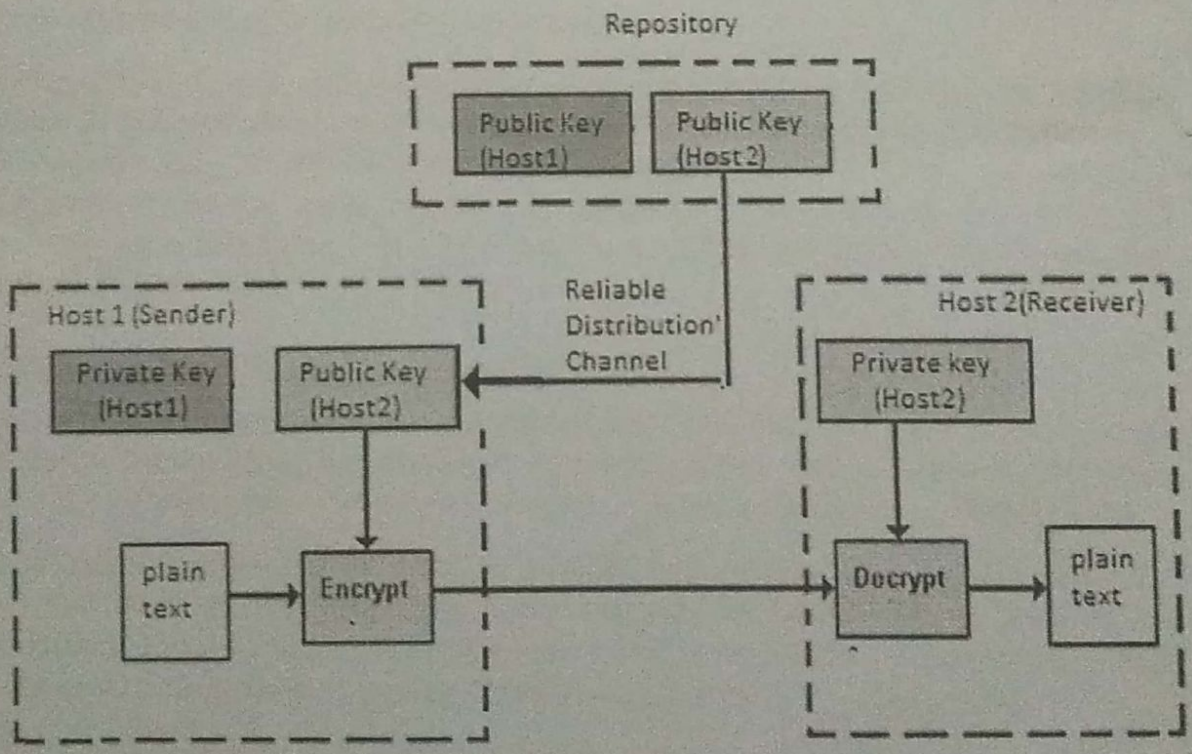
- **Key establishment** – Before any communication, both the sender and the receiver need to agree on a secret symmetric key. It requires a secure key establishment mechanism in place.

- **Trust Issue** – Since the sender and the receiver use the same symmetric key, there is an implicit requirement that the sender and the receiver 'trust' each other. For example, it may happen that the receiver has lost the key to an attacker and the sender is not informed.

These two challenges are highly restraining for modern day communication. Today, people need to exchange information with non-familiar and non-trusted parties. For example, a communication between online seller and customer. These limitations of symmetric key encryption gave rise to asymmetric key encryption schemes.

Asymmetric Key Encryption

The encryption process where **different keys are used for encrypting and decrypting the information** is known as Asymmetric Key Encryption. Though the keys are different, they are mathematically related and hence, retrieving the plaintext by decrypting ciphertext is feasible. The process is depicted in the following illustration:



Asymmetric Key Encryption was invented in the 20th century to come over the necessity of pre-shared secret key between communicating persons. The salient features of this encryption scheme are as follows:

- Every user in this system needs to have a pair of dissimilar keys, **private key** and **public key**. These keys are mathematically related – when one key is used for encryption, the other can decrypt the ciphertext back to the original plaintext.

- It requires to put the public key in public repository and the private key as a well-guarded secret. Hence, this scheme of encryption is also called **Public Key Encryption**.

- Though public and private keys of the user are related, it is computationally not feasible to find one from another. This is a strength of this scheme.

- When *Host1* needs to send data to *Host2*, he obtains the public key of *Host2* from repository, encrypts the data, and transmits.

- *Host2* uses his private key to extract the plaintext.

- Length of Keys (number of bits) in this encryption is large and hence, the process of encryption-decryption is slower than symmetric key encryption.

- Processing power of computer system required to run asymmetric algorithm is higher.

Symmetric cryptosystems are a natural concept. In contrast, public-key cryptosystems are quite difficult to comprehend.

You may think, *how can the encryption key and the decryption key are 'related', and yet it is impossible to determine the decryption key from the encryption key?* The answer lies in the mathematical concepts. It is possible to design a cryptosystem whose keys have this property. The concept of public-key cryptography is relatively new. There are fewer public-key algorithms known than symmetric algorithms.

Challenge of Public Key Cryptosystem

Public-key cryptosystems have one significant challenge: the user needs to trust that the public key that he is using in communications with a person really is the public key of that person and has not been spoofed by a malicious third party.

This is usually accomplished through a Public Key Infrastructure (PKI) consisting a trusted third party. The third party securely manages and attests to the authenticity of public keys. When the third party is requested to provide the public key for any communicating person X, they are trusted to provide the correct public key.

The third party satisfies itself about user identity by the process of attestation, notarization, or some other process - that X is the one and only, or globally unique, X. The most common method of making the verified public keys available is to embed them in a certificate which is digitally signed by the trusted third party.

Relation between Encryption Schemes

A summary of basic key properties of two types of cryptosystems is given below:

	Symmetric Cryptosystems	Public Key Cryptosystems
Relation between Keys	Same	Different, but mathematically related
Encryption Key	Symmetric	Public
Decryption Key	Symmetric	Private

Due to the advantages and disadvantage of both the systems, symmetric key and public-key cryptosystems are often used together in the practical information security systems.

Kerckhoff's Principle for Cryptosystem

In the 19th century, a Dutch cryptographer A. Kerckhoff furnished the requirements of a good cryptosystem. Kerckhoff stated that a cryptographic system should be secure even if everything about the system, except the key, is public knowledge. The six design principles defined by Kerckhoff for cryptosystem are:

- The cryptosystem should be unbreakable practically, if not mathematically.
- Falling of the cryptosystem in the hands of an intruder should not lead to any compromise of the system, preventing any inconvenience to the user.
- The key should be easily communicable, memorable, and changeable.
- The ciphertext should be transmissible by telegraph, an unsecure channel.
- The encryption apparatus and documents should be portable and operable by a single person.
- Finally, it is necessary that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe.

The second rule is currently known as **Kerckhoff principle**. It is applied in virtually all the contemporary encryption algorithms such as DES, AES, etc. These public algorithms are considered to be thoroughly secure. The security of the encrypted message depends solely on the security of the secret encryption key.

Keeping the algorithms secret may act as a significant barrier to cryptanalysis. However, keeping the algorithms secret is possible only when they are used in a strictly limited circle.

1.5.1 Attacks: A General View

From a common person's point of view, we can classify attacks into three categories, as shown in Fig. 1.9.

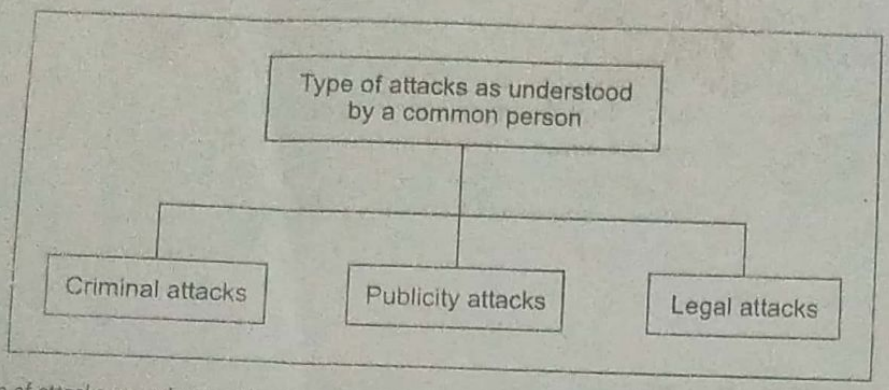


Fig. 1.9 Classification of attacks as understood in general terms

Let us now discuss these attacks.

1. Criminal Attacks

Criminal attacks are the simplest to understand. Here, the sole aim of the attackers is to maximize financial gain by attacking computer systems. Table 1.1 lists some forms of criminal attacks.

2. Publicity Attacks

Publicity attacks occur because the attackers want to see their names appear on television news channels and newspapers. History suggests that these types of attackers are usually not hardcore criminals. They are people such as students in universities or employees in large organizations, who seek publicity by adopting a novel approach of attacking computer systems.

One form of publicity attacks is to damage (or deface) the Web pages of a site by attacking it. One of the most famous of such attacks occurred on the *US Department of Justice*'s Web site in 1996. The *New York Times* home page was also infamously defaced two years later.

3. Legal Attacks

This form of attack is quite novel and unique. Here, the attacker tries to make the judge or the jury doubtful about the security of a computer system. This works as follows. The attacker attacks the computer system, and the attacked party (say a bank or an organization) manages to take the attacker to the court. While the case is being fought, the attacker tries to convince the judge and the jury that there is inherent weakness in the computer system and that she has done nothing wrongful. The aim of the attacker is to exploit the weakness of the judge and the jury in technological matters.

For example, an attacker may sue a bank for performing an online transaction, which he/she never wanted to perform. In court, the attacker could innocently say something like: *The bank's Web site forced me to enter a password and that is all that I provided; I do not know what happened thereafter.* The judge is unwittingly likely to sympathize with the attacker!

(2)

Table 1.1 Types of criminal attacks

Attack	Description
Fraud	Modern fraud attacks concentrate on manipulating some aspects of electronic currency, credit cards, electronic stock certificates, checks, letters of credit, purchase orders, ATMs, etc.
Scams	Scams come in various forms, some of the most common ones being sale of services, auctions, multilevel marketing schemes, general merchandise, and business opportunities, etc. People are enticed to send money in return of great returns, but end up losing their money. A very common example is the *Nigeria scam*, where an email from Nigeria (and other African countries) entices people to deposit money into a bank account with a promise of hefty gains. Whosoever gets caught in this scam loses money heavily.
Destruction	Some sort of grudge is the motive behind such attacks. For example, unhappy employees attack their own organization, whereas terrorists strike at much bigger levels. For example, in the year 2000, there was an attack against popular Internet sites such as Yahoo!, CNN, eBay, Buy.com, Amazon.com, and e*Trade where authorized users of these sites failed to log in or access these sites.
Identity theft	This is best understood with a quote from Bruce Schneier: *Why steal from someone when you can just become that person?* In other words, an attacker does not steal anything from a legitimate user—he/she becomes that legitimate user! For example, it is much easier to get the password of someone else's bank account, or to actually be able to get a credit card on someone else's name. Then that privilege can be misused until it gets detected.
Intellectual property theft	Intellectual property theft ranges from stealing companies' trade secrets, databases, digital music and videos, electronic documents and books, software, and so on.
Brand theft	It is quite easy to set up fake Web sites that look like real Web sites. How would a common user know if he/she is visiting the HDFC Bank site or an attacker's site? Innocent users end up providing their secrets and personal details on these fake sites to the attackers. The attackers use these details to then access the real site, causing an *identity theft*.

1.5.2 Attacks: A Technical View

From a technical point of view, we can classify the types of attacks on computers and network systems into two categories for better understanding: (a) Theoretical concepts behind these attacks, and (b) Practical approaches used by the attackers. Let us discuss these one by one.

1. Theoretical Concepts

As we discussed earlier, the principles of security face threat from various attacks. These attacks are generally classified into four categories, as mentioned earlier. These are the following:

Interception It has been discussed in the context of *confidentiality* earlier. It means that an unauthorized party has gained access to a resource. The party can be a person, program, or computer-based system. Examples of interception are copying of data or programs, and listening to network traffic.

Fabrication It has been discussed in the context of *authentication* earlier. This involves the creation of illegal objects on a computer system. For example, the attacker may add fake records to a database.

Modification It has been discussed in the context of *integrity* earlier. Here, the attacker may modify the values in a database.

Interruption It has been discussed in the context of *availability* earlier. Here, the resource becomes unavailable, lost, or unusable. Examples of interruption are causing problems to a hardware device, erasing program, data, or operating-system components.

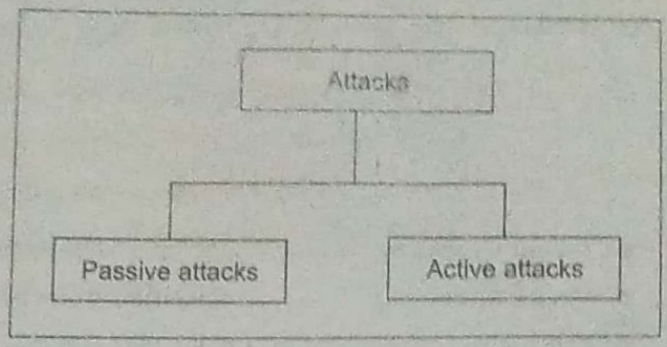


Fig. 1.10 Types of attacks

These attacks are further grouped into two types: **passive attacks** and **active attacks**, as shown in Fig. 1.10.

Let us discuss these two types of attacks now.

(a) Passive Attacks *Passive attacks* are those wherein the attacker indulges in eavesdropping or monitoring of data transmission. In other words, the attacker aims to obtain information that is in transit. The term *passive* indicates that the attacker does not attempt to perform any modifications to the data. In fact, this is also why passive attacks are harder to detect. Thus, the general approach to deal with passive attacks is to think about prevention, rather than detection or corrective actions.

Passive attacks do not involve any modifications to the contents of an original message.

Figure 1.11 shows further classification of passive attacks into two sub-categories. These categories are, namely **release of message contents** and **traffic analysis**.

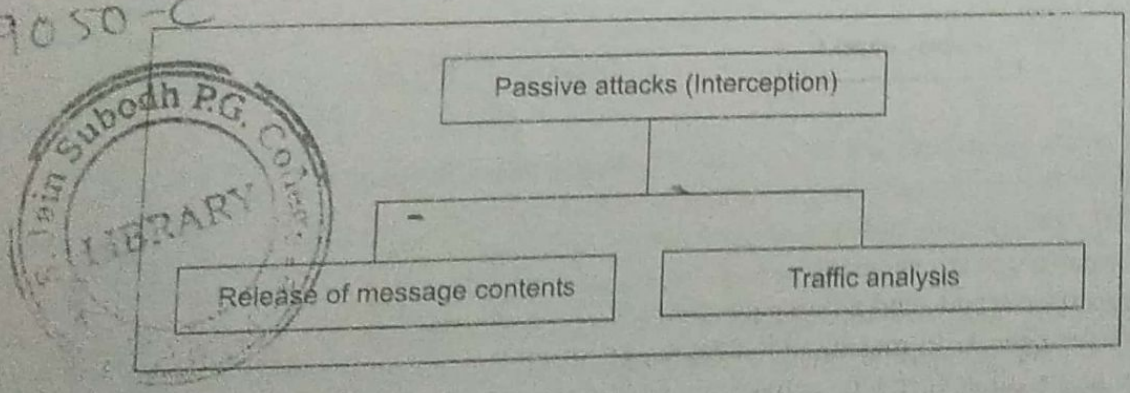


Fig. 1.11 Passive attacks

Release of message contents is quite simple to understand. When you send a confidential email message to your friend, you desire that only he/she be able to access it. Otherwise, the contents of the message are released against our wishes to someone else. Using certain security mechanisms, we can prevent the *release of message contents*. For example, we can encode messages using a code language, so that only the desired parties understand the contents of a message, because only they know the code language. However, if many such messages are passing through, a passive attacker could try to figure out similarities between them to come up with some sort of pattern that provides her some clues regarding the communication that is taking place. Such attempts of analyzing (encoded) messages to come up with likely patterns are the work of the *traffic-analysis* attack.

(b) Active Attacks Unlike *passive attacks*, the *active attacks* are based on the modification of the original message in some manner, or in the creation of a false message. These attacks cannot be prevented easily. However, they can be detected with some effort, and attempts can be made to recover from them. These attacks can be in the form of interruption, modification and fabrication.

In active attacks, the contents of the original message are modified in some way.

- Trying to pose as another entity involves **masquerade** attacks.
- Modification attacks can be classified further into **replay attacks** and **alteration of messages**.
- Fabrication causes **Denial Of Service (DOS)** attacks.

This classification is shown in Fig. 1.12.

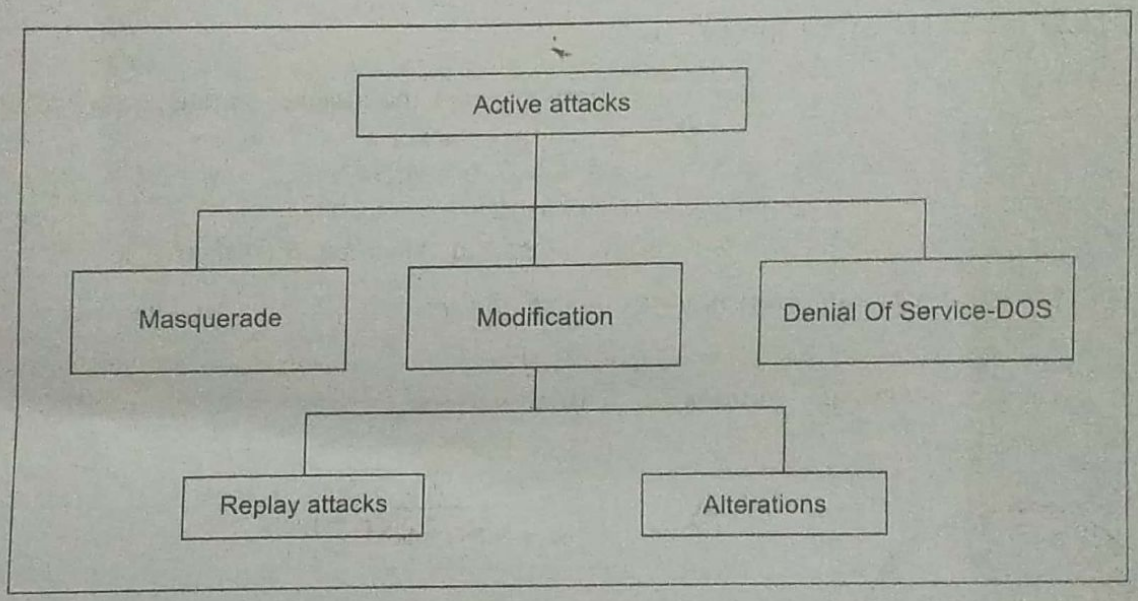


Fig. 1.12 Active attacks

Masquerade is caused when an unauthorized entity pretends to be another entity. As we have seen, user C might pose as user A and send a message to user B. User B might be led to believe that the message indeed came from user A. In masquerade attacks, an entity poses as another entity. In masquerade attacks, usually some other forms of active attacks are also embedded. As an instance, the attack may involve capturing the user's authentication sequence (e.g. user ID and password). Later, those details can be replayed to gain illegal access to the computer system.

In a *replay attack*, a user captures a sequence of events, or some data units, and re-sends them. For instance, suppose user A wants to transfer some amount to user C's bank account. Both users A and C have accounts with bank B. User A might send an electronic message to bank B, requesting for the funds transfer. User C could capture this message, and send a second copy of the same to bank B. Bank B would have no idea that this is an unauthorized message, and would treat this as a second, and *different*, funds transfer request from user A. Therefore, user C would get the benefit of the funds transfer twice: once authorized, once through a replay attack.

Alteration of messages involves some change to the original message. For instance, suppose user A sends an electronic message *Transfer $1000 to D's account* to bank B. User C might capture this, and change it to

Transfer $10000 to C's account. Note that both the beneficiary and the amount have been changed—instead, only one of these could have also caused alteration of the message.

Denial Of Service (DOS) attacks make an attempt to prevent legitimate users from accessing some services, which they are eligible for. For instance, an unauthorized user might send too many login requests to a server using random user ids in quick succession, so as to flood the network and deny other legitimate users to use the network facilities.

1.5.3 The Practical Side of Attacks

The attacks discussed earlier can come in a number of forms in real life. They can be classified into two broad categories: application-level attacks and network-level attacks, as shown in Fig. 1.13.

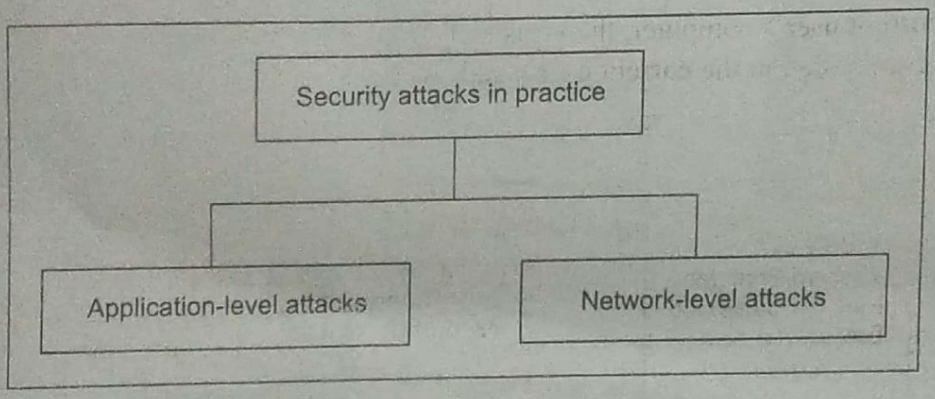


Fig. 1.13 Practical side of attacks

Let us discuss these now.

1. Application-level Attacks

These attacks happen at an application level in the sense that the attacker attempts to access, modify, or prevent access to information of a particular application, or the application itself. Examples of this are trying to obtain someone's credit-card information on the Internet, or changing the contents of a message to change the amount in a transaction, etc.

2. Network-level Attacks

These attacks generally aim at reducing the capabilities of a network by a number of possible means. These attacks generally make an attempt to either slow down, or completely bring to halt, a computer network. Note that this automatically can lead to application-level attacks, because once someone is able to gain access to a network, usually he/she is able to access/modify at least some sensitive information, causing havoc.

These two types of attacks can be attempted by using various mechanisms, as discussed next. We will not classify these attacks into the above two categories, since they can span across application as well as network levels.

Security attacks can happen at the application level or the network level.

1.5.4 Programs that Attack

Let us now discuss a few programs that attack computer systems to cause some damage or to create confusion.

1. Virus

One can launch an application-level attack or a network level attack using a **virus**. In simple terms, a virus is a piece of program code that attaches itself to legitimate program code, and runs when the legitimate program runs. It can then infect other programs in that computer, or programs that are in other computers but on the same network. This is shown in Fig. 1.14. In this example, after deleting all the files from the current user's computer, the virus self-propagates by sending its code to all users whose email addresses are stored in the current user's address book.

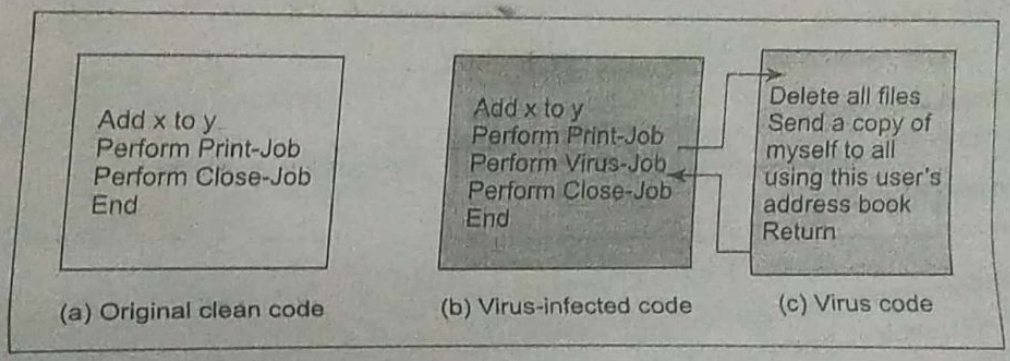


Fig. 1.14 Virus

Viruses can also be triggered by specific events (e.g. a virus could automatically execute at 12 p.m. every day). Usually viruses cause damage to computer and network systems to the extent that they can be repaired, assuming that the organization deploys good backup and recovery procedures.

A virus is a computer program that attaches itself to another legitimate program, and causes damage to the computer system or to the network.

During its lifetime, a virus goes through four phases:

(a) Dormant Phase Here, the virus is idle. It gets activated based on a certain action or event (e.g. the user typing a certain key or a certain date or time is reached, etc). This is an optional phase.

(b) Propagation Phase In this phase, a virus copies itself, and each copy starts creating more copies of itself, thus propagating the virus.

(c) Triggering Phase A dormant virus moves into this phase when the action/event for which it was waiting is initiated.

(d) Execution Phase This is the actual work of the virus, which could be harmless (display some message on the screen) or destructive (delete a file on the disk).

Viruses can be classified into the following categories:

(a) Parasitic Virus This is the most common form of virus. Such a virus attaches itself to executable files and keeps replicating. Whenever the infected file is executed, the virus looks for other executable files to attach itself and spread.

(b) Memory-resident Virus This type of virus first attaches itself to an area of the main memory and then infects every executable program that is executed.

(c) Boot sector Virus This type of virus infects the master boot record of the disk and spreads on the disk when the operating system starts booting the computer.

(d) Stealth Virus This virus has intelligence built in, which prevents anti-virus software programs from detecting it.

(e) Polymorphic Virus A virus that keeps changing its signature (i.e. identity) on every execution, making it very difficult to detect.

(f) Metamorphic Virus In addition to changing its signature like a polymorphic virus, this type of virus keeps rewriting itself every time, making its detection even harder.

There is another popular category of viruses, called the **macro virus**. This virus affects specific application software, such as Microsoft Word or Microsoft Excel. They affect the documents created by users, and spread quite easily since such documents are very commonly exchanged over email. There is a feature called *macro* in these application-software programs, which allows users to write small, useful, utility programs within the documents. Viruses attack these macros, and hence the name *macro virus*.

2. Worm

Similar in concept to a virus, a **worm** is actually different in implementation. A virus modifies a program (i.e. it attaches itself to the program under attack). A worm, however, does not modify a program. Instead, it replicates itself again and again. This is shown in Fig. 1.15. The replication grows so much that ultimately the computer or the network on which the worm resides, becomes very slow, ultimately coming to a halt. Thus, the basic purpose of a worm attack is different from that of a virus. A worm attack attempts to make the computer or the network under attack unusable by eating all its resources.

> *A worm does not perform any destructive actions, and instead, only consumes system resources to bring it down.*

3. Trojan Horse

A Trojan horse is a hidden piece of code, like a virus. However, the purpose of a Trojan horse is different. Whereas the main purpose of a virus is to make some sort of modifications to the target computer or network, a Trojan horse attempts to reveal confidential information to an attacker. The name (Trojan horse) comes from the epic poem *Iliad*. The story says that Greek soldiers hid inside a large hollow horse, which was pulled into the city of Troy by its citizens, unaware of its *contents*. Once the Greek soldiers entered the city of Troy, they opened the gates for the rest of the Greek soldiers.

- The attacker can use many techniques to attack the bank's customers. We illustrate the most common one below.

The attacker sends an email to the legitimate customers of the bank. The email itself appears to have come from the bank. For ensuring this, the attacker exploits the email system to suggest that the sender of the email is some bank official (e.g. accountmanager@citibank.com). This fake email warns the user that there has been some sort of attack on Citibank's computer systems and that the bank wants to issue new passwords to all its customers, or verify their existing PINs, etc. For this purpose, the customer is asked to visit a URL mentioned in the same email. This is conceptually shown in Fig. 1.19.

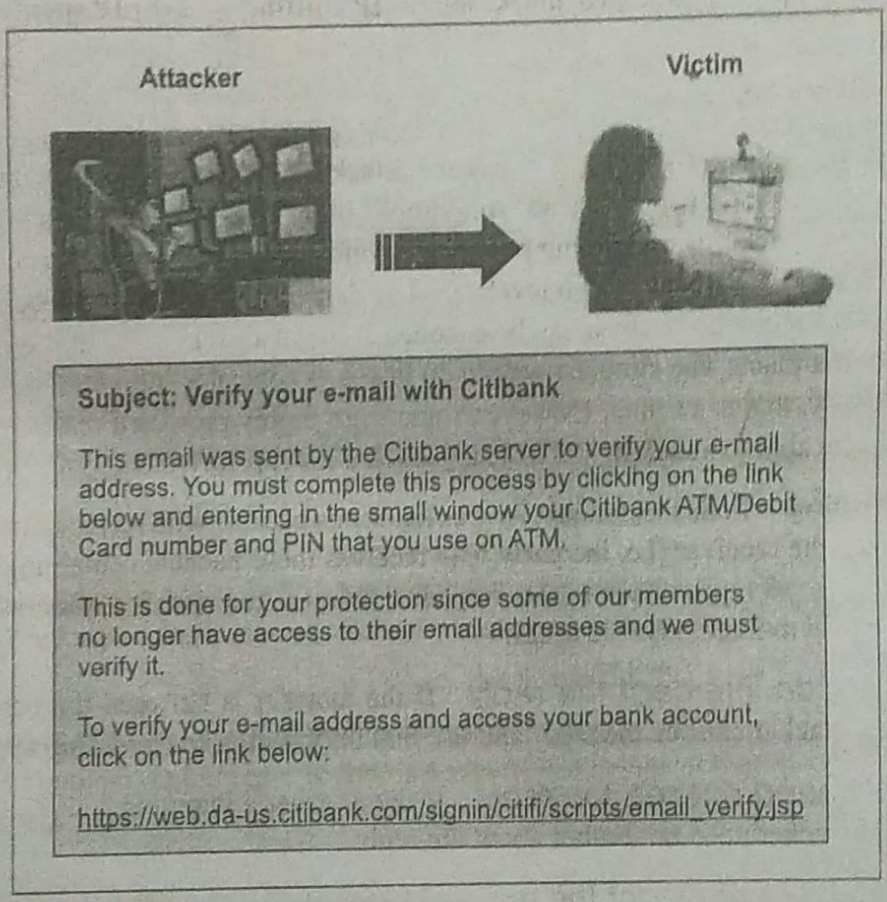


Fig. 1.19 Attacker sends a forged email to the innocent victim (customer)

- When the customer (i.e. the victim) innocently clicks on the URL specified in the email, he/she is taken to the attacker's site, and not the bank's original site. There, the customer is prompted to enter confidential information, such as his/her password or PIN. Since the attacker's fake site looks exactly like the original bank site, the customer provides this information. The attacker gladly accepts this information and displays a *Thank you* to the unsuspecting victim. In the meanwhile, the attacker now uses the victim's password or PIN to access the bank's real site and can perform any transaction as if he/she is the victim!

A real-life example of this kind of attack is reproduced below from the site http://www.fraudwatchinternational.com.

Figure 1.20 shows a fake email sent by an attacker to an authorized PayPal user.

1.5.6 Specific Attacks

1. Sniffing and Spoofing

On the Internet, computers exchange messages with each other in the form of small groups of data, called packets. A packet, like a postal envelope contains the actual data to be sent, and the addressing information. Attackers target these packets, as they travel from the source computer to the destination computer over the Internet. These attacks take two main forms: (a) **Packet sniffing** (also called **snooping**), and (b) **Packet spoofing**. Since the protocol used in this communication is called Internet Protocol (IP), other names for these two attacks are (a) **IP sniffing**, and (b) **IP spoofing**. The meaning remains the same.

Let us discuss these two attacks.

(a) Packet Sniffing Packet sniffing is a passive attack on an ongoing conversation. An attacker need not *hijack* a conversation, but instead, can simply observe (i.e. *sniff*) packets as they pass by. Clearly, to prevent an attacker from sniffing packets, the information that is passing needs to be protected in some ways. This can be done at two levels: (i) The data that is traveling can be encoded in some ways, or (ii) The transmission link itself can be encoded. To read a packet, the attacker somehow needs to access it in the first place. The simplest way to do this is to control a computer via which the traffic goes through. Usually, this is a router. However, routers are highly protected resources. Therefore, an attacker might not be able to attack it, and instead, attack a less-protected computer on the same path.

(b) Packet Spoofing In this technique, an attacker sends packets with an incorrect source address. When this happens, the receiver (i.e. the party who receives these packets containing false addresses) would inadvertently send replies back to this forged address (called **spoofed address**), and not to the attacker. This can lead to three possible cases:

(i) The attacker can intercept the reply If the attacker is between the destination and the forged source, the attacker can see the reply and use that information for *hijacking* attacks.

(ii) The attacker need not see the reply If the attacker's intention was a Denial Of Service (DOS) attack, the attacker need not bother about the reply.

(iii) The attacker does not want the reply The attacker could simply be *angry* with the host, so it may put that host's address as the forged source address and send the packet to the destination. The attacker does not want a reply from the destination, as it wants the host with the forged address to receive it and get confused.

2. Phishing

Phishing has become a big problem in recent times. In 2004, the estimated losses due to phishing were to the tune of USD 137 million, according to Tower Group. Attackers set up fake Web sites, which look like real Web sites. It is quite simple to do so, since creating Web pages involves relatively simple technologies such as HTML, JavaScript, CSS (Cascading Style Sheets), etc. Learning and using these technologies is quite simple. The attacker's modus operandi works as follows.

- The attacker decides to create his/her own Web site, which looks very identical to a real Web site. For example, the attacker can clone Citibank's Web site. The cloning is so clever that the human eye will not be able to distinguish between the real (Citibank's) and fake (attacker's) site.

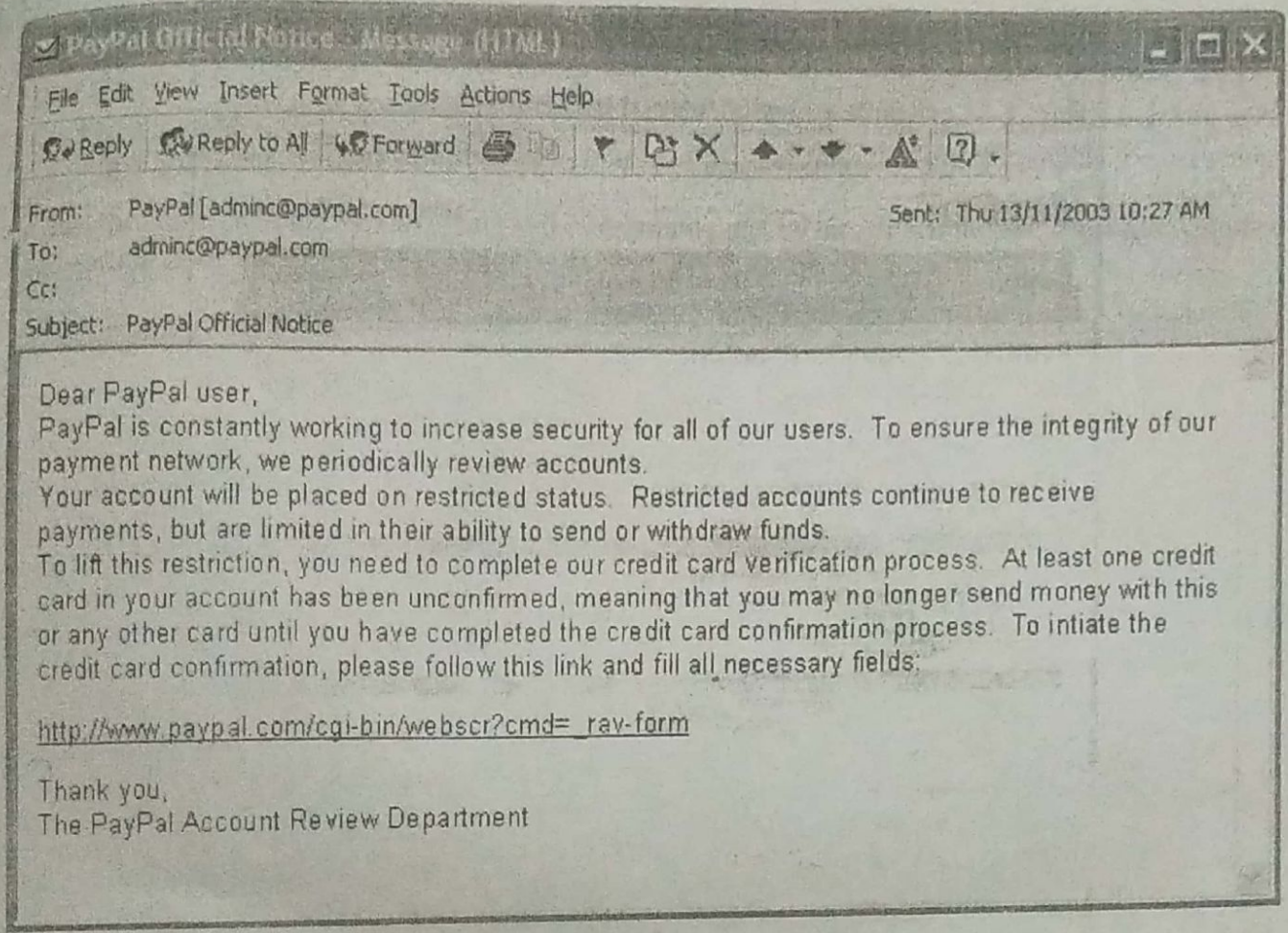


Fig. 1.20 Fake email from the attacker to a PayPal user

As we can see, the attacker is trying to fool the PayPal customer to verify his/her credit-card details. Quite clearly, the aim of the attacker is to access the credit-card information of the customer and then misuse it. Figure 1.21 shows the screen that appears when the user clicks on the URL specified in the fake email.

Once the user provides these details, the attacker's job is easy! He/she simply uses these credit-card details to make purchases on behalf of the cheated card holder!

3. Pharming (DNS Spoofing)

Another attack, known earlier as **DNS spoofing** or **DNS poisoning**, is now called **pharming** attack. As we know, using the **Domain Name System (DNS)**, people can identify Web sites with human-readable names (such as www.yahoo.com), and computers can continue to treat them as IP addresses (such as 120.10.81.67). For this, a special server computer called a DNS server maintains the mappings between domain names and the corresponding IP addresses. The DNS server could be located anywhere. Usually, it is with the Internet Service Provider (ISP) of the users. With this background, the DNS spoofing attack works as follows.

- Suppose that there is a merchant (Bob) whose site's domain name is www.bob.com, and the IP address is 100.10.10.20. Therefore, the DNS entry for Bob in all the DNS servers is maintained as follows:

  www.bob.com 100.10.10.20

- The attacker (say, Trudy) manages to hack and replace the IP address of Bob with her own (say 100.20.20.20) in the DSN server maintained by the ISP of a user, say Alice. Therefore, the DNS server maintained by the ISP of Alice now has the following entry:

  www.bob.com 100.20.20.20

Thus, the contents of the hypothetical DNS table maintained by the ISP would be changed. A hypothetical portion of this table (before and after the attack) is shown in Fig. 1.22.

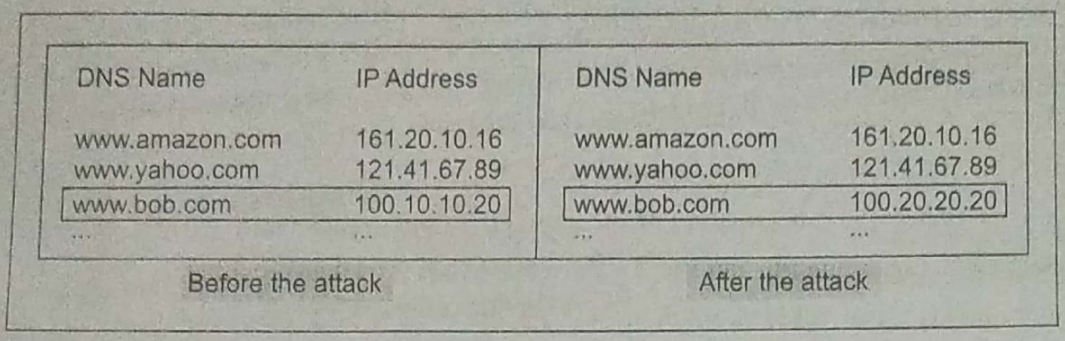


Fig. 1.22 Effect of the DNS attack

- When Alice wants to communicate with Bob's site, her Web browser queries the DNS server maintained by her ISP for Bob's IP address, providing it the domain name (i.e. www.bob.com). Alice gets the replaced (i.e. Trudy's) IP address, which is 100.20.20.20.
- Now, Alice starts communicating with Trudy, believing that she is communicating with Bob!

Such attacks of DNS spoofing are quite common, and cause a lot of havoc. Even worse, the attacker (Trudy) does not have to listen to the conversation on the wire! She has to simply be able to hack the DNS server of the ISP and replace a single IP address with her own!

A protocol called **DNSSec (Secure DNS)** is being used to thwart such attacks. Unfortunately, it is not widely used.

Summary

- Network and Internet security has gained immense prominence in the last few years, as conducting business using these technologies have become very crucial.
- Automation of attacks, privacy concerns, and distance becoming immaterial are some of the key characteristics of modern attacks.
- The principles of any security mechanism are confidentiality, authentication, integrity, non-repudiation, access control, and availability.
- Confidentiality specifies that only the sender and the intended recipients should be able to access the contents of a message.
- Authentication identifies the user of a computer system, and builds a trust with the recipient of a message.

Steganography

It is a technique that facilitates hiding of a message that is to be kept secret inside other messages. This results in the concealment of the secret message itself. This is not encryption

A simple form of steganography is one in which an arrangement of words or letters within an apparently innocuous text spells out the real message. For eg. the sequence of first letter of each word of the overall message spell out the hidden message.

Various techniques used are following:-

1) Character Marking:- Selected letters of printed or typewritten text are overwritten in pencil. The marks are ordinarily not visible unless the paper is held at an angle to bright light.

2) Invisible Ink:- A number of substances can be used for writing but leave no visible trace until heat or some chemical is applied to paper.

3) Pin Punctures:- Small pin punctures on selected letters are ordinarily not visible unless the paper is held in front of light

4) Typewriter Correction Ribbon → used between lines typed with a black ribbon, the result of typing with the Correction tape are visible only under strong light

A Except above mentioned method, people also hide secret message within graphic images. For eg. Suppose that we have a secret

message to send. We can take another image file & replace the last two rightmost bit of each byte of that image with (the next) two bits of our secret message. The resulting image would not look too different and yet carry a secret message inside The receiver reads the last two bits of each byte of image file & reconstruct the secret message.

Advantages: The advantage of steganography is that it can be employed by parties who have something to lose should the fact of their secret communication be discovered.

Drawback: ① It requires a lot of overhead to hide a relatively few bits of information.

② once the system is discovered it becomes virtually worthless.

Hybrid Encryption

Hybrid encryption is a mode of encryption that merges two or more encryption systems. It incorporates a combination of asymmetric and symmetric encryption to benefit from the strengths of each form of encryption. These strengths are respectively defined as speed (symmetric encryption) and security (asymmetric encryption).A hybrid encryption scheme is one that blends the convenience of an asymmetric encryption scheme with the effectiveness of a symmetric encryption scheme. Hybrid encryption is considered a highly secure type of encryption as long as the public and private keys are fully secure.

Hybrid encryption is achieved through data transfer using unique session keys along with symmetrical encryption. Public key encryption is implemented for random symmetric key encryption. The recipient then uses the public key encryption method to decrypt the symmetric key. Once the symmetric key is recovered, it is then used to decrypt the message.

The combination of encryption methods has various advantages. One is that a connection channel is established between two users' sets of equipment. Users then have the ability to communicate through hybrid encryption. Asymmetric encryption can slow down the encryption process, but with the simultaneous use of symmetric encryption, both forms of encryption are enhanced. The result is the added security of the transmittal process along with overall improved system performance.

In cryptography, public-key cryptosystems are convenient in that they do not require the sender and receiver to share a common secret in order to communicate securely (among other useful properties). However, they often rely on complicated mathematical computations and are thus generally much more inefficient than comparable symmetric-key cryptosystems. In many applications, the high cost of encrypting long messages in a public-key cryptosystem can be prohibitive. A **hybrid cryptosystem** is one which combines the convenience of a public-key cryptosystem with the efficiency of a symmetric-key cryptosystem.

A hybrid cryptosystem can be constructed using any two separate cryptosystems:

- a **key encapsulation** scheme, which is a public-key cryptosystem (asymmetric key cryptosystem) and
- a **data encapsulation** scheme, which is a symmetric-key cryptosystem

The hybrid cryptosystem is itself a public-key system, whose public and private keys are the same as in the key encapsulation scheme. The most commonly used hybrid cryptosystems are the **OpenPGP (RFC 4880)** file format and the **PKCS #7 (RFC 2315)** file format, both used by many different systems.

Example

To encrypt a message addressed to Alice (receiver) in a hybrid cryptosystem, Bob (sender)does the following:

1. Obtains Alice's public key.
2. Generates a fresh symmetric key for the data encapsulation scheme.
3. Encrypts the message under the data encapsulation scheme, using the symmetric key just generated.
4. Encrypt the symmetric key under the key encapsulation scheme, using Alice's public key.
5. Send both of these encryptions to Alice.

To decrypt this hybrid ciphertext, Alice does the following:

1. Uses her private key to decrypt the symmetric key contained in the key encapsulation segment.
2. Uses this symmetric key to decrypt the message contained in the data encapsulation segment.

Security

If both the key encapsulation and data encapsulation schemes are secure against chosen cipher attack, then the hybrid scheme inherits that property as well. However, it is possible to construct a hybrid scheme secure against chosen ciphertext attack even if the key encapsulation has a slightly weakened security definition (though the security of the data encapsulation must be slightly stronger).

Key Recovery Attack on Block Cipher:-

A key recovery attack is an adversary's attempt to recover the cryptographic key of an encryption scheme. Historically, cryptanalysis of block ciphers has focused on key recovery, but security against these sorts of attack is a very weak guarantee since it may not necessary to recover the key to obtain partial information about the message or decrypt msg entirely. But modern cryptographic method uses more robust notion of security.

Attacking on a block cipher is called cryptanalysis of block cipher.

- Suppose we have a block cipher $E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ having key size k, and block size n. It is assumed that attacker knows the description of E and can compute it.

Cryptanalysis of Block cipher:- A k-bit key T (called target key) is chosen at random. Let $q \geq 0$ be some integer ~~parameter~~. The adversary has a sequence of q input (plaintext or message) output (ciphertext) —

$$(M_1, C_1), (M_2, C_2) \ldots (M_q, C_q)$$

where $C_1 - C_q$ and $M_1 - M_q$ are all distinct n-bit string. Now the adversary wants to find the target key T.

Let us say that a key K is consistent with the input (plaintext) output (cipher text)

$$(M_1, C_1) \ldots (M_q, C_q) \text{ if } E_K(M_i) = C_i \text{ for all } 1 \leq i \leq q$$

$$\text{Cons}_E((M_1, C_1) \ldots (M_q, C_q))$$

be the set of all keys consistent with the input (plaintext) output (ciphertext). and the target key T is in this set. The set of key is larger containing other keys. The goal of adversary is to find the ~~poor~~ target

key T from this set. If few plain text, ~~block~~ cipher text are the size of the above set will be one, so the adversary can indeed find the target key.

Choosen Message Attack :- $M_1 \dots M_q$ chosen by the adversary He uses a Message (Plain text) function E_K (Encryption standard) and get back the cipher text $C_1 = E_K(M_1)$. It can then decide on a value M_2, feed them & gets back C_2.

This type of attack gives the adversary more power but it may be less realistic in practice.

The most obvious attack strategy is exhaustive key search Here the adversary goes through all possible keys $K \in \{0,1\}^k$ until it finds one that explains the input-output (plaintext, ciphertext) pair.

using $q = 1, i = 1 \dots 2^k$ let T_i denote the i^{th} k bit string

```
EKSE(M1, C1)
  for i = 1 ... 2^k do
  if (E(Ti Mi) = C1) then return Ti
```

This attack always returns a key consistent with the given input output (M_1, C_1), whether or not it is the target key depends on the block cipher. If one imagine the blockcipher to be random, the the blockcipher's key length and block length are relevant in assessing if the above attack will find the right key.

The chances of the attack returning the target key can be increased by testing against more input/output example A small value of q, say k/n is enough that this attack will usually return the target key itself.

To make blockcipher perfectly secure it is necessary that block should be designed in such a way that key computation task would be computationally prohibitive

In cryptography, **DES-X** (or **DESX**) is a variant on the DES (Data Encryption Standard) symmetric-key block cipher intended to increase the complexity of a brute force attack using a technique called *key whitening*.

The original DES algorithm was specified in 1976 with a 56-bit key size: 2^{56} possibilities for the key. There was criticism that an exhaustive search might be within the capabilities of large governments, particularly the United States' National Security Agency (NSA). One scheme to increase the key size of DES without substantially altering the algorithm was DES-X, proposed by Ron Rivest in May 1984.

The algorithm has been included in RSA Security's BSAFE cryptographic library since the late 1980s.

DES-X augments DES by XORing an extra 64 bits of key (K_1) to the plaintext *before* applying DES, and then XORing another 64 bits of key (K_2) *after* the encryption:

$$\text{DES-X}(M) = K_2 \oplus \text{DES}_K(M \oplus K_1)$$

The key size is thereby increased to 56 + (2 × 64) = 184 bits.

However, the effective key size (security) is only increased to 56+64-1-*lb(M)* = 119 - *lb(M)* = ~119 bits, where *M* is the number of chosen plaintext/ciphertext pairs the adversary can obtain, and *lb* denotes the binary logarithm. Moreover key size drops to 88 bits given $2^{32.5}$ known plaintext and using advanced slide attack. (Because of this, some implementations actually make K_2 a strong one way function of K_1 and K.)

DES-X also increases the strength of DES against differential cryptanalysis and linear cryptanalysis, although the improvement is much smaller than in the case of brute force attacks. It is estimated that differential cryptanalysis would require 2^{61} chosen plaintexts (vs. 2^{47} for DES), while linear cryptanalysis would require 2^{60} known plaintexts (vs. 2^{43} for DES.) Note that with 2^{64} plaintexts (known or chosen being the same in this case), DES (or indeed any other block cipher with a 64 bit block size) is totally broken via the elementary codebook attack.

key-recovery attack

A **key-recovery attack** is an adversary's attempt to recover the cryptographic key of an encryption scheme.[1]:52 Historically, cryptanalysis of block ciphers has focused on key-recovery, but security against these sorts of attacks is a very weak guarantee since it may not be necessary to recover the key to obtain partial information about the message or decrypt message entirely.[1]:52 Modern cryptography uses more robust notions of security. Recently, indistinguishability under adaptive chosen-ciphertext attack (IND-CCA2 security) has become the "golden standard" of security.[2]:506 The most obvious key-recovery attack is the exhaustive key-search attack. But modern ciphers often have a key space of size 2^{128} or greater, making such attacks infeasible with current technology.

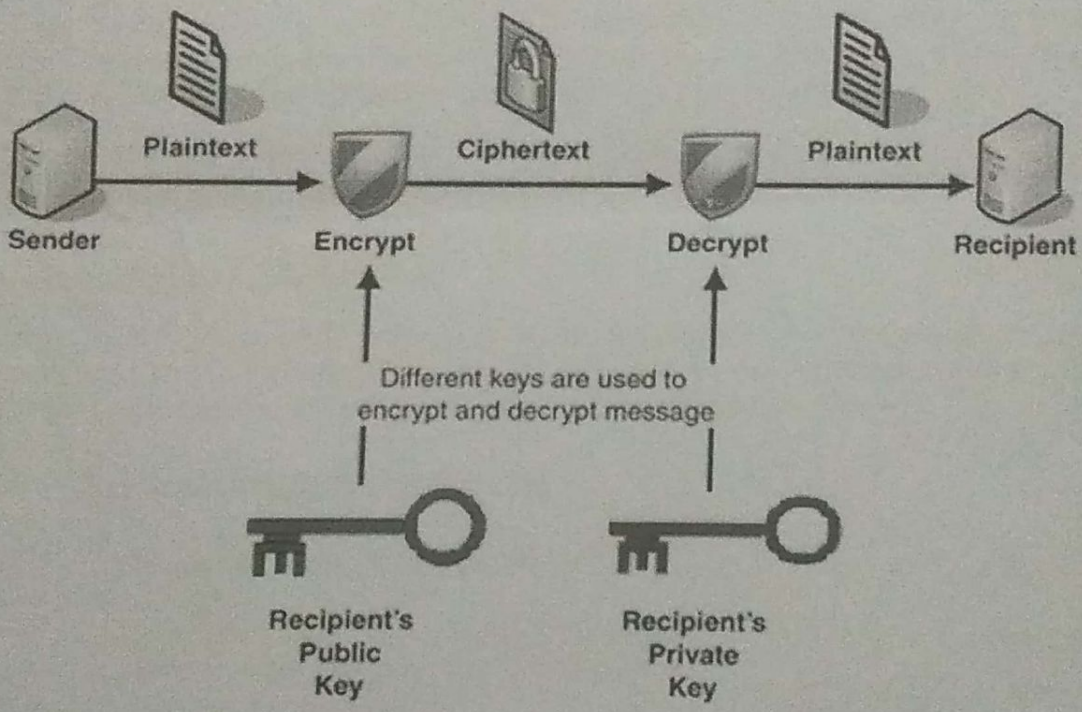
Public Key Cryptography

Unlike symmetric key cryptography, we do not find historical use of public-key cryptography. It is a relatively new concept.

Symmetric cryptography was well suited for organizations such as governments, military, and big financial corporations were involved in the classified communication.

With the spread of more unsecure computer networks in last few decades, a genuine need was felt to use cryptography at larger scale. The symmetric key was found to be non-practical due to challenges it faced for key management. This gave rise to the public key cryptosystems.

The process of encryption and decryption is depicted in the following illustration:



The most important properties of public key encryption scheme are:

- Different keys are used for encryption and decryption. This is a property which set this scheme different than symmetric encryption scheme.
- Each receiver possesses a unique decryption key, generally referred to as his private key.

- Receiver needs to publish an encryption key, referred to as his public key.
- Some assurance of the authenticity of a public key is needed in this scheme to avoid spoofing by adversary as the receiver. Generally, this type of cryptosystem involves trusted third party which certifies that a particular public key belongs to a specific person or entity only.
- Encryption algorithm is complex enough to prohibit attacker from deducing the plaintext from the ciphertext and the encryption (public) key.
- Though private and public keys are related mathematically, it is not be feasible to calculate the private key from the public key. In fact, intelligent part of any public-key cryptosystem is in designing a relationship between two keys.

There are three types of Public Key Encryption schemes. We discuss them in following sections:

RSA Cryptosystem

This cryptosystem is one the initial system. It remains most employed cryptosystem even today. The system was invented by three scholars **Ron Rivest**, **Adi Shamir**, and **Len Adleman** and hence, it is termed as RSA cryptosystem.

We will see two aspects of the RSA cryptosystem, firstly generation of key pair and secondly encryption-decryption algorithms.

Generation of RSA Key Pair

Each person or a party who desires to participate in communication using encryption needs to generate a pair of keys, namely public key and private key. The process followed in the generation of keys is described below:

- **Generate the RSA modulus (n)**
  - Select two large primes, p and q.
  - Calculate n=p*q. For strong unbreakable encryption, let n be a large number, typically a minimum of 512 bits.
- **Find Derived Number (e)**
  - Number **e** must be greater than 1 and less than $(p - 1)(q - 1)$.
  - There must be no common factor for e and $(p - 1)(q - 1)$ except for 1. In other words two numbers e and $(p - 1)(q - 1)$ are coprime.