Catalog

Model of network security

A model for much of what we will be discussing is captured, in very general terms, in Figure below. 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.

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

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

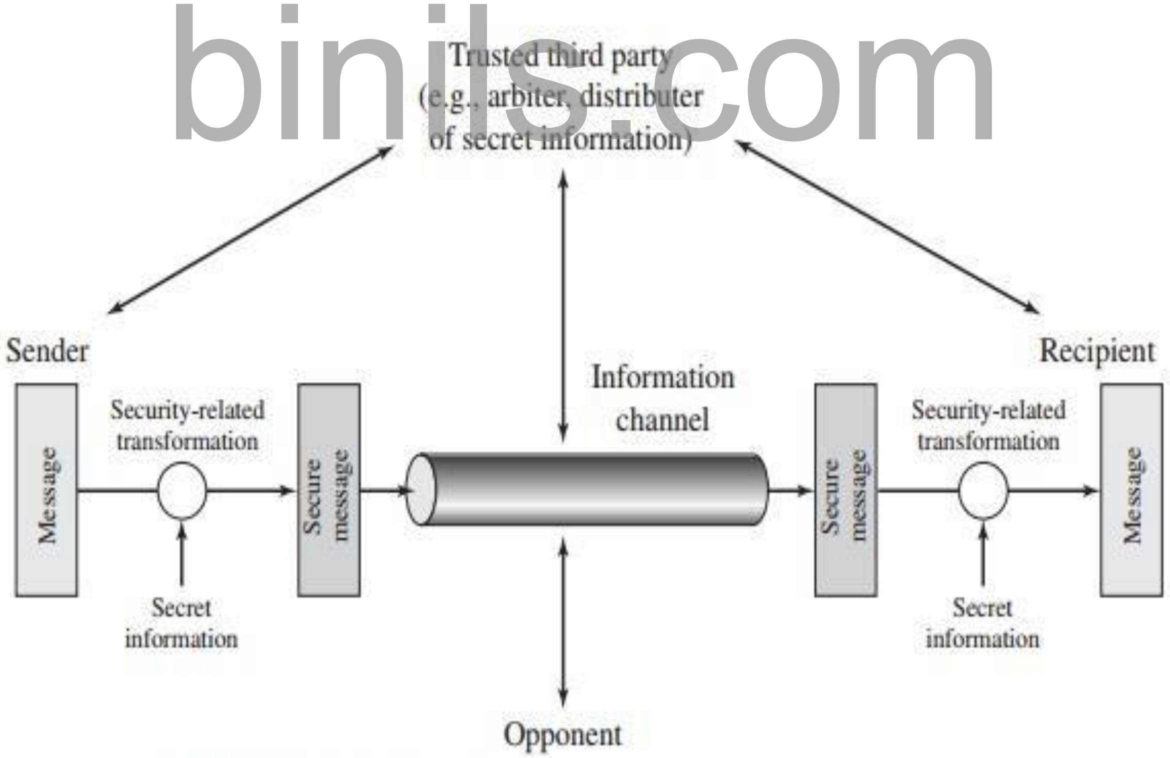


Figure 1.4 Model for Network Security

● 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.

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.

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

A general model of these other situations is illustrated by Figure , which reflects a concern for protecting an information system from unwanted access.

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).

Programs can present two kinds of threats:

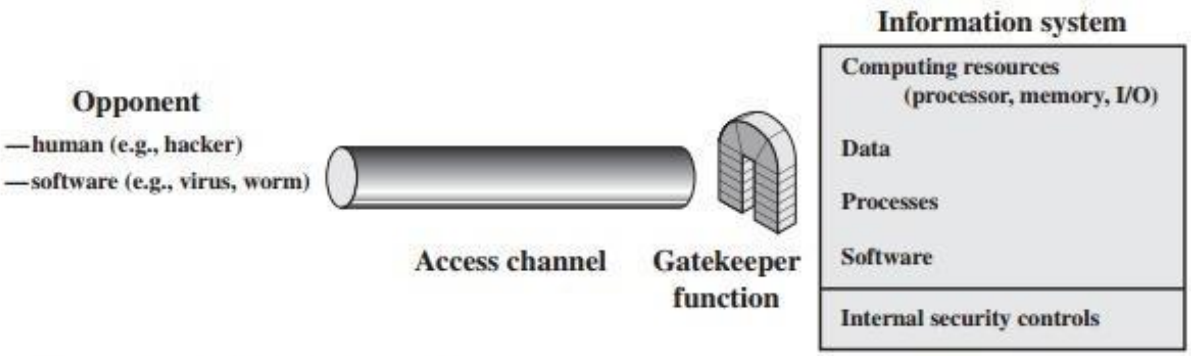


Figure 1.5 Network Access Security Model

- ▫ 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 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 and wide area networks, the problems are compounded. ITU-T3 Recommendation X.800, Security Architecture for OSI, defines such a systematic approach.4 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 archi-tecture focuses on security attacks, mechanisms, and services. These can be defined briefly as

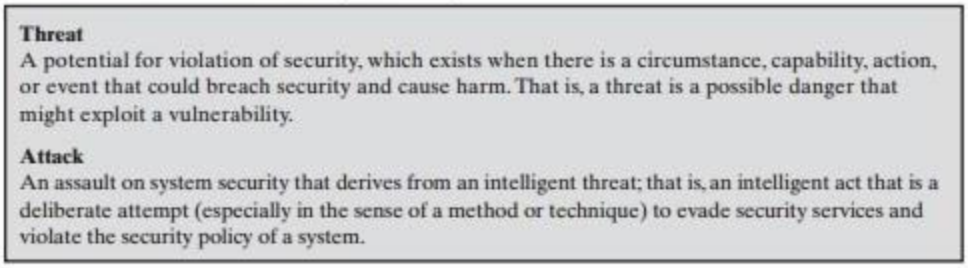
• **Security attack:** Any action that compromises the security of informationowned by an organization.

• **Security mechanism:** A process (or a device incorporating such a process) thatis designed to detect, prevent, or recover from a security attack.

• **Security service:** A processing or communication service that enhances thesecurity 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.

Table provides definitions taken from RFC 2828, Internet Security Glossary.

Table 1.1 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.

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

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 the release of message contents and traffic analysis.

Release of message contents:

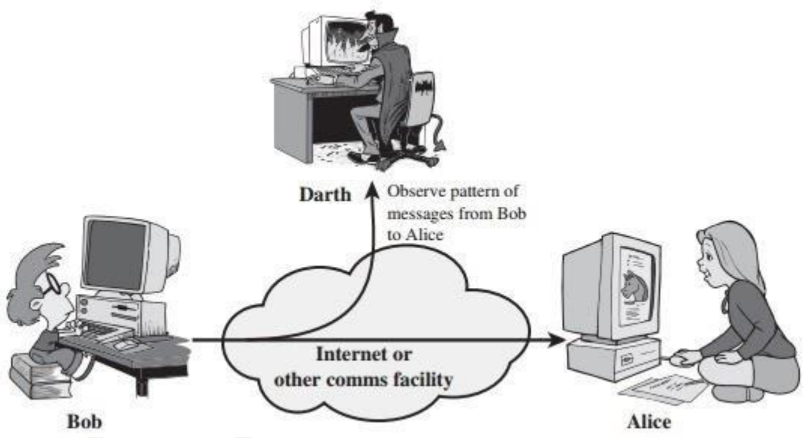
It is easily understood (Figure 1.2a). A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions.

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

Traffic analysis:

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.


Darth
Observe pattern of messages from Bob to Alice
Internet or other comms facility
Bob
Alice

(b) Traffic analysis

Figure 1.2 Passive Attacks

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

Passive attacks are very difficult to detect, because they do not involve any alteration of the data. Typically, the message traffic is sent and received in an apparently normal fashion, and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern.

However, it is feasible to pre-vent the success of these attacks, usually by means of encryption. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

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.

Masquerade:

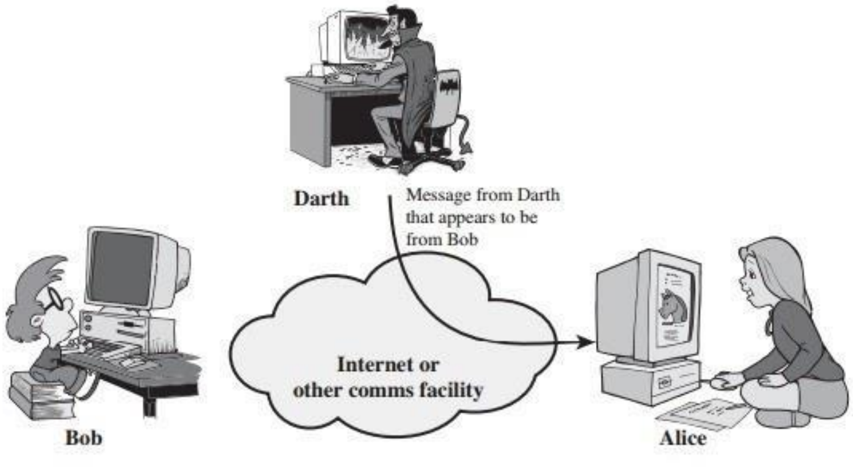
It takes place when one entity pretends to be a different entity (Figure 1.3a). A masquerade attack usually includes one of the other forms of active attack.

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.

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

Modification of messages:

It means that some portion of a legitimatemessage is altered, or that messages are delayed or reordered, to produce an unauthorized effect (Figure 1.3c). 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.”

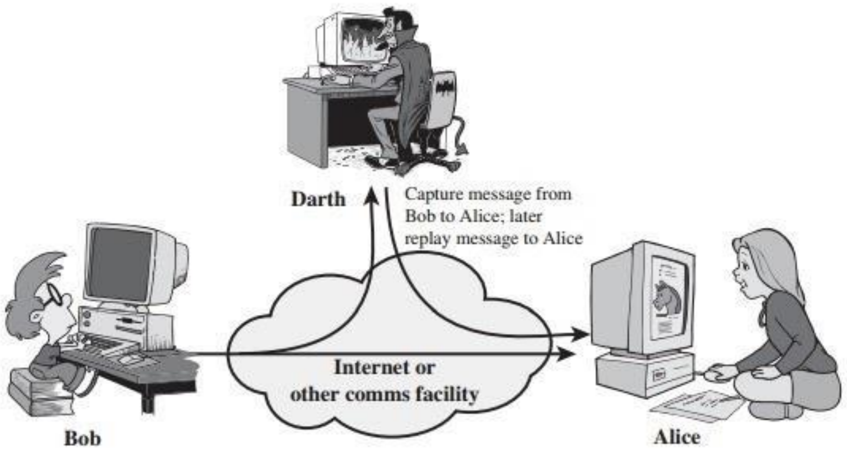

Darth
Message from Darth that appears to be from Bob
Internet or other comms facility
Bob
Alice

(a) Masquerade

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

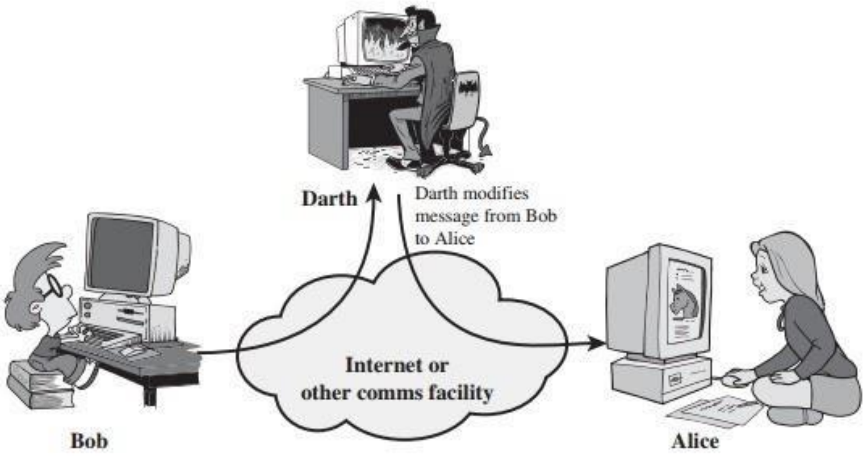
Replay:

It involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect (Figure 1.3b).


Darth
Capture message from Bob to Alice; later replay message to Alice
Internet or other comms facility
Bob
Alice

(b) Replay

Figure 1.3 Active attacks (*Continued*)


Darth
Darth modifies message from Bob to Alice
Internet or other comms facility
Bob
Alice

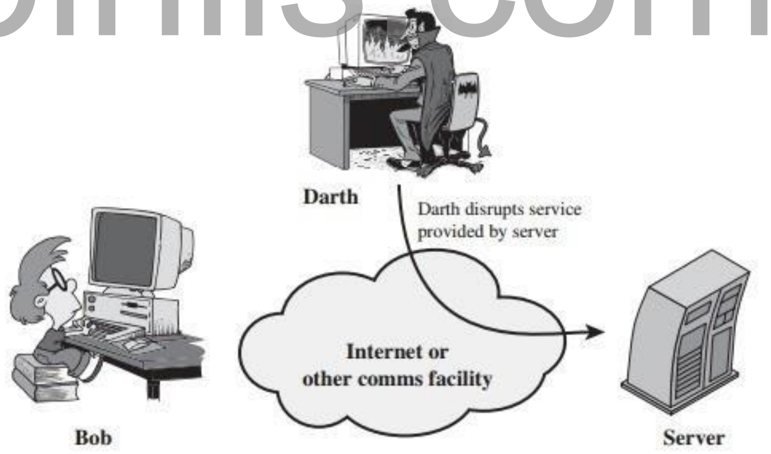
(c) Modification of messages

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

The **denial of service:**

It prevents or inhibits the normal use or management of communications facilities (Figure 1.3d). 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.


Darth
Darth disrupts service provided by server
Internet or other comms facility
Bob
Server

(d) Denial of service

Figure 1.3 Active attacks

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

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 of the wide variety of potential physical, software, and network vulnerabilities. Instead, the goal is to detect active attacks and to recover from any disruption or delays caused by them. If the detection has a deterrent effect, it may also contribute to prevention.

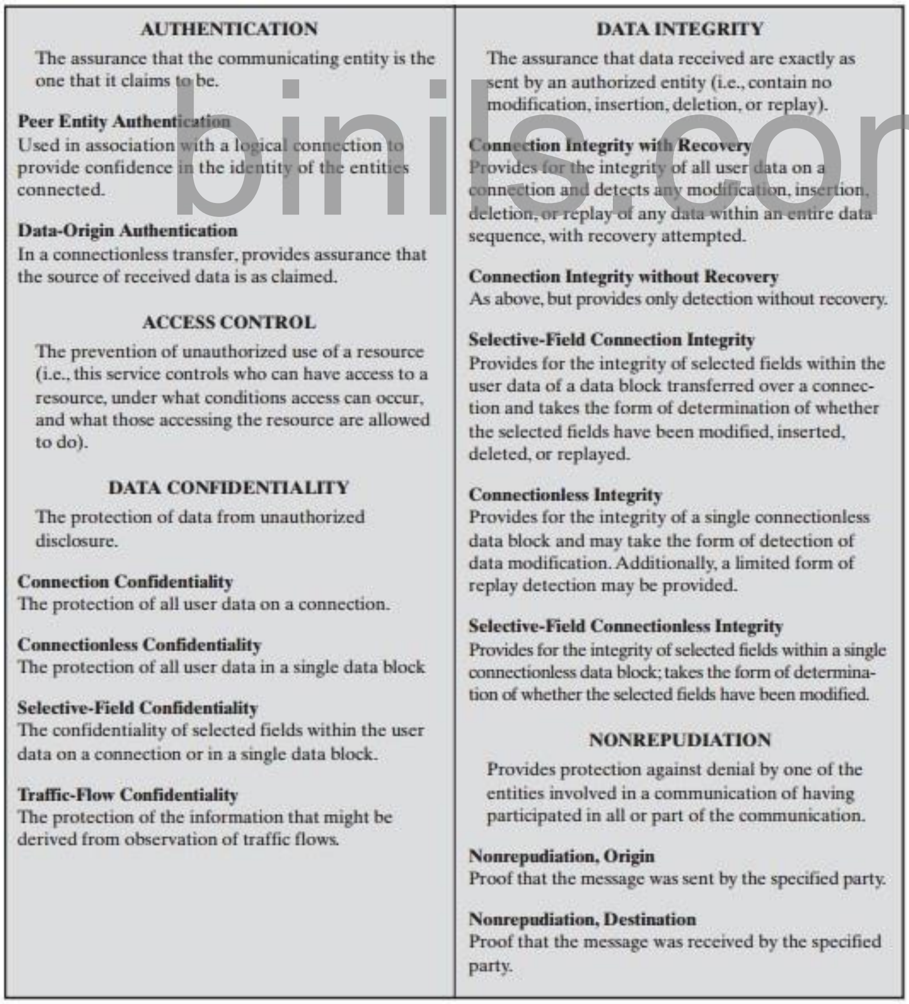
SECURITY SERVICES

X.800 defines a security service as a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers.

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

X.800 divides these services into five categories and fourteen specific services (Table 1.2). We look at each category in turn.

Table 1.2 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.

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

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 X.800:

• **Peer entity authentication:** Provides for the corroboration of the identityof a peer entity in an association. Two entities are considered peers if they implement to same protocol in different systems; e.g., two TCP modules in two communicating systems. Peer entity authentication 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 performing either a masquerade or an unauthorized replay of a previous connection.

• **Data origin authentication:** Provides for the corroboration of the sourceof 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. 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, when a TCP connection is set up between two systems, this broad protection prevents the release of any user data transmitted over the TCP connection.

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 without regard to any larger context, generally provides protection against message modification only.

We can make a distinction between 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 alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message.

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.

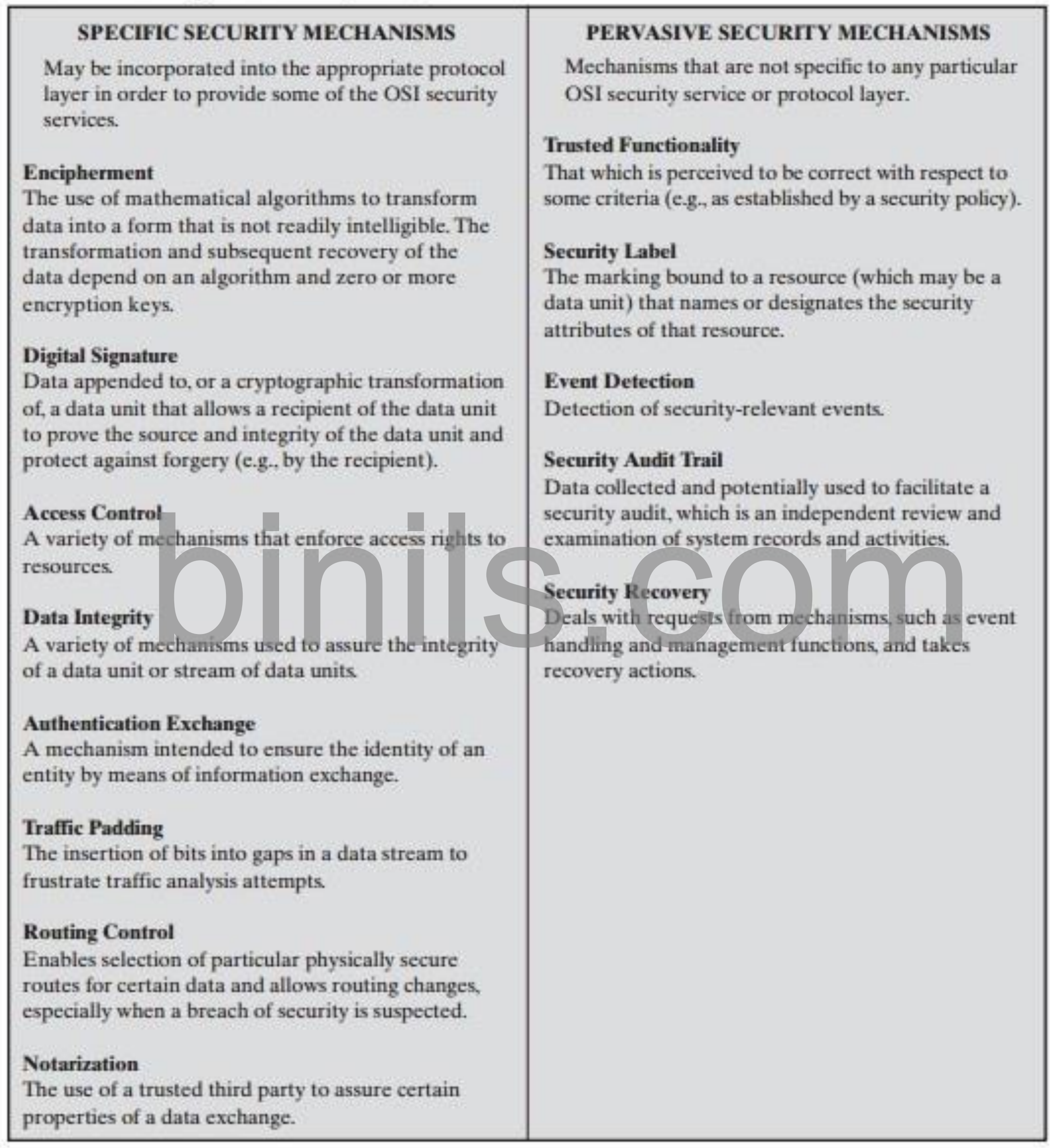
X.800 treats availability as a property to be associated with various security services. However, it makes sense to call out specifically an availability service. An availability service is one that protects a system to ensure its availability.

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

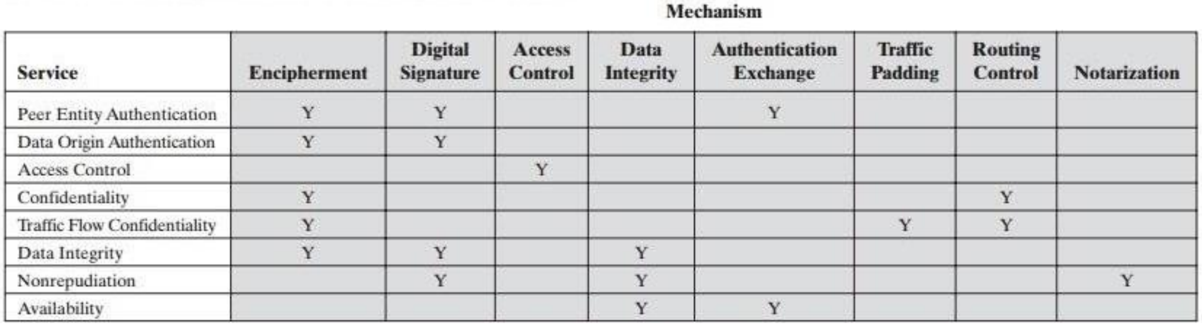
Table 1.3 lists the security mechanisms defined in X.800. The mechanisms are divided into those that are implemented in a specific protocol layer, such as TCP or an application-layer protocol, and those that are not specific to any particular protocol layer or security service.

Table 1.3 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.	

Table 1.4 Relationship Between Security Services and Mechanisms



	Mechanism							
Service	**Encipherment**	**Digital Signature**	**Access Control**	**Data Integrity**	**Authentication Exchange**	**Traffic Padding**	**Routing Control**	**Notarization**
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			

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

X.800 distinguishes between reversible encipherment mechanisms and irreversible encipherment mechanisms. A reversible encipherment mechanism is simply an encryption algorithm that allows data to be encrypted and subsequently decrypted. Irreversible encipherment mechanisms include hash algo-rithms and message authentication codes, which are used in digital signature and message authentication applications.

Table 1.4, based on one in X.800, indicates the relationship between security services and security mechanisms.

A model for much of what we will be discussing is captured, in very general terms, in Figure 1.4. A message is to be transferred from one party to another across some sort of Internet service. 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.

TRANSPOSITION TECHNIQUES

All the techniques examined so far involve the substitution of a ciphertext symbol for a plaintext symbol. A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a transposition cipher.

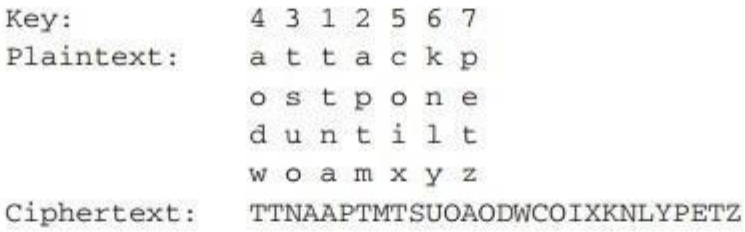
The simplest such cipher is the **rail fence** technique, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows. For example, to encipher the message “meet me after the toga party” with a rail fence of depth 2, we write the following:

m e m a t r h t g p r y
e t e f e t e o a a t

The encrypted message is

MEMATRHTGPRYETEFETEOAAT

This sort of thing would be trivial to cryptanalyze. A more complex scheme is to write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns. The order of the columns then becomes the key to the algorithm. For example,

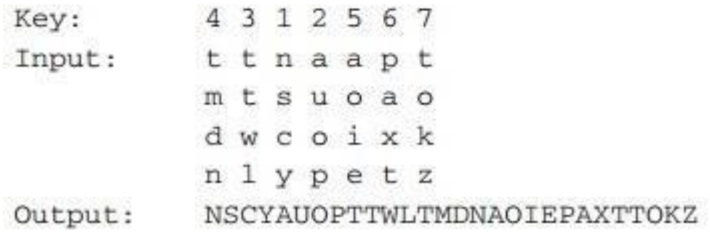


```
Key:            4 3 1 2 5 6 7
Plaintext:      a t t a c k p
                o s t p o n e
                d u n t i l t
                w o a m x y z
Ciphertext:     TTNAAPTMTSUOAODWCOIXKNLYPETZ
```

Thus, in this example, the key is 4312567. To encrypt, start with the column that is labeled 1, in this case column 3. Write down all the letters in that column. Proceed to column 4, which is labeled 2, then column 2, then column 1, then columns 5, 6, and 7.

A pure transposition cipher is easily recognized because it has the same letter frequencies as the original plaintext. For the type of columnar transposition just shown, cryptanalysis is fairly straightforward and involves laying out the ciphertext in a matrix and playing around with column positions. Digram and trigram frequency tables can be useful.

The transposition cipher can be made significantly more secure by performing more than one stage of transposition. The result is a more complex permutation that is not easily reconstructed. Thus, if the foregoing message is reencrypted using the same algorithm,



```
Key:      4 3 1 2 5 6 7
Input:    t t n a a p t
          m t s u o a o
          d w c o i x k
          n l y p e t z
Output:   NSCYAUOPTTWLTMDNAOIEPAXTTOKZ
```

To visualize the result of this double transposition, designate the letters in the original plaintext message by the numbers designating their position. Thus, with 28 letters in the message, the original sequence of letters is

01 02 03 04 05 06 07 08 09 10 11 12 13 14
15 16 17 18 19 20 21 22 23 24 25 26 27 28

After the first transposition, we have

03 10 17 24 04 11 18 25 02 09 16 23 01 08
15 22 05 12 19 26 06 13 20 27 07 14 21 28

which has a somewhat regular structure. But after the second transposition, we have

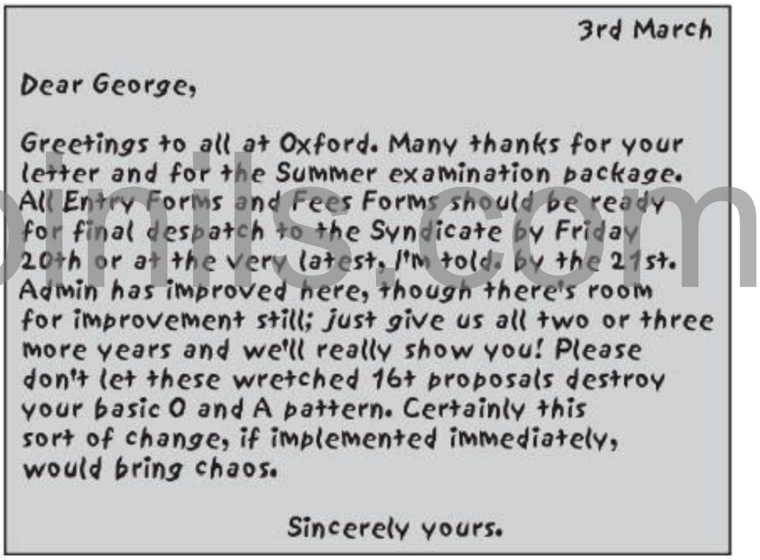
17 09 05 27 24 16 12 07 10 02 22 20 03 25
15 13 04 23 19 14 11 01 26 21 18 08 06 28

This is a much less structured permutation and is much more difficult to cryptanalyze.

STEGANOGRAPHY

A plaintext message may be hidden in one of two ways. The methods of **steganography** conceal the existence of the message, whereas the methods of cryp-tography render the message unintelligible to outsiders by various transformations of the text.

A simple form of steganography, but one that is time-consuming to construct, is one in which an arrangement of words or letters within an apparently innocuous text spells out the real message. For example, the sequence of first letters of each word of the overall message spells out the hidden message. Figure 2.9 shows an example in which a subset of the words of the overall message is used to convey the hidden message. See if you can decipher this; it’s not too hard.



3rd March

Dear George,

Greetings to all at Oxford. Many thanks for your
letter and for the Summer examination package.
All Entry Forms and Fees Forms should be ready
for final despatch to the Syndicate by Friday
20th or at the very latest, I'm told, by the 21st.
Admin has improved here, though there's room
for improvement still; just give us all two or three
more years and we'll really show you! Please
don't let these wretched 16+ proposals destroy
your basic O and A pattern. Certainly this
sort of change, if implemented immediately,
would bring chaos.

Sincerely yours.

Figure 2.9 A Puzzle for Inspector Morse
(From The Silent World of Nicholas Quinn, *by Colin Dexter)*

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

Various other techniques have been used historically; some examples are the following:

- **Character marking:** Selected letters of printed or typewritten text are over-written in pencil. The marks are ordinarily not visible unless the paper is held at an angle to bright light.
- **Invisible ink:** A number of substances can be used for writing but leave novisible trace until heat or some chemical is applied to the paper.

- **Pin punctures:** Small pin punctures on selected letters are ordinarily not visible unless the paper is held up in front of a light.
- **Typewriter correction ribbon:** Used between lines typed with a black ribbon, theresults of typing with the correction tape are visible only under a strong light.

Although these techniques may seem archaic, they have contemporary equivalents. Hiding a message by using the least significant bits of frames on a CD. For example, the Kodak Photo CD format's maximum resolution is 2048 _ 3072 pixels, with each pixel containing 24 bits of RGB color information.

The least significant bit of each 24-bit pixel can be changed without greatly affecting the quality of the image. The result is that you can hide a 2.3-megabyte message in a single digital snapshot. There are now a number of software packages available that take this type of approach to steganography.

Steganography has a number of drawbacks when compared to encryption. It requires a lot of overhead to hide a relatively few bits of information, although using a scheme like that proposed in the preceding paragraph may make it more effective.Also, once the system is discovered, it becomes virtually worthless. This problem, too, can be overcome if the insertion method depends on some sort of key. Alternatively, a message can be first encrypted and then hidden using steganography.

The advantage of steganography is that it can be employed by parties who have something to lose should the fact of their secret communication (not necessarily the content) be discovered. Encryption flags traffic as important or secret or may identify the sender or receiver as someone with something to hide.

Foundations of modern cryptography

Modern cryptography is the cornerstone of computer and communications security. Its foundation is based on various concepts of mathematics such as number theory, computational-complexity theory, and probability theory.

Characteristics of Modern Cryptography

There are three major characteristics that separate modern cryptography from the classical approach.

Perfect Security

- A cipher system is said to offer perfect secrecy if, on seeing the ciphertext the interceptor gets **no extra information**about the plaintext than he had before the ciphertext was observed.
- In a cipher system with perfect secrecy the interceptor is "forced" to guess the plaintext.

Classic Cryptography	**Modern Cryptography**
It manipulates traditional characters, i.e., letters and digits directly.	It operates on binary bit sequences.
It is mainly based on 'security through obscurity'. The techniques employed for coding were kept secret and only the parties involved in communication knew about them.	It relies on publicly known mathematical algorithms for coding the information. Secrecy is obtained through a secrete key which is used as the seed for the algorithms. The computational difficulty of algorithms, absence of secret key, etc., make it impossible for an attacker to obtain the original information even if he knows the algorithm used for coding.
It requires the entire cryptosystem for communicating confidentially.	Modern cryptography requires parties interested in secure communication to possess the secret key only.

- An encryption scheme satisfies perfect secrecy if for all messages m1, m2 in message space M and all ciphertexts c ∈ C, we have

where both probabilities are taken over the choice of K in K and over the coin tosses of the (possibly) probabilistic algorithm Enc.

- Intuitively, we might want to define perfect security of an encryption scheme as follows: Given a ciphertext all messages are equally likely.
- This can be formulated as: For all m(0) , m(1) ∈ M and c ∈ C we have:

  $Pr[M = m (0) |C = c] = Pr[M = m (1) |C = c]$

- The probability here is over the randomness used in the Gen and Enc algorithms and the probability distribution over the message space.

Definition (One: Perfect Security)

- We want the ciphertext to provide no additional information about the message
- Definition (One: Perfect Security)

  For all m ∈ M and c ∈ C, we have:

  $$Pr[M = m|C = c] = Pr[M = m]$$

- Here we are assuming that c ∈ C has Pr[C = c] > 0. Everywhere this assumption will be implicit

Definition (Two: Perfect Security)

- We want to say that the probability to generate a ciphertext given a message is independent of the message
- Definition (Two: Perfect Security)

  For all m ∈ M and c ∈ C we have:

  $$Pr[C = c|M = m] = Pr[C = c]$$

Definition (Three: Perfect Security)

- We want to say that the probability of generating a ciphertext given as message m(0) , is same as the probability of generating that ciphertext given any other different message m(1)
- Definition (Three: Perfect Security)

For any messages m(0) , m(1) ∈ M and c ∈ C we have:

$\Pr[C = c|M = m\ (0)\ ] = \Pr[C = c|M = m\ (1)\ ]$

Shannon's Original Definition of Secrecy

- Shannon defines perfect secrecy for secret-key systems and shows that they exist.
- A secret-key cipher obtains perfect secrecy if for all plaintexts *x* and all ciphertexts *y* it holds that $Pr(x) = Pr(x|y)$.
- In other words, a ciphertext y gives no information about the plaintext

Information theory

- **Information theory** studies the quantification, storage, and communication of information.
- A key measure in information theory is entropy. Entropy quantifies the amount of uncertainty involved in the value of a random variable or the outcome of a random process.
- For example, identifying the outcome of a fair coin flip (with two equally likely outcomes) provides less information (lower entropy) than specifying the outcome from a roll of a die (with six equally likely outcomes).
- Some other important measures in information theory are mutual information, channel capacity, error exponents, and relative entropy.

Quantities of information

- Information theory is based on probability theory and statistics.

- Information theory often concerns itself with measures of information of the distributions associated with random variables.
- Important quantities of information are entropy, a measure of information in a single random variable, and mutual information, a measure of information in common between two random variables.

A common unit of information is the bit, based on the binary logarithm

Entropy of an information source

- Based on the probability mass function of each source symbol to be communicated, the Shannon entropy *H*, in units of bits (per symbol), is given by
- where p_i is the probability of occurrence of the *i*-th possible value of the source symbol.
- This equation gives the entropy in the units of "bits" (per symbol) because it uses a logarithm of base 2, and this base-2 measure of entropy has sometimes been called the shannon in his honor.

- If one transmits 1000 bits (0s and 1s), and the value of each of these bits is known to the receiver (has a specific value with certainty) ahead of transmission, it is clear that no information is transmitted.
- If, however, each bit is independently equally likely to be 0 or 1, 1000 shannons of information (more often called bits) have been transmitted. Between these two extremes, information can be quantified as follows.
- If $\mathbb{X}$ is the set of all messages $\{x_1, ..., x_n\}$ that *X* could be, and *p*(*x*) is the probability of some x € X , then the entropy, *H*, of *X* is defined:[

$$H(X) = \mathbb{E}_X[I(x)] = -\sum_{x \in \mathbb{X}} p(x) \log p(x).$$

- The special case of information entropy for a random variable with two outcomes is the binary entropy function, usually taken to the logarithmic base 2, thus having the shannon (Sh) as unit:

$$H_{\mathrm{b}}(p) = -p\log_2 p - (1-p)\log_2(1-p).$$

Joint entropy

- The *joint entropy* of two discrete random variables *X* and *Y* is merely the entropy of their pairing: (*X*, *Y*). This implies that if *X* and *Y* are independent, then their joint entropy is the sum of their individual entropies.
- For example, if (*X*, *Y*) represents the position of a chess piece — *X* the row and *Y* the column, then the joint entropy of the row of the piece and the column of the piece will be the entropy of the position of the piece.
  - Despite similar notation, joint entropy should not be confused with *cross entropy*.

$$H(X,Y) = \mathbb{E}_{X,Y}[-\log p(x,y)] = -\sum_{x,y} p(x,y)\log p(x,y)$$

Conditional entropy (equivocation)

- The *conditional entropy* or *conditional uncertainty* of *X* given random variable *Y* (also called the *equivocation* of *X* about *Y*) is the average conditional entropy over *Y*:
- Because entropy can be conditioned on a random variable or on that random variable being a certain value, care should be taken not to confuse these two definitions of conditional entropy, the former of which is in more common use. A basic property of this form of conditional entropy is that:

$$H(X|Y) = \mathbb{E}_Y[H(X|y)] = -\sum_{y\in Y} p(y)\sum_{x\in X} p(x|y)\log p(x|y) = -\sum_{x,y} p(x,y)\log p(x|y).$$

- The *conditional entropy* or *conditional uncertainty* of *X* given random variable *Y* (also called the *equivocation* of *X* about *Y*) is the average conditional entropy over *Y*:

- Because entropy can be conditioned on a random variable or on that random variable being a certain value, care should be taken not to confuse these two definitions of conditional entropy, the former of which is in more common use. A basic property of this form of conditional entropy is that:

$$H(X|Y) = H(X,Y) - H(Y).$$

Mutual information (Transinformation)

- *Mutual information* measures the amount of information that can be obtained about one random variable by observing another. It is important in communication where it can be used to maximize the amount of information shared between sent and received signals. The mutual information of *X* relative to *Y* is given by:

$$I(X;Y) = \mathbb{E}_{X,Y}[SI(x,y)] = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)}$$

- where SI (*S*pecific mutual *I*nformation) is the pointwise mutual information.
- A basic property of the mutual information is that

$$I(X;Y) = H(X) - H(X|Y).$$

- That is, knowing *Y*, we can save an average of *I*(*X*; *Y*) bits in encoding *X* compared to not knowing *Y*.
- Mutual information is symmetric:

$$I(X;Y) = I(Y;X) = H(X) + H(Y) - H(X,Y).$$

Kullback–Leibler Divergence (Information Gain):

- The *Kullback–Leibler divergence* (or *information divergence*, *information gain*, or *relative entropy*) is a way of comparing two distributions: a "true" probability distribution *p(X)*, and an arbitrary probability distribution *q(X)*.
- If we compress data in a manner that assumes *q(X)* is the distribution underlying some data, when, in reality, *p(X)* is the correct distribution, the Kullback–Leibler divergence is the number of average additional bits per datum necessary for compression.
- It is thus defined

$$D_{\mathrm{KL}}(p(X)\|q(X)) = \sum_{x\in X} -p(x)\log q(x) - \sum_{x\in X} -p(x)\log p(x) = \sum_{x\in X} p(x)\log\frac{p(x)}{q(x)}.$$

Coding theory

- Coding theory is one of the most important and direct applications of information theory.
- It can be subdivided into source coding theory and channel coding theory.
- Using a statistical description for data, information theory quantifies the number of bits needed to describe the data, which is the information entropy of the source.

- Data compression (source coding): There are two formulations for the compression problem:
    - lossless data compression: the data must be reconstructed exactly;
    - lossy data compression: allocates bits needed to reconstruct the data, within a specified fidelity level measured by a distortion function. This subset of information theory is called *rate–distortion theory*.
- Error-correcting codes (channel coding): While data compression removes as much redundancy as possible, an error correcting code adds just the right kind of redundancy (i.e., error correction) needed to transmit the data efficiently and faithfully across a noisy channel.

Product Cryptosystems

- Data encryption scheme in which the ciphertext produced by encrypting a plaintext document is subjected to further encryption.
- By combining two or more simple transposition ciphers or substitution ciphers, a more secure encryption may result.
- A cryptosystem ***S=(P,K, C,e,d)*** with the sets ofplaintexts ***P***, keys ***K*** and cryptotexts***C*** andencryption (decryption) algorithms ***e*** (***d***) is called **endomorphic** if ***P=C***.
- If $S_1=(P,K_1, P,e^{(1)},d^{(1)})$ and $S_2=(P,K_2, P,e^{(2)},d^{(2)})$ are endomorphic cryptosystems,then the**product cryptosystem** is
- $\boldsymbol{S_1}$Ä$\boldsymbol{S_2=(P,K_1}$Ä$\boldsymbol{K_2, P,e,d)}$**,**
- where encryption is performed by the procedure
- $\boldsymbol{e_{(k1, k2)}(w) = e_{k2}(e_{k1}(w))}$
- and decryption by the procedure
- $\boldsymbol{d_{(k1, k2)}(c) = d_{k1}(d_{k2}(c))}$

Cryptanalysis

- Cr**yptanalysis** is the study of analyzing information systems in order to study the hidden aspects of the systems.
- Cryptanalysis is used to breach cryptographic security systems and gain access to the contents of encrypted messages, even if the cryptographic key is unknown.
- In addition to mathematical analysis of cryptographic algorithms, cryptanalysis includes the study of side-channel attacks that do not target weaknesses in the cryptographic algorithms themselves, but instead exploit weaknesses in their implementation.

Methods

- *Ciphertext-only*: the cryptanalyst has access only to a collection of ciphertexts or codetexts.

- *Known-plaintext*: the attacker has a set of ciphertexts to which he knows the corresponding plaintext.
- *Chosen-plaintext* (*chosen-ciphertext*): the attacker can obtain the ciphertexts (plaintexts) corresponding to an arbitrary set of plaintexts (ciphertexts) of his own choosing.
- *Adaptive chosen-plaintext*: like a chosen-plaintext attack, except the attacker can choose subsequent plaintexts based on information learned from previous encryptions. Similarly *Adaptive chosen ciphertext attack*.
- *Related-key attack*: Like a chosen-plaintext attack, except the attacker can obtain ciphertexts encrypted under two different keys. The keys are unknown, but the relationship between them is known; for example, two keys that differ in the one bit.