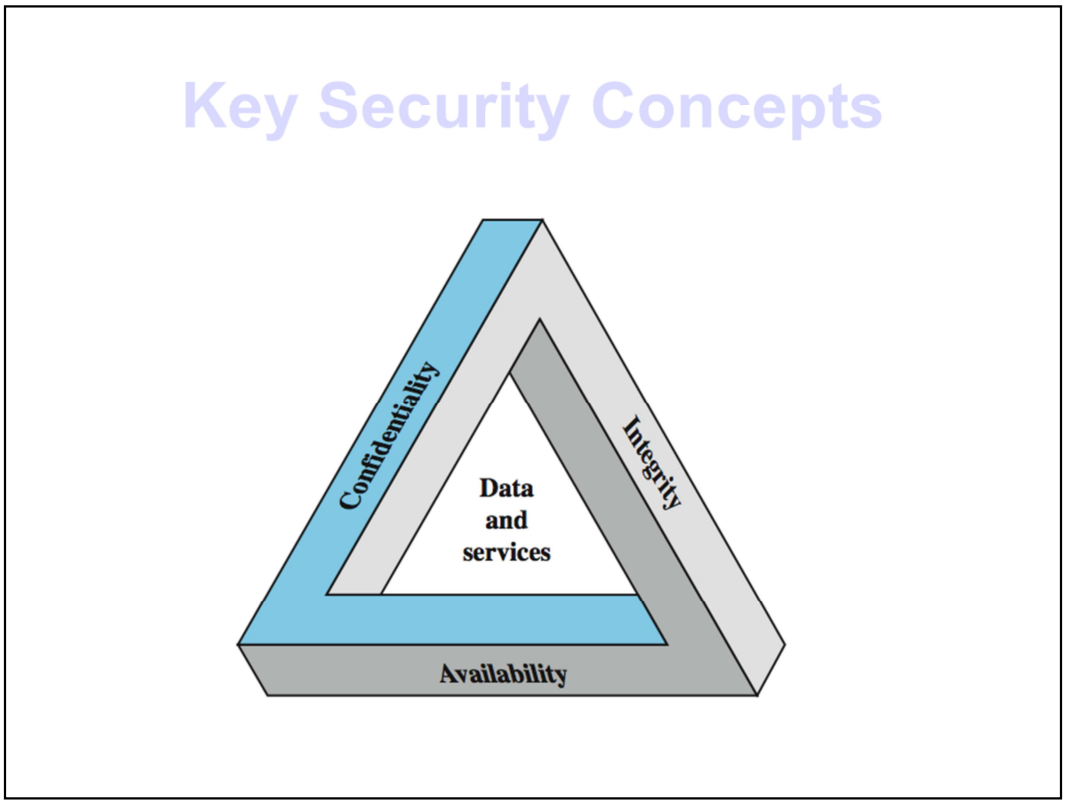
Cryptography and Network Security



These three concepts form what is often referred to as the **CIA triad** (Figure 1.1). The three concepts embody the fundamental security objectives for both data and for information and computing services. FIPS PUB 199 provides a useful characterization of these three objectives in terms of requirements and the definition of a loss of security in each category:

• **Confidentiality** (covers both data confidentiality and privacy): preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. A loss of confidentiality is the unauthorized disclosure of information.

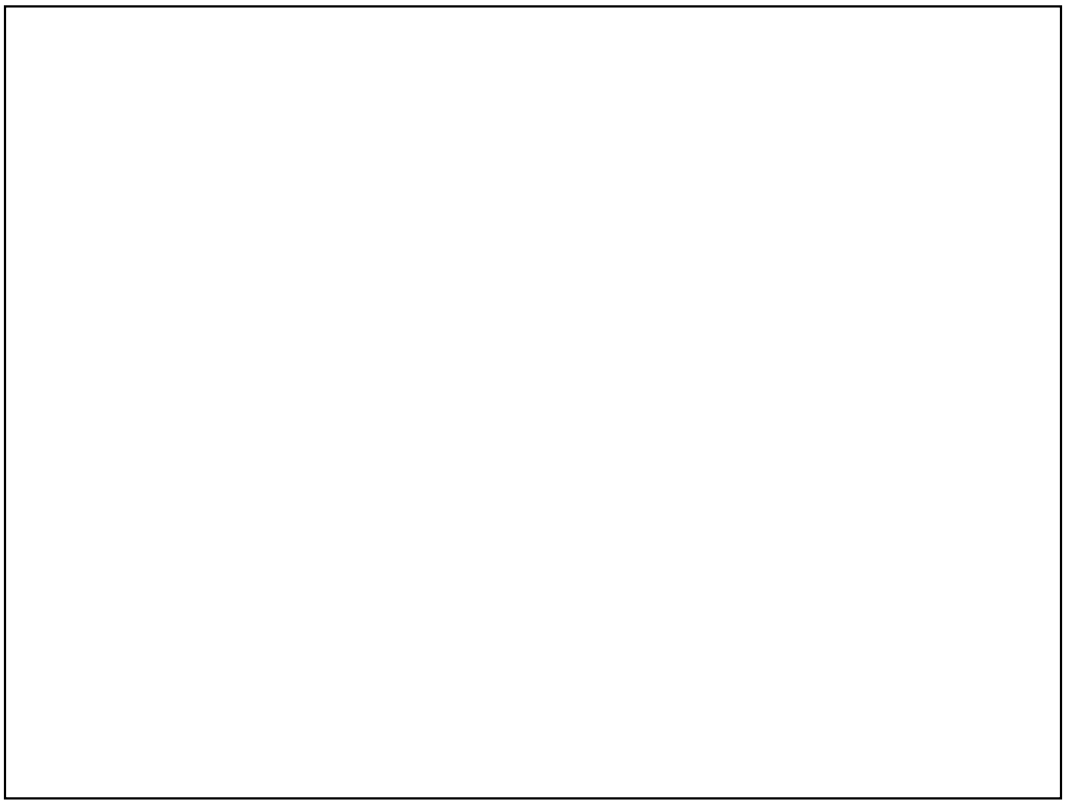
• **Integrity** (covers both data and system integrity)**:** Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity. A loss of integrity is the unauthorized modification or destruction of information.

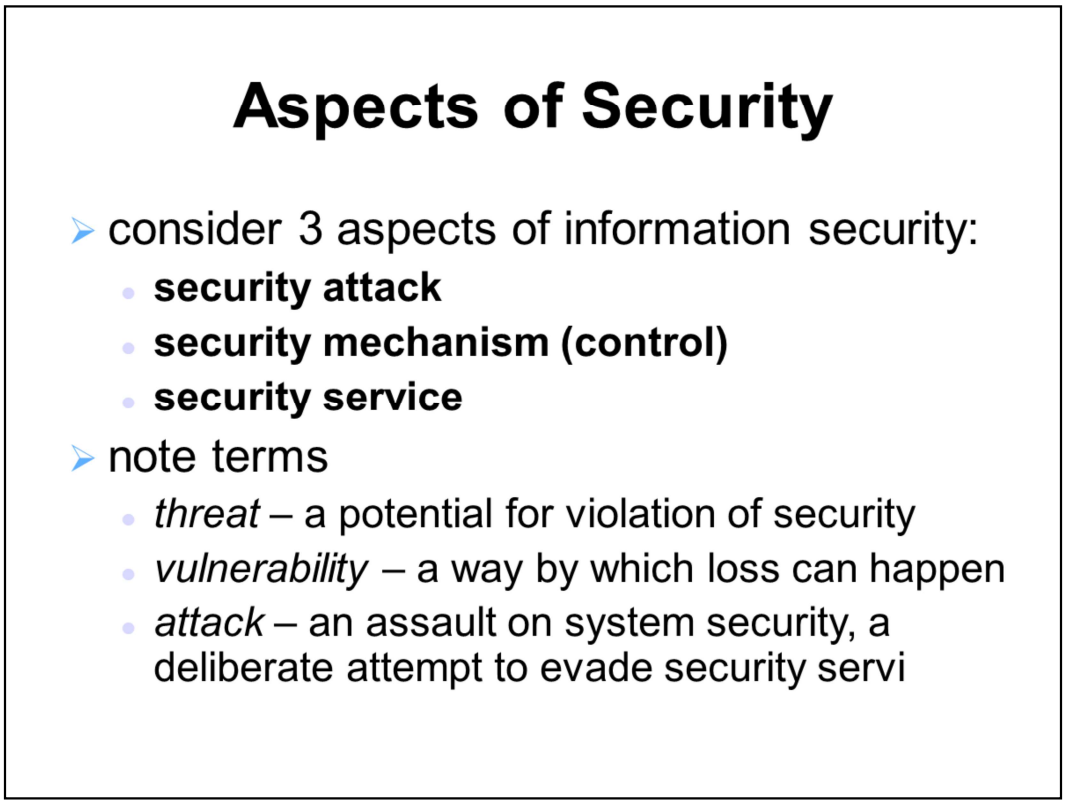
• **Availability:** Ensuring timely and reliable access to and use of information. A loss of availability is the disruption of access to or use of information or an information system.

Although the use of the CIA triad to define security objectives is well established, some in the security field feel that additional concepts are needed to present a complete picture. Two of the most commonly mentioned are:

• **Authenticity:** The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator.

• **Accountability:** The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity.





The OSI security architecture focuses on security attacks, mechanisms, and services. These can be defined briefly as follows:

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

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

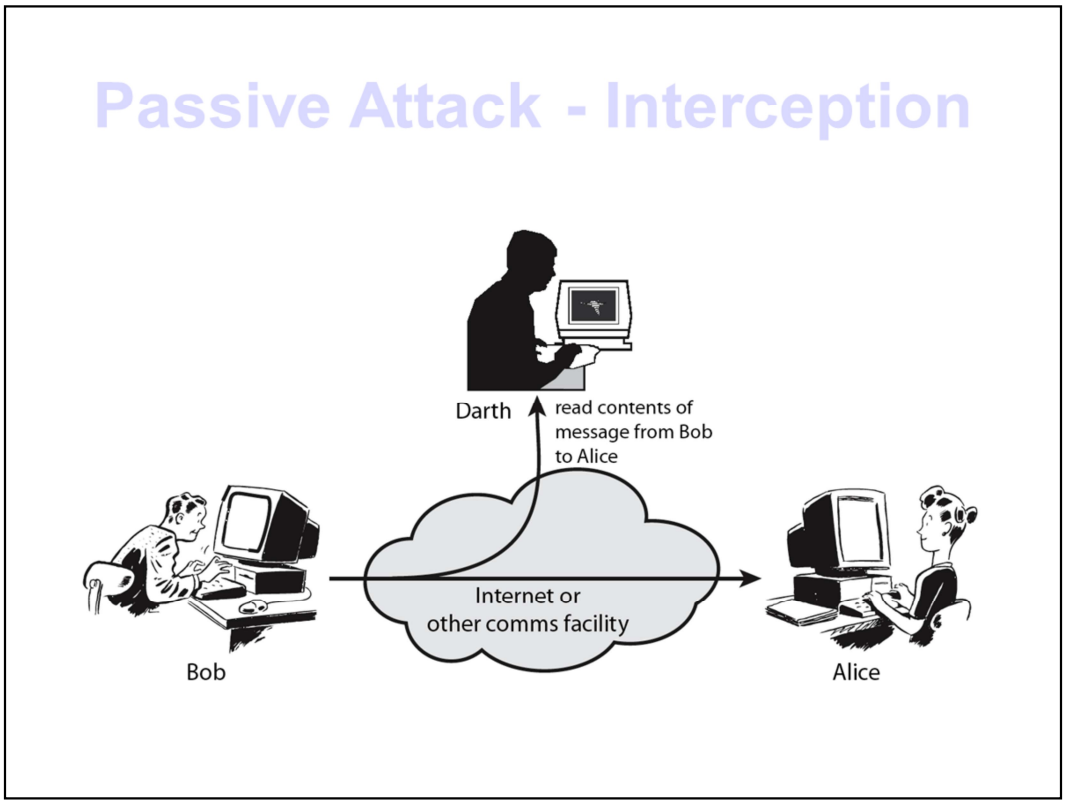
• **Security service**: A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.

In the literature, the terms *threat and attack* are commonly used to mean more or less the same thing. Table 1.1 provides definitions taken from RFC 2828, *Internet Security Glossary.*

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.



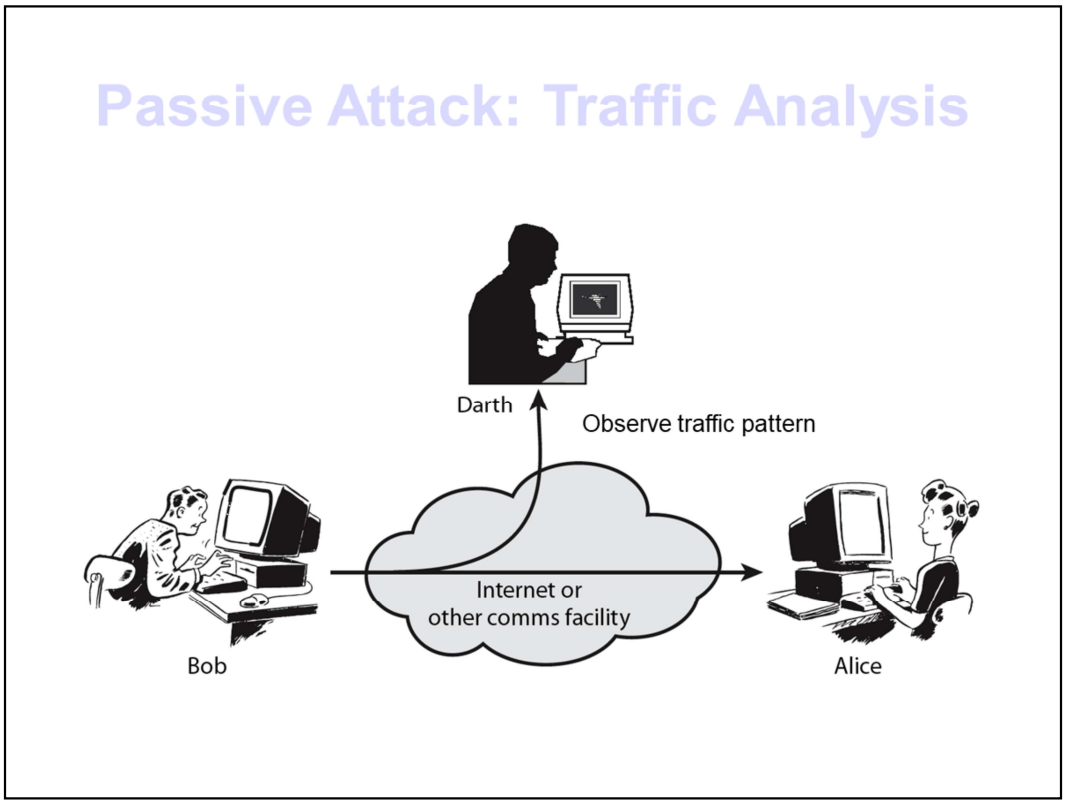
A useful means of classifying security attacks, used both in X.800 and RFC 2828, is in terms of *passive attacks* and *active attacks.* A passive attack attempts to learn or make use of information from the system but does not affect system resources.

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 - as shown above in Stallings Figure 1.2a here

+ traffic analysis - monitor traffic flow to determine location and identity of communicating hosts and could observe the frequency and length of messages being exchanged

These attacks are difficult to detect because they do not involve any alteration of the data.



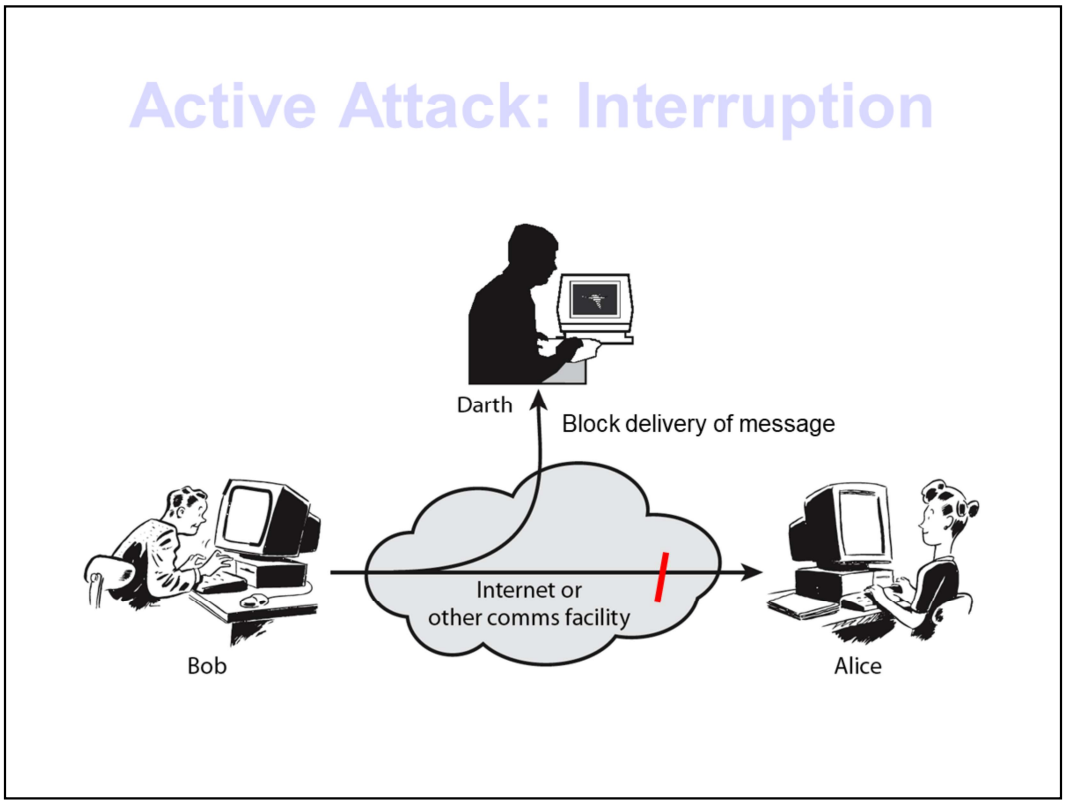
A useful means of classifying security attacks, used both in X.800 and RFC 2828, is in terms of *passive attacks* and *active attacks.* A passive attack attempts to learn or make use of information from the system but does not affect system resources.

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 - as shown above in Stallings Figure 1.2a here

+ traffic analysis - monitor traffic flow to determine location and identity of communicating hosts and could observe the frequency and length of messages being exchanged

These attacks are difficult to detect because they do not involve any alteration of the data.



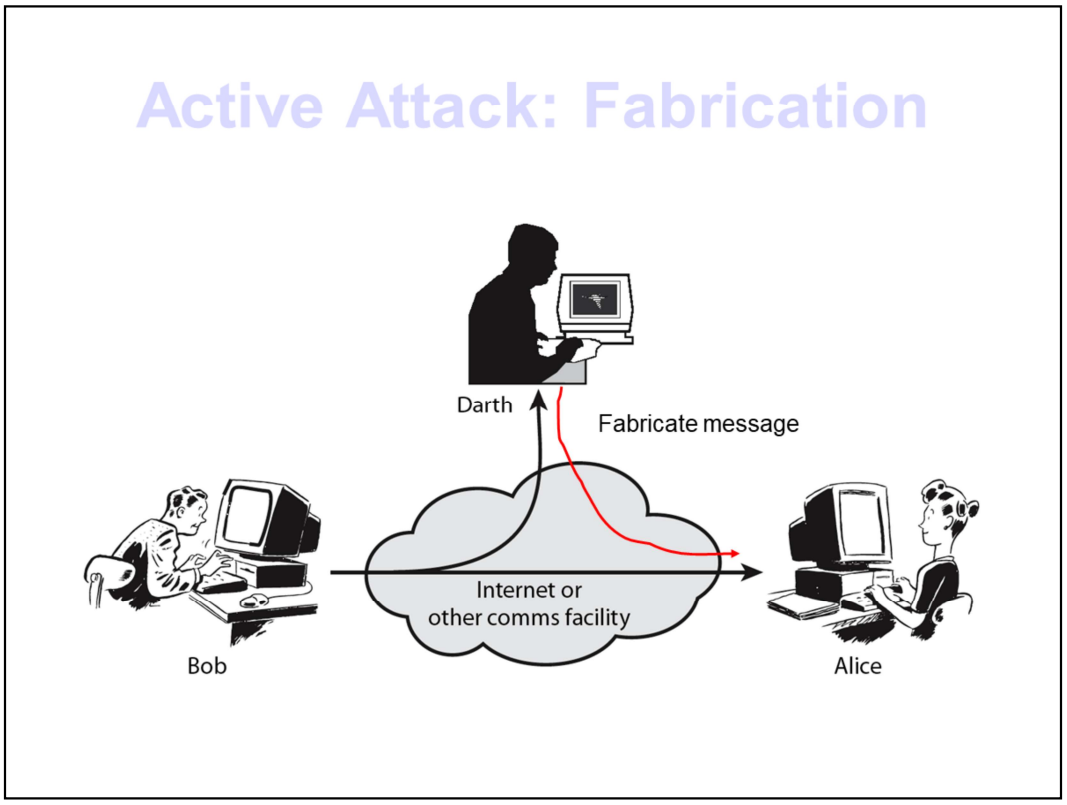
A useful means of classifying security attacks, used both in X.800 and RFC 2828, is in terms of *passive attacks* and *active attacks.* A passive attack attempts to learn or make use of information from the system but does not affect system resources.

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 - as shown above in Stallings Figure 1.2a here

+ traffic analysis - monitor traffic flow to determine location and identity of communicating hosts and could observe the frequency and length of messages being exchanged

These attacks are difficult to detect because they do not involve any alteration of the data.



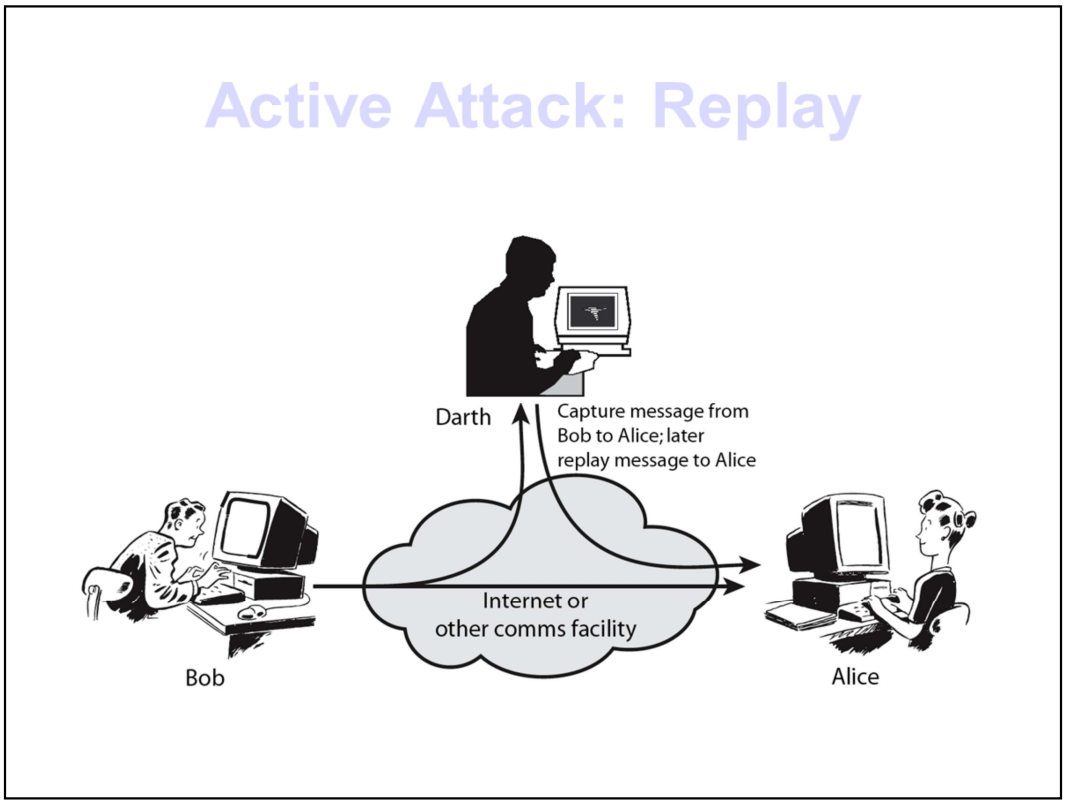
A useful means of classifying security attacks, used both in X.800 and RFC 2828, is in terms of *passive attacks* and *active attacks.* A passive attack attempts to learn or make use of information from the system but does not affect system resources.

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 - as shown above in Stallings Figure 1.2a here

+ traffic analysis - monitor traffic flow to determine location and identity of communicating hosts and could observe the frequency and length of messages being exchanged

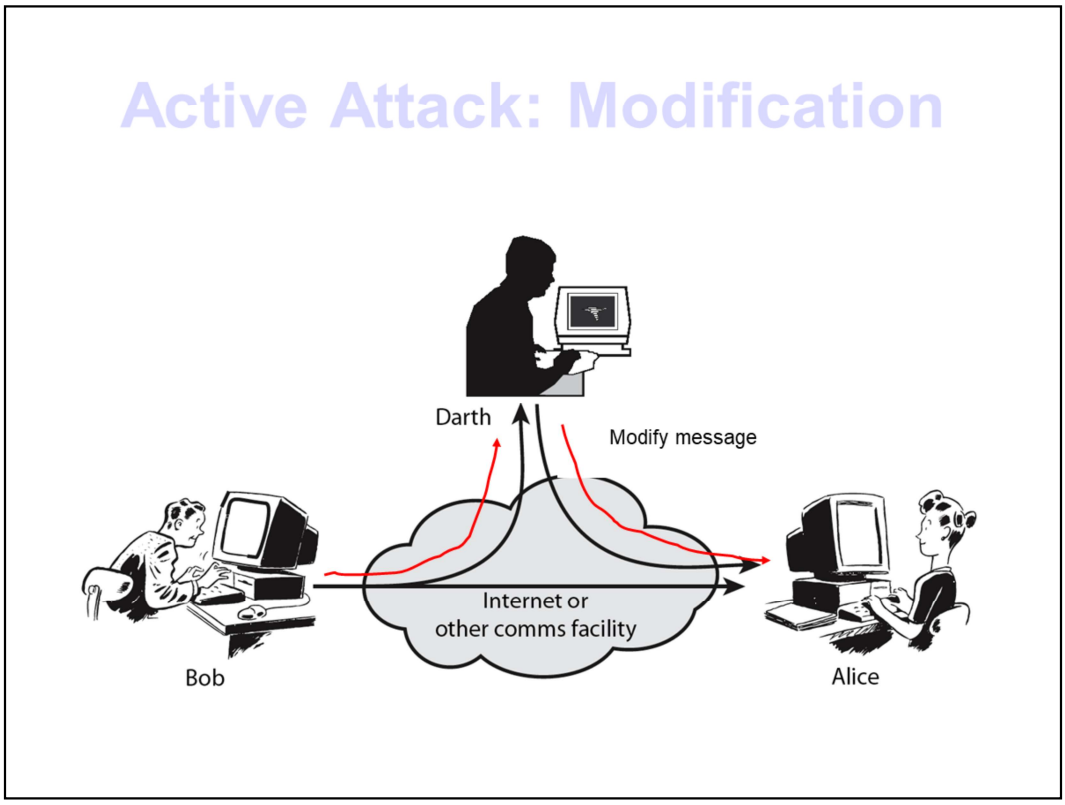
These attacks are difficult to detect because they do not involve any alteration of the data.



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 of one entity as some other
- replay previous messages (as shown above in Stallings Figure 1.3b)
- modify/alter (part of) messages in transit to produce an unauthorized effect
- denial of service - prevents or inhibits the normal use or management of communications facilities

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.



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 of one entity as some other
- replay previous messages (as shown above in Stallings Figure 1.3b)
- modify/alter (part of) messages in transit to produce an unauthorized effect
- denial of service - prevents or inhibits the normal use or management of communications facilities

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.

Symmetric Encryption

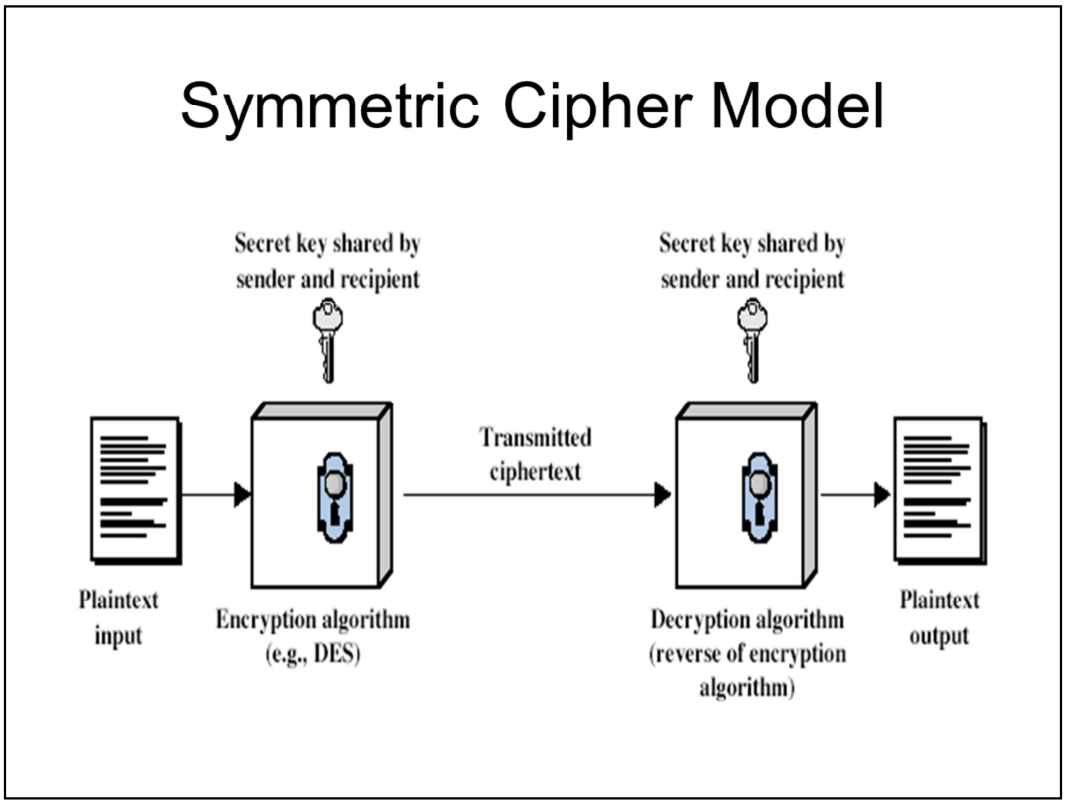
- or conventional / private-key / single-key
- sender and recipient share a common key
- all classical encryption algorithms are private-key
- was only type prior to invention of public-key in 1970's

*All traditional schemes are **symmetric** / **single key** / **private-key** encryption algorithms, with a **single key**, used for both encryption and decryption, since both sender and receiver are equivalent, either can encrypt or decrypt messages using that common key.*

Basic Terminology

- **plaintext** - the original message
- **ciphertext** - the coded message
- **cipher** - algorithm for transforming plaintext to ciphertext
- **key** - info used in cipher known only to sender/receiver
- **encipher (encrypt)** - converting plaintext to ciphertext
- **decipher (decrypt)** - recovering ciphertext from plaintext
- **cryptography** - study of encryption principles/methods
- **cryptanalysis (codebreaking)** - the study of principles/ methods of deciphering ciphertext *without* knowing key
- **cryptology** - the field of both cryptography and cryptanalysis

Briefly review some terminology used throughout the course.



Detail 5 ingredients of the symmetric cipher model:

-plaintext

-encryption algorithm – performs substitutions/transformations on plaintext

-secret key – control exact substitutions/transformations used in encryption algorithm

-ciphertext

-decryption algorithm – inverse of encryption algorithm

Requirements

- two requirements for secure use of symmetric encryption:
  - a strong encryption algorithm
  - a secret key known only to sender / receiver

    $Y = E_K(X)$

    $X = D_K(Y)$
- assume encryption algorithm is known
- implies a secure channel to distribute key

Generally assume that the algorithm is known.

This allows easy distribution of s/w and h/w implementations.

Hence assume just keeping key secret is sufficient to secure encrypted messages.

Have plaintext X, ciphertext Y, key K, encryption alg Ek, decryption alg Dk.

Cryptography

- can characterize by:
  - type of encryption operations used
    - substitution / transposition / product
  - number of keys used
    - single-key or private / two-key or public
  - way in which plaintext is processed
    - block / stream

Types of Cryptanalytic Attacks

- **ciphertext only**
  - only know algorithm / ciphertext, statistical, can identify plaintext
- **known plaintext**
  - know/suspect plaintext & ciphertext to attack cipher
- **chosen plaintext**
  - select plaintext and obtain ciphertext to attack cipher
- **chosen ciphertext**
  - select ciphertext and obtain plaintext to attack cipher
- **chosen text**
  - select either plaintext or ciphertext to en/decrypt to attack cipher

Brute Force Search

- always possible to simply try every key
- most basic attack, proportional to key size
- assume either know / recognise plaintext

Key Size (bits)	Number of Alternative Keys	Time required at 1 encryption/μs	Time required at 10^6 encryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\ \mu s = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\ \mu s = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\ \mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\ \mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\ \mu s = 6.4 \times 10^{12}$ years	6.4×10^{6} years

More Definitions

- **unconditional security**
  - no matter how much computer power is available, the cipher cannot be broken since the ciphertext provides insufficient information to uniquely determine the corresponding plaintext
- **computational security**
  - given limited computing resources (eg time needed for calculations is greater than age of universe), the cipher cannot be broken

Unconditional security would be nice, but the only known such cipher is the **one-time pad** (later). For all reasonable encryption algorithms, have to assume computational security where it either takes too long, or is too expensive, to bother breaking the cipher.

Classical Substitution Ciphers

- where letters of plaintext are replaced by other letters or by numbers or symbols
- or if plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns

In this section and the next, we examine a sampling of what might be called classical encryption

techniques. A study of these techniques enables us to illustrate the basic approaches to

symmetric encryption used today and the types of cryptanalytic attacks that must be anticipated.

The two basic building blocks of all encryption techniques: substitution and transposition.

We examine these in the next two sections. Finally, we discuss a system that combine both

substitution and transposition.

Caesar Cipher

- earliest known substitution cipher
- by Julius Caesar
- first attested use in military affairs
- replaces each letter by 3rd letter on
- example:

```
meet me after the toga party
PHHW PH DIWHU WKH WRJD SDUWB
```

Substitution ciphers form the first of the fundamental building blocks. The core idea is to replace one basic unit (letter/byte) with another. Whilst the early Greeks described several substitution ciphers, the first attested use in military affairs of one was by Julius Caesar, described by him in *Gallic Wars* (cf. Kahn pp83-84). Still call any cipher using a simple letter shift a **caesar cipher**, not just those with shift 3.

Note: when letters are involved, the following conventions are used in this course: Plaintext is always in lowercase; ciphertext is in uppercase; key values are in italicized lowercase.

Caesar Cipher

- can define transformation as:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

- mathematically give each letter a number

```
a b c d e f g h i j k  l  m
0 1 2 3 4 5 6 7 8 9 10 11 12
n  o  p  q  r  s  t  u  v  w  x  y  Z
13 14 15 16 17 18 19 20 21 22 23 24 25
```

- then have Caesar cipher as:

  $C = E(p) = (p + k) \bmod (26)$

  $p = D(C) = (C - k) \bmod (26)$

This mathematical description uses **modulo arithmetic** (ie clock arithmetic). Here, when you reach Z you go back to A and start again. Mod 26 implies that when you reach 26, you use 0 instead (ie the letter after Z, or 25 + 1 goes to A or 0).

Example: howdy (7,14,22,3,24) encrypted using key *f* (5) is MTBID

Cryptanalysis of Caesar Cipher

- only have 26 possible ciphers
  - A maps to A,B,..Z
- could simply try each in turn
- a **brute force search**
- given ciphertext, just try all shifts of letters
- do need to recognize when have plaintext
- eg. break ciphertext "GCUA VQ DTGCM"

With a caesar cipher, there are only 26 possible keys, of which only 25 are of any use, since mapping A to A etc doesn't really obscure the message! cf. basic rule of cryptanalysis "check to ensure the cipher operator hasn't goofed and sent a plaintext message by mistake"!

Can try each of the keys (shifts) in turn, until can recognise the original message. See Stallings Fig 2.3 for example of search.

Note: as mentioned before, do need to be able to **recognise** when have an original message (ie is it English or whatever). Usually easy for humans, hard for computers. Though if using say compressed data could be much harder.

Example "GCUA VQ DTGCM" when broken gives "easy to break", with a shift of 2 (key C).

Monoalphabetic Cipher

- rather than just shifting the alphabet
- could shuffle (jumble) the letters arbitrarily
- each plaintext letter maps to a different random ciphertext letter
- hence key is 26 letters long

```
Plain:  abcdefghijklmnopqrstuvwxyz
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN
Plaintext:  ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA
```

Monoalphabetic Cipher Security

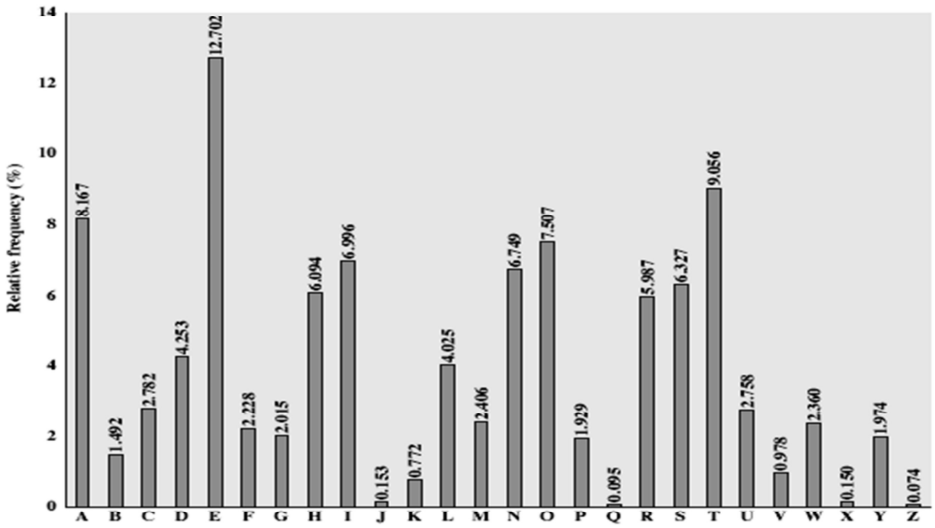
- now have a total of 26! keys
- with so many keys, might think is secure
- but would be **!!!WRONG!!!**
- problem is language characteristics

Language Redundancy and Cryptanalysis

- human languages are **redundant**
- eg "th lrd s m shphrd shll nt wnt"
- letters are not equally commonly used
- in English **e** is by far the most common letter
- then T,R,N,I,O,A,S
- other letters are fairly rare
- cf. Z,J,K,Q,X
- have tables of single, double & triple letter frequencies

As the example shows, we don't actually need all the letters in order to understand written English text. Here vowels were removed, but they're not the only redundancy. cf written Hebrew has no vowels for same reason. Are usually familiar with "party conversations", can hear one person speaking out of hubbub of many, again because of redundancy in aural language also. This redundancy is also the reason we can compress text files, the computer can derive a more compact encoding without losing any information. Basic idea is to count the relative frequencies of letters, and note the resulting pattern.

English Letter Frequencies



This graph is based on counts done at ADFA in the late 1980's, and used to develop the tables published in Seberry & Pieprzyk [SEBE89].

Note that all human languages have varying letter frequencies, though the number of letters and their frequencies varies.

Seberry & Pieprzyk [SEBE89] Appendix A has graphs for 20 languages (most European & Japanese & Malay).

Use in Cryptanalysis

- key concept - monoalphabetic substitution ciphers do not change relative letter frequencies
- discovered by Arabian scientists in 9th century
- calculate letter frequencies for ciphertext
- compare counts/plots against known values
- if Caesar cipher look for common peaks/troughs
  - peaks at: A-E-I triple, NO pair, RST triple
  - troughs at: JK, X-Z
- for monoalphabetic must identify each letter
  - tables of common double/triple letters help

The simplicity and strength of the monoalphabetic substitution cipher meant it dominated cryptographic use for the first millenium AD. It was broken by Arabic scientists. The earliest known description is in Abu al-Kindi's "A Manuscript on Deciphering Cryptographic Messages", published in the 9th century but only rediscovered in 1987 in Istanbul, but other later works also attest to their knowledge of the field.

Example Cryptanalysis

- given ciphertext:

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
```

- count relative letter frequencies (see text)
- guess P & Z are e and t
- guess ZW is th and hence ZWP is the
- proceeding with trial and error fially get:

```
it was disclosed yesterday that several informal but
direct contacts have been made with political
representatives of the viet cong in moscow
```

Polyalphabetic Ciphers

- another approach to improving security is to use multiple cipher alphabets
- called **polyalphabetic substitution ciphers**
- makes cryptanalysis harder with more alphabets to guess and flatter frequency distribution
- use a key to select which alphabet is used for each letter of the message
- use each alphabet in turn
- repeat from start after end of key is reached

One approach to reducing the "spikyness" of natural language text is used the Playfair cipher which encrypts more than one letter at once. We now consider the other alternative, using multiple cipher alphabets in turn. This gives the attacker more work, since many alphabets need to be guessed, and because the frequency distribution is more complex, since the same plaintext letter could be replaced by several ciphertext letters, depending on which alphabet is used.

Vigenère Cipher

- simplest polyalphabetic substitution cipher is the **Vigenère Cipher**
- effectively multiple caesar ciphers
- key is multiple letters long K = k1 k2 ... kd
- i^{th} letter specifies i^{th} alphabet to use
- use each alphabet in turn
- repeat from start after d letters in message
- decryption simply works in reverse

Simply create a set of caesar cipher translation alphabets, then use each in turn, as shown next.

Example

- write the plaintext out
- write the keyword repeated above it
- use each key letter as a caesar cipher key
- encrypt the corresponding plaintext letter
- eg using keyword *deceptive*

```
key:        deceptivedeceptivedeceptive
plaintext:  wearediscoveredsaveyourself
ciphertext:ZICVTWQNGRZGVTWAVZHCQYGLMGJ
```

One-Time Pad

- if a truly random key as long as the message is used, the cipher will be secure
- called a One-Time pad
- is unbreakable since ciphertext bears no statistical relationship to the plaintext
- since for **any plaintext** & **any ciphertext** there exists a key mapping one to other
- can only use the key **once** though
- have problem of safe distribution of key

The One-Time Pad is an evolution of the Vernham cipher, which was invented by Gilbert Vernham in 1918, and used a long tape of random letters to encrypt the message. An Army Signal Corp officer, Joseph Mauborgne, proposed an improvement using a random key that was truly as long as the message, with no repetitions, which thus totally obscures the original message.

Since any plaintext can be mapped to any ciphertext given some key, there is simply no way to determine which plaintext corresponds to a specific instance of ciphertext.

Transposition Ciphers

- now consider classical **transposition** or **permutation** ciphers
- these hide the message by rearranging the letter order
- without altering the actual letters used
- can recognise these since have the same frequency distribution as the original text

Transposition Ciphers form the second basic building block of ciphers. The core idea is to rearrange the order of basic units (letters/bytes/bits) without altering their actual values.

Row Transposition Ciphers

- a more complex scheme
- write letters of message out in rows over a specified number of columns
- then reorder the columns according to some key before reading off the rows

```
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
```

Product Ciphers

- ciphers using substitutions or transpositions are not secure because of language characteristics
- hence consider using several ciphers in succession to make harder, but:
  - two substitutions make a more complex substitution
  - two transpositions make more complex transposition
  - but a substitution followed by a transposition makes a new much harder cipher
- this is bridge from classical to modern ciphers