UNIT – I CONVENTIONAL AND MODERN ENCRYPTION


COMPUTER SECURITY CONCEPTS

Computer Security
The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information / data, and telecommunications)

Confidentiality
- Data confidentiality
  - Assures that private or confidential information is not made available or disclosed to unauthorized
- Privacy
  - Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

Integrity
- Data integrity
  - Assures that information and programs are changed only in a specified and authorized manner.
- System integrity
  - Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

Availability
- Assures that systems work promptly and service is not denied to authorized users.

CIA Triad

Confidentiality
- 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
- Guarding against improper information modification or destruction, including ensuring information nonrepudiation 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.

Authenticity
- The property of being genuine and being able to be verified and trusted

Accountability
- The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity
The OSI Security Architecture

- ITU-T Recommendation X.800, Security Architecture for OSI, defines such a systematic approach
- The OSI security architecture focuses on security attacks, mechanisms, and services.

Security attack

- Any action that compromises the security of information owned by an organization.

Security mechanism

- A process (or a device) 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.

Security Attacks

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

Passive Attacks

- in the nature of eavesdropping on, or monitoring of, transmissions.
- The goal is to obtain information that is being transmitted.
- very difficult to detect, because they do not involve any alteration of the data
- feasible to prevent the success of these attacks, usually by means of encryption
- emphasis in dealing with passive attacks is on prevention rather than detection

Two types of passive attacks

- Release of message contents
- Traffic analysis.

Release Of Message Contents

- A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information
- prevent an opponent from learning the contents of these transmissions

Traffic Analysis

- 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
Active Attacks

- Active attacks involve some modification of the data stream or the creation of a false stream
- detect and to recover from any disruption or delays caused by them
- can be subdivided into four categories:
  o masquerade,
  o replay,
  o modification of messages
  o denial of service

Masquerade

- one entity pretends to be a different entity
- usually includes one of the other forms of active attack

Example

- authentication sequences can be captured and replayed after a valid authentication sequence

Replay

- passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect

Modification Of Messages

- some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect

Example

- a message meaning “Allow John Smith to read confidential file accounts” is modified to mean “Allow Fred Brown to read confidential file accounts.”

Denial Of Service

- prevents or inhibits the normal use or management of communications facilities
- may have a specific target; for example, an entity may suppress all messages directed to a particular destination
- disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance
Security Services in X.800

- 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.
- RFC 2828, defines as 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.

X.800
- divides these services into five categories and fourteen specific services

Authentication
- The assurance that the communicating entity is the one that it claims to be
- Two types
  - Peer Entity Authentication
  - Data-Origin Authentication

Access control
- The prevention of unauthorized use of a resource

Data confidentiality
- The protection of data from unauthorized disclosure.
- Four Types
  - Connection Confidentiality
  - Connectionless Confidentiality
  - Selective-Field Confidentiality
  - Traffic-Flow Confidentiality

Data integrity
- The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).
- Five types
  - Connection Integrity with Recovery
  - Connection Integrity without Recovery
  - Selective-Field Connection Integrity
  - Connectionless Integrity
  - Selective-Field Connectionless Integrity

Nonrepudiation
- Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication
- Two types
  - Nonrepudiation, Origin
  - Nonrepudiation, Destination

Security Mechanisms in X.800.

- feature designed to detect, prevent, or recover from a security attack
- no single mechanism that will support all services required

Specific security mechanisms:
- those that are implemented in a specific protocol layer, such as TCP or an application-layer protocol
- encipherment, digital signatures, access controls, data integrity, authentication exchange, traffic padding, routing control, notarization

Pervasive security mechanisms:
- trusted functionality, security labels, event detection, security audit trails, security recovery
- those that are not specific to any particular protocol layer or security service
- 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.

All the techniques for providing security have two components:

- A security-related transformation on the information to be sent.
  - Examples: encryption of the message, addition of a code based on the contents.
- Some secret information shared by the two principals, unknown to the opponent.
  - Example: encryption key used in conjunction with the transformation.

A trusted third party may be needed to achieve secure transmission:

- for distributing the secret information to the two principals.
- to arbitrate disputes between the two principals concerning the authenticity of a message transmission.

Four basic tasks in designing a particular security service:

1. Design an algorithm for performing the security-related transformation 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.

Network Access Security Model
protecting an information system from unwanted access from hacker, intruder
hacker who, with no malign intent, simply gets satisfaction from breaking and entering a computer system.
intruder can be a disgruntled employee who wishes to do damage or a criminal who seeks to exploit computer assets for financial gain
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:
  Information access threats: Intercept or modify data on behalf of users who should not have access
  Service threats: Exploit service flaws in computers to inhibit use by legitimate users
  Examples: Viruses and worms, spread using disks & inserted over network

Classical Encryption Techniques

Symmetric Cipher Model
  Cryptanalysis and Brute-Force Attack
Substitution Techniques
  Caesar Cipher
  Monoalphabetic Ciphers
  Playfair Cipher
  Hill Cipher
  Polyalphabetic Ciphers
  One-Time Pad
Transposition Techniques
  Rotor Machines
Steganography

Introduction

Symmetric encryption is a form of cryptosystem in which encryption and decryption are performed using the same key. It is also known as conventional encryption.
Symmetric encryption transforms plaintext into ciphertext using a secret key and an encryption algorithm. Using the same key and a decryption algorithm, the plaintext is recovered from the ciphertext.
The two types of attack on an encryption algorithm are cryptanalysis, based on properties of the encryption algorithm, and brute-force, which involves trying all possible keys.
Traditional (precomputer) symmetric ciphers use substitution and/or transposition techniques. Substitution techniques map plaintext elements (characters, bits) into ciphertext elements. Transposition techniques systematically transpose the positions of plaintext elements.
Rotor machines are sophisticated precomputer hardware devices that use substitution techniques.
Steganography is a technique for hiding a secret message within a larger one in such a way that others cannot discern the presence or contents of the hidden message.
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 encryption 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.
Symmetric Cipher Model

A symmetric encryption scheme has five ingredients

- Plaintext
- Encryption algorithm
  - performs various substitutions and transformations
- Secret key
  - another input to the encryption algorithm
  - a value independent of the plaintext and of the algorithm
- Ciphertext
  - For a given message, two different keys will produce two different ciphertexts
- Decryption algorithm
  - encryption algorithm run in reverse

Simplified Model of Symmetric Encryption

Two requirements for secure use of conventional / symmetric encryption

- need a strong encryption algorithm
  - 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
- 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
  - do not need to keep the algorithm secret; we need to keep only the key secret
  - the principal security problem is maintaining the secrecy of the key

Model of Symmetric Cryptosystem

Plain Text: $X = [X_1, X_2, \ldots, X_M]$
Key: $K = [K_1, K_2, \ldots, K_J]$
Cipher text $Y = [Y_1, Y_2, \ldots, Y_N]$
$Y = E(K, X)$
$X = D(K, Y)$
Cryptanalysis and Brute-Force Attack

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.
- various types of cryptanalytic attacks based on the amount of information known to the cryptanalyst

<table>
<thead>
<tr>
<th>Type of Attack</th>
<th>Known to Cryptanalyst</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ciphertext Only</td>
<td>Encryption algorithm</td>
</tr>
<tr>
<td>Known Plaintext</td>
<td>Ciphertext</td>
</tr>
<tr>
<td>Chosen Plaintext</td>
<td>Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key</td>
</tr>
<tr>
<td>Chosen Ciphertext</td>
<td>Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key</td>
</tr>
<tr>
<td>Chosen Text</td>
<td></td>
</tr>
</tbody>
</table>

Two schemes

- **unconditionally secure**
  - if the ciphertext generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available

- **computationally secure**
  - meets either of the following criteria:
    - The cost of breaking the cipher exceeds the value of the encrypted information.
    - The time required to break the cipher exceeds the useful lifetime of the information.

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.

Cryptographic systems characterization

Three independent dimensions

- The type of operations used for transforming plaintext to ciphertext.
  - substitution
    - each element is mapped into another element
  - transposition
    - elements are rearranged
  - product systems, involve multiple stages of substitutions and transpositions
- 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 use different keys, the system is referred to as **asymmetric**, two-key, or public-key encryption
- 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
**Substitution Techniques**

- A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols.
- If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.

**Julius Caesar Cipher**

- replacing each letter of the alphabet with the letter standing three places further down the alphabet.
- alphabet is wrapped around, so that the letter following Z is A.

can define transformation as:

\[
\begin{align*}
\text{c} &= E(p) = (p + k) \mod (26) \\
\text{p} &= D(c) = (c - k) \mod (26)
\end{align*}
\]

**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.

**Monoalphabetic Ciphers**

- rather than just shifting the alphabet shuffle (jumble) the letters arbitrarily.
- each plaintext letter maps to a different random ciphertext letter.
- hence key is 26 letters long.
- the “cipher” line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than 4x10^{26} possible keys.
- This is 10 orders of magnitude greater than the key space for DES and would seem to eliminate brute-force techniques for cryptanalysis.
- Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet.
- A countermeasure is to provide multiple substitutes, known as **homophones**, for a single letter.
- For example, the letter e could be assigned a number of different cipher symbols, such as 16, 74, 35, and 21, with each homophone assigned to a letter in rotation or randomly.

**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.
- followed by T, R, N, I, O, A, S.
- other letters like Z, J, K, Q, X are fairly rare.
- have tables of single, double & triple letter frequencies for various languages.
- two-letter combinations, known as **digrams** (ex: th).
**Playfair Cipher**

- best-known multiple-letter encryption cipher
- treats digrams in the plaintext as single units and translates these units into ciphertext digrams

**Playfair Key Matrix**

- 5 × 5 matrix of letters constructed using a keyword
- filling in the letters of the keyword (minus duplicates) from left to right and from top to bottom,
- filling in the remainder matrix with the remaining letters in alphabetic order.
- The letters I and J count as one letter
- Example matrix using the keyword MONARCHY

<table>
<thead>
<tr>
<th>M</th>
<th>O</th>
<th>N</th>
<th>A</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>H</td>
<td>Y</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>E</td>
<td>F</td>
<td>G</td>
<td>I</td>
<td>J</td>
</tr>
<tr>
<td>L</td>
<td>P</td>
<td>O</td>
<td>S</td>
<td>T</td>
</tr>
<tr>
<td>U</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Z</td>
</tr>
</tbody>
</table>

**Plaintext is encrypted two letters at a time, according to the following rules**

- Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x,
  - Ex: balloon would be treated as ba lx lo on.
- Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last.
  - Ex: ar is encrypted as RM.
- Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last.
  - Ex: mu is encrypted as CM.
- Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter.
  - Ex: hs becomes BP and ea becomes IM (or JM, as the encipherer wishes)

**Example**

Given the key MONARCHY apply Play fair cipher to plain text “FACTIONALISM”

**Solution**

(p) FA CT IO NA LI SM

(c) IO DL FA AR SE LA

(d) FA CT IO NA LI SM

**Security of Playfair Cipher**

- security much improved over monoalphabetic since have 26 × 26 = 676 digrams
- would need a 676 entry frequency table to analyse and correspondingly more ciphertext
- was widely used for many years eg. by US & British military in WW1
- it can be broken, given a few hundred letters since still has much of plaintext structure

**Hill Cipher**

Finding the inverse of a matrix

\[
A^{-1} = \left[ \begin{array}{cc} a & b \\ c & d \end{array} \right]^{-1} = \frac{1}{\det(A)} \left[ \begin{array}{cc} d & -b \\ -c & a \end{array} \right] = \frac{1}{ad - bc} \left[ \begin{array}{cc} d & -b \\ -c & a \end{array} \right].
\]

**3x3 Matrix**

\[
\begin{pmatrix}
k_{11} & k_{12} & k_{13} \\
k_{21} & k_{22} & k_{23} \\
k_{31} & k_{32} & k_{33}
\end{pmatrix}
\]

\[
|A| = k_{11}k_{22}k_{33} - k_{11}k_{23}k_{32} - k_{12}k_{21}k_{33} + k_{12}k_{23}k_{31} + k_{13}k_{21}k_{32} - k_{13}k_{23}k_{31}
\]
Example:

\[
\text{det } \begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix} = (5 \times 3) - (8 \times 17) = -121 \mod 26 = 9
\]

We can show that \(9^{-1} \mod 26 = 3\), because \(9 \times 3 = 27 \mod 26 = 1\) (see Chapter 4 or Appendix E). Therefore, we compute the inverse of \(A\) as

\[
A^{-1} \mod 26 = \begin{pmatrix} 3 & -8 \\ -17 & 5 \end{pmatrix} = \begin{pmatrix} 3 & 18 \\ 9 & 5 \end{pmatrix} = \begin{pmatrix} 9 & 54 \\ 27 & 15 \end{pmatrix} = \begin{pmatrix} 9 & 2 \\ 15 & 15 \end{pmatrix}
\]

The Hill algorithm

- This encryption algorithm takes \(m\) successive plaintext letters and substitutes for them \(m\) ciphertext letters.
- The substitution is determined by \(m\) linear equations in which each character is assigned a numerical value \((a = 0, b = 1, \ldots, z = 25)\).
- For \(m = 3\), the system can be described as

\[
c_1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \mod 26
\]
\[
c_2 = (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \mod 26
\]
\[
c_3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \mod 26
\]

This can be expressed in terms of row vectors and matrices:

\[
\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \mod 26
\]

or

\[
C = PK \mod 26
\]

- where \(C\) and \(P\) are row vectors of length 3 representing the plaintext and ciphertext, and \(K\) is a 3x3 matrix representing the encryption key.
- Operations are performed \(\mod 26\).
- In general terms, the Hill system can be expressed as

\[
C = E(K, P) = PK \mod 26
\]
\[
P = D(K, C) = CK^{-1} \mod 26 = PKK^{-1} = P
\]

Example

Encrypt the message “meet me at the usual place at ten rather than eight oclock” using the Hill cipher with the key \(\begin{pmatrix} 9 \\ 4 \\ 7 \end{pmatrix}\). Show your calculations and the result. Show the calculations for the corresponding decryption of the ciphertext to recover the original plaintext.

1) mathematically give each letter a number

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

2) 1st pair from plain text \(\text{“me”} \Rightarrow \begin{pmatrix} 12 \\ 5 \end{pmatrix}\)

\[
\begin{pmatrix} 9 \\ 4 \end{pmatrix} \begin{pmatrix} 12 \\ 5 \end{pmatrix} = \begin{pmatrix} 9x12 + 4x4 \\ 5x12 + 7x4 \end{pmatrix} = \begin{pmatrix} 124 \\ 88 \end{pmatrix} \Rightarrow mod 26 \Rightarrow \begin{pmatrix} 20 \\ 10 \end{pmatrix} \Rightarrow \begin{pmatrix} u \\ k \end{pmatrix}
\]

3) 2nd pair from plain text \(\text{“et”}\)
4) Cipher text for “meet” is “ukix”

5) To get plain text from cipher text, we need to find the inverse of K

6) \(|A| = (9x7 - 5x4) = 43\)

7) \(\text{Adj}(A) \Rightarrow (\begin{vmatrix} 7 & -4 \\ -5 & 9 \end{vmatrix}) \Rightarrow \frac{1}{43} (\begin{vmatrix} 7 & -4 \\ -5 & 9 \end{vmatrix}) = \frac{1}{17} \begin{vmatrix} 7 & -4 \\ -5 & 9 \end{vmatrix} (\because 43 \% 26 = 17)\)

8) Find the multiplier for 17, using \(17 \times X = 1 \mod 26 \Rightarrow X = 23\)

9) \(\begin{vmatrix} 161 & -92 \\ -115 & 207 \end{vmatrix} \Rightarrow \mod 26 \Rightarrow \begin{vmatrix} 5 & -14 \\ -11 & 25 \end{vmatrix} \Rightarrow \begin{vmatrix} 5 & 12 \\ 15 & 25 \end{vmatrix} (\because \text{Add 26 for negative values})\)

10) \(P = CK^{-1} \Rightarrow \text{For the cipher text of “uk”,} \)

\[
\begin{vmatrix} 5 & 12 \\ 15 & 25 \end{vmatrix} \begin{vmatrix} 20 \\ 10 \end{vmatrix} = \begin{vmatrix} 5x20 + 12x10 \\ 15x20 + 25x10 \end{vmatrix} \Rightarrow \begin{vmatrix} 220 \\ 550 \end{vmatrix} \mod 26 \Rightarrow \begin{vmatrix} 12 \\ 4 \end{vmatrix} = \begin{vmatrix} m \\ e \end{vmatrix}
\]

Hence the plain text is “me”

### Polyalphabetic Ciphers

- use different monoalphabetic substitutions as one proceeds through the plaintext message.
- improve security using multiple cipher alphabets
- make cryptanalysis harder with more alphabets to guess and flatter frequency distribution
- general name for this approach is **polyalphabetic substitution cipher**

has the following features in common:
- A set of related monoalphabetic substitution rules is used.
- A key determines which particular rule is chosen for a given transformation.

### One-Time Pad

- improvement to the Vernam cipher that yields the ultimate in security
- using a random key that is as long as the message, so that the key need not be repeated
- the key is to be used to encrypt and decrypt a single message, and then is discarded.
- Each new message requires a new key of the same length as the new message

**Example**

ciphertext: ANKYODKYUREPFJBYOJDSPREYIUNOFDOUIERFPLUYTS
key: pxlmvmsydofuyrvzwctnlebncvudpahfzzlmnyih
plaintext: mr mustard with the candlestick in the hall

ciphertext: ANKYODKYUREPFJBYOJDSPREYIUNOFDOUIERFPLUYTS
key: mfugpmiydgaxoufhkllmhqsddqgtewbqfgyovuhwt
plaintext: miss scarlet with the knife in the library

two fundamental difficulties
- problem of making large quantities of random keys
- problem of key distribution and protection

### Transposition Techniques

A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters

### Rail Fence Technique

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:

```
mematrhtgpy
etefteteoaat
```

The encrypted message is

```
MEMATRHTGPRYETEFEAOAT
```

**Pure Transposition Cipher**

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.

**Example**

Key: 4 3 1 2 5 6 7

Plaintext: atack postpone
dontilt
woamxyz

Ciphertext: TTNAAPTMTSUOADWCOIKNXLYPETZ

**Double Transposition**

performing more than one stage of transposition

**Example**

if the foregoing message is reencrypted using the same algorithm

Key: 4 3 1 2 5 6 7

Input:

```
ttnaapt
mtsuoao
dwcoixk
```

Output: NSCYAUOPTTWLTMDNAIEPATXZ

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

**Rotor Machines (Skip)**

The machine consists of a set of independently rotating cylinders through which electrical pulses can flow.

Each cylinder has 26 input pins and 26 output pins, with internal wiring that connects each input pin to a unique output pin.

**Steganography**

A plaintext message may be hidden in one of two ways.

- The methods of steganography conceal the existence of the message
- The methods of cryptography render the message unintelligible to outsiders
  - by various transformations of the text

Various ways to conceal the message

arrangement of words or letters within an apparently innocuous text spells out the real message
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.

Invisible ink
A number of substances can be used for writing but leave no visible trace until heat or some chemical is applied

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, the results of typing with the correction tape are visible only under a strong light.

hiding a message by using the least significant bits of frames on a CD
- the Kodak Photo CD format's maximum resolution is 2048 by 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
- Thus you can hide a 2.3-megabyte message in a single digital snapshot

Number of drawbacks
- lot of overhead to hide a relatively few bits of information
- once the system is discovered, it becomes virtually worthless
- the insertion method depends on some sort of key
  - Alternatively, a message can be first encrypted and then hidden using steganography

Advantage of steganography
- can be employed by parties who have something to lose should the fact of their secret communication be discovered
- Encryption flags traffic as important or secret or may identify the sender or receiver as someone with something to hide

Block Ciphers
- A block cipher is an encryption/decryption scheme in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.
- A stream cipher is one that encrypts a digital data stream one bit or one byte at a time.
  - Examples of classical stream ciphers are the autokeyed Vigenère cipher and the Vernam cipher.
- A block cipher is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length
- Many block ciphers have a Feistel structure. Such a structure consists of a number of identical rounds of processing. In each round, a substitution is performed on one half of the data being processed, followed by a permutation that interchanges the two halves. The original key is expanded so that a different key is used for each round.
- The Data Encryption Standard (DES) has been the most widely used encryption algorithm until recently. It exhibits the classic Feistel structure. DES uses a 64-bit block and a 56-bit key.
- Two important methods of cryptanalysis are differential cryptanalysis and linear cryptanalysis. DES has been shown to be highly resistant to these two types of attack

Diffusion and Confusion
Shannon suggests two methods for frustrating statistical cryptanalysis: diffusion and confusion.

Diffusion
- the statistical structure of the plaintext is dissipated into long-range statistics of the ciphertext.
- This is achieved by having each plaintext digit affect the value of many ciphertext digits;
- generally this is equivalent to having each ciphertext digit be affected by many plaintext digits

Confusion
• seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible, again to thwart attempts to discover the key.
• Thus, even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key.

Feistel Cipher Structure

- All rounds have the same structure.
- A substitution is performed on the left half of the data.
- This is done by applying a round function \( F \) to the right half of the data and then taking the exclusive-OR of the output of that function and the left half of the data.
- The round function has the same general structure and parameterized by the round subkey \( K_i \)
- Following this, a permutation is performed that consists of the interchange of the two halves of the data
- This structure is a particular form of the substitution-permutation network (SPN)

Feistel network depends on the choice of the following parameters and design features

Block size, Key size, Number of rounds, Subkey generation algorithm, Round function, Fast software encryption/decryption, Ease of analysis

Feistel Encryption and Decryption
Simplified DES

- educational rather than a secure encryption algorithm.
- It has similar properties and structure to DES with much smaller parameters

Simplified DES Scheme

- The S-DES encryption algorithm takes an 8-bit block of plaintext (example: 10111101) and a 10-bit key as input and produces an 8-bit block of ciphertext as output.
- The S-DES decryption algorithm takes an 8-bit block of ciphertext and the same 10-bit key used to produce that ciphertext as input and produces the original 8-bit block of plaintext.

Involves five functions:

- an initial permutation (IP);
- a complex function labeled $f_K$, which involves both permutation and substitution operations and depends on a key input;
- a simple permutation function that switches (SW) the two halves of the data;
- the function $f_K$ again;
- finally a permutation function that is the inverse of the initial permutation ($IP^{-1}$).

Algorithm

\[
\begin{align*}
\text{ciphertext} & = IP^{-1}(f_K_2(SW(f_K_1(IP(\text{plaintext})))))) \\
\text{plaintext} & = IP^{-1}(f_K_1(SW(f_K_2(IP(\text{ciphertext}))))))
\end{align*}
\]

The 8-bit subkey $K_1 = (k_{11}, k_{12}, k_{13}, k_{14}, k_{15}, k_{16}, k_{17}, k_{18})$ is added

rename these 8 bits
Key Generation for Simplified DES

\[ S0 = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 \\ 2 & 0 & 2 & 1 \\ 3 & 3 & 1 & 3 \end{bmatrix} \quad S1 = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \]

Simplified DES Encryption Detail

<table>
<thead>
<tr>
<th>Plain Text</th>
<th>1 2 3 4 5 6 7 8 9 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>1 1 0 0 0 1 1 1 1 0</td>
</tr>
<tr>
<td>P10</td>
<td>3 5 2 7 4 10 1 9 8 6</td>
</tr>
<tr>
<td>LS-1</td>
<td>0 0 1 1 0 0 1 1 1 1</td>
</tr>
<tr>
<td>P8 (K1)</td>
<td>0 1 1 0 0 1 1 1 1 0</td>
</tr>
<tr>
<td>LS-2</td>
<td>6 3 7 4 8 5 10 9</td>
</tr>
<tr>
<td>P8</td>
<td>1 1 1 0 1 0 0 1</td>
</tr>
</tbody>
</table>
The Data Encryption Standard (DES)

- Overview
- DES Encryption
  - General Depiction of DES Encryption Algorithm
  - Initial Permutation
  - Permutation Tables for DES
  - Details of Single Round
  - Calculation of F(R, K)
  - Definition of DES S-Boxes
  - Key Generation
- DES Decryption
- The Avalanche Effect

Overview
- Data are encrypted in 64-bit blocks using a 56-bit key.
- The algorithm transforms 64-bit input in a series of steps into a 64-bit output.
- The same steps, with the same key, are used to reverse the encryption

DES Encryption
- There are two inputs to the encryption function: the plaintext to be encrypted and the key.
- In this case, the plaintext must be 64 bits in length and the key is 56 bits in length

General Depiction of DES Encryption Algorithm
- Processing of the plaintext proceeds in three phases.
- First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input.
- This is followed by a phase consisting of 16 rounds of the same function, which involves both permutation and substitution functions.
  - The output of the sixteenth round consists of 64 bits that are a function of the input plaintext and the key.
  - The left and right halves of the output are swapped to produce the preoutput.
Finally, the preoutput is passed through a permutation (IP-1) that is the inverse of the initial permutation function, to produce the 64-bit ciphertext.

**Permutation Tables for DES**

**Initial Permutation**
- The initial permutation and its inverse are defined by tables.
- The tables are to be interpreted as follows.
- The input to a table consists of 64 bits numbered from 1 to 64.
- The 64 entries in the permutation table contain a permutation of the numbers from 1 to 64.
- Each entry in the permutation table indicates the position of a numbered input bit in the output, which also consists of 64 bits.

### Initial Permutation (IP)

| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2  |
| 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4  |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6  |
| 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8  |
| 57 | 49 | 41 | 33 | 25 | 17 |  9 | 1  |
| 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3  |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5  |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7  |

### Inverse Initial Permutation (IP⁻¹)

| 40 |  7 |  8 | 48 | 16 |  56 |  24 |  64 |  32 |
| 39 |  7 | 47 | 15 |  55 |  23 |  63 |  31 |
| 38 |  6 | 46 | 14 |  54 |  22 |  62 |  30 |
| 37 |  5 | 45 | 13 |  53 |  21 |  61 |  29 |
| 36 |  4 | 44 | 12 |  52 |  20 |  60 |  28 |
| 35 |  3 | 43 | 11 |  51 |  19 |  59 |  27 |
| 34 |  2 | 42 | 10 |  50 |  18 |  58 |  26 |
| 33 |  1 | 41 |  9 |  49 |  17 |  57 |  25 |

### Expansion Permutation (E)

| 32 | 1  | 2  | 3  | 4  | 5  |
|  4 | 5  | 6  | 7  | 8  | 9  |
|  8 | 9  |10  |11  |12  |13  |
| 12 |13  |14  |15  |16  |17  |
| 16 |17  |18  |19  |20  |21  |
| 20 |21  |22  |23  |24  |25  |
| 24 |25  |26  |27  |28  |29  |
| 28 |29  |30  |31  |32  | 1  |

### Permutation Function (P)

| 16 | 7  | 20 | 21 | 29 | 12  | 28  | 17  |
|  1 |15  |23  |26  | 5  | 18  | 31  | 10  |
|  2 | 8  |24  |32  |27  | 3   |  9  |    |
| 19 |13  |30  | 6  | 22  |11  | 4   |  25 |
Details of Single Round

Calculation of $F(R, K)$

Definition of DES S-Boxes ($S_1$ .. $S_8$)
Key Generation
- A 64-bit key is used as input to the algorithm.
- The bits of the key are numbered from 1 through 64; every eighth bit is ignored.
- The key is first subjected to a permutation governed by a table labeled Permuted Choice One.
- The resulting 56-bit key is then treated as two 28-bit quantities, labeled C0 and D0.
- At each round, C\textsubscript{i-1} and D\textsubscript{i-1} are separately subjected to a circular left shift, or rotation, of 1 or 2 bits.
- These shifted values serve as input to the next round.
- They also serve as input to Permuted Choice Two, which produces a 48-bit output that serves as input to the function F(R\textsubscript{i-1}, K\textsubscript{i}).

DES Decryption
As with any Feistel cipher, decryption uses the same algorithm as encryption, except that the application of the subkeys is reversed.

The Avalanche Effect
- A small change in either the plaintext or the key should produce a significant change in the ciphertext.
- A change in one bit of the plaintext or one bit of the key should produce a change in many bits of the ciphertext.
- DES exhibits a strong avalanche effect.

Example
Two plaintexts that differ by one bit were used:

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

With the key:
0000001 1001011 0100100 1100010 0011100 0011000 0011100 0110010

- After just three rounds, 21 bits differ between the two blocks.
- On completion, the two ciphertexts differ in 34 bit positions.

Similar test in which a single plaintext is input with two keys that differ in only one bit position:

01101000 10000101 00101111 01111010 0010011 01110110 11101011 10100100

Keys:
1110010 1111011 1101111 0011000 0011101 0000100 0110001 11011100
0110010 1111011 1101111 0011000 0011101 0000100 0110001 11011100

About half of the bits in the ciphertext differ, and that the avalanche effect is pronounced after just a few rounds.

Differential and Linear Cryptanalysis

Differential Cryptanalysis
- A powerful method to analyse block ciphers.
- A statistical attack against Feistel ciphers.
- Uses cipher structure not previously used.
- The analysis compares differences between two related encryptions, and looks for a known difference in leading to a known difference out with some (pretty small but still significant) probability.
- If a number of such differences are determined, it is feasible to determine the subkey used in the function f.
- Compares two related pairs of encryptions with
  - Known difference in the input and
  - Searching for a known difference in output when same subkeys are used.
In differential cryptanalysis, we start with two messages, \( m \) and \( m' \), with a known XOR difference \( \Delta m = m \oplus m' \), and consider the difference between the intermediate message halves: \( \Delta m_i = m_i \oplus m'_i \). Then we have

\[
\Delta m_{i+1} = m_{i+1} \oplus m'_{i+1} = [m_{i-1} \oplus f(m_i, K_i)] \oplus [m'_{i-1} \oplus f(m'_i, K_i)] = \Delta m_{i-1} \oplus [f(m_i, K_i) \oplus f(m'_i, K_i)]
\]

difference. Therefore, if we know \( \Delta m_{i-1} \) and \( \Delta m_i \) with high probability, then we know \( \Delta m_{i+1} \) with high probability. Furthermore, if a number of such differences are determined, it is feasible to determine the subkey used in the function \( f \).

begin with two plaintext messages \( m \) and \( m' \)

with a given difference and trace through a probable pattern of differences after each round to yield a probable difference for the ciphertext. Actually, there are two probable patterns of differences for the two 32-bit halves: \((\Delta m_{i-1} || \Delta m_i)\). Next, we submit \( m \) and \( m' \) for encryption to determine the actual difference under the unknown key and compare the result to the probable difference. If there is a match, \( E(K, m) \oplus E(K, m') = (\Delta m_{i-1} || \Delta m_i) \)
then we suspect that all the probable patterns at all the intermediate rounds are correct. With that assumption, we can make some deductions about the key bits. This procedure must be repeated many times to determine all the key bits.

**Differential Propagation through Three Rounds of DES**

after three rounds, the probability that the output difference is as shown is equal to \(0.25 \times 1 \times 0.25 = 0.0625\)
Linear Cryptanalysis

- This attack is based on finding linear approximations to describe the transformations performed in DES
- This method can find a DES key given $2^{43}$ known plaintexts, as compared to $2^{47}$ chosen plaintexts for differential cryptanalysis
- it may be easier to acquire known plaintext rather than chosen plaintext
- infeasible as an attack on DES
- For a cipher with $n$ bit plaintext and ciphertext blocks and an $m$-bit key, let the plaintext block be labeled $P[1], \ldots, P[n]$, the ciphertext block $C[1], \ldots, C[n]$, and the key $K[1], \ldots, K[m]$.

Then define $A[i; j, \ldots, k] = A[i] \oplus A[j] \oplus \ldots \oplus A[k]$

The objective of linear cryptanalysis is to find an effective linear equation of the form

$$P[\alpha_1, \alpha_2, \ldots, \alpha_d] \oplus C[\beta_1, \beta_2, \ldots, \beta_b] = K[\gamma_1, \gamma_2, \ldots, \gamma_c]$$

(where $x = 0$ or $1$; $1 \leq a; b \leq n; c \leq m$; and where the $\alpha$, $\beta$, and $\gamma$ terms represent fixed, unique bit locations) that holds with probability $p \neq 0.5$. The further $p$ is from 0.5, the more effective the equation.

Once a proposed relation is determined, the procedure is to compute the results of the lefthand side of the preceding equation for a large number of plaintext-ciphertext pairs.

If the result is 0 more than half the time, assume $K[\gamma_1, \gamma_2, \ldots, \gamma_c] = 0$.

If it is 1 most of the time, assume $K[\gamma_1, \gamma_2, \ldots, \gamma_c] = 1$. This gives us a linear equation on the key bits.

Modes of operation

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
<th>Typical Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic Codebook (ECB)</td>
<td>Each block of 64 plaintext bits is encoded independently using the same key</td>
<td>Secure transmission of single values (e.g., an encryption key)</td>
</tr>
<tr>
<td>Cipher Block Chaining (CBC)</td>
<td>The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of ciphertext</td>
<td>General-purpose block-oriented transmission Authentication</td>
</tr>
<tr>
<td>Cipher Feedback (CFB)</td>
<td>Input is processed $j$ bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext</td>
<td>General-purpose stream-oriented transmission Authentication</td>
</tr>
<tr>
<td>Output Feedback (OFB)</td>
<td>Similar to CFB, except that the input to the encryption algorithm is the preceding DES output.</td>
<td>Stream-oriented transmission over noisy channel (e.g., satellite communication)</td>
</tr>
<tr>
<td>Counter (CTR)</td>
<td>Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block</td>
<td>General-purpose block-oriented transmission Useful for high-speed requirements</td>
</tr>
</tbody>
</table>

Various Modes

1. Electronic Codebook Mode
2. Cipher Block Chaining Mode
3. Cipher Feedback Mode
4. Output Feedback Mode
5. Counter Mode
Electronic Codebook Mode

Encryption

\[
\begin{align*}
\text{Time } = 1 & : & P_1 & \xrightarrow{K} & \text{Encrypt} & \xrightarrow{C_1} & K \\
\text{Time } = 2 & : & P_2 & \xrightarrow{K} & \text{Encrypt} & \xrightarrow{C_2} & K \\
\text{Time } = N & : & P_N & \xrightarrow{K} & \text{Encrypt} & \xrightarrow{C_N} & K \\
\end{align*}
\]

Decryption

\[
\begin{align*}
\text{Time } = 1 & : & C_1 & \xrightarrow{K} & \text{Decrypt} & \xrightarrow{P_1} & K \\
\text{Time } = 2 & : & C_2 & \xrightarrow{K} & \text{Decrypt} & \xrightarrow{P_2} & K \\
\text{Time } = N & : & C_N & \xrightarrow{K} & \text{Decrypt} & \xrightarrow{P_N} & K \\
\end{align*}
\]

Cipher Block Chaining Mode

Encryption

\[
\begin{align*}
\text{Time } = 1 & : & \text{IV} & \xrightarrow{K} & \text{Encrypt} & \xrightarrow{C_1} & K \\
\text{Time } = 2 & : & P_2 & \xrightarrow{K} & \text{Encrypt} & \xrightarrow{C_2} & \text{Encrypt} \\
\text{Time } = N & : & P_N & \xrightarrow{K} & \text{Encrypt} & \xrightarrow{C_N} & \text{Encrypt} \\
\end{align*}
\]

Decryption

\[
\begin{align*}
\text{Time } = 1 & : & C_1 & \xrightarrow{K} & \text{Decrypt} & \xrightarrow{IV} & \text{Decrypt} \\
\text{Time } = 2 & : & C_2 & \xrightarrow{K} & \text{Decrypt} & \xrightarrow{C_{N-1}} & \text{Decrypt} \\
\text{Time } = N & : & C_N & \xrightarrow{K} & \text{Decrypt} & \xrightarrow{C_{N-1}} & \text{Decrypt} \\
\end{align*}
\]
Cipher Feedback Mode – Encryption / Decryption

Output Feedback Mode – Encryption / Decryption
Counter Mode

Encryption

Counter

\[ K \rightarrow \text{Encrypt} \]

\[ P_1 \rightarrow C_1 \]

\[ K \rightarrow \text{Encrypt} \]

\[ P_2 \rightarrow C_2 \]

\[ \ldots \rightarrow \ldots \]

\[ K \rightarrow \text{Encrypt} \]

\[ C_N \rightarrow P_N \]

Decryption

Counter

\[ K \rightarrow \text{Encrypt} \]

\[ C_1 \rightarrow P_1 \]

\[ K \rightarrow \text{Encrypt} \]

\[ C_2 \rightarrow P_2 \]

\[ \ldots \rightarrow \ldots \]

\[ K \rightarrow \text{Encrypt} \]

\[ C_N \rightarrow P_N \]

- IV: initialization vector
- plaintext (padded as necessary) consists of a sequence of $b$-bit blocks, $P_1, P_2, \ldots, P_N$;
- the corresponding sequence of ciphertext blocks is $C_1, C_2, \ldots, C_N$.
- the unit of transmission is $s$ bits; a common value is $s = 8$

Advantages Of CTR Mode

- Hardware efficiency
- Software efficiency
- Preprocessing
- Random access
- Provable security
- Simplicity

Encryption Algorithms

Advanced Encryption Standard

- The AES Cipher
- AES Parameters
- AES Encryption and Decryption
- AES Data Structures
- AES Encryption Round
- Substitute Bytes Transformation
- ShiftRows Transformation
- AddRoundKey Transformation
- AES Key Expansion
The AES Cipher

- The Rijndael proposal for AES defined a cipher in which the block length and the key length can be independently specified to be 128, 192, or 256 bits.
- The AES specification uses the same three key size alternatives but limits the block length to 128 bits.
- A number of AES parameters depend on the key length.
- In the description of this section, we assume a key length of 128 bits, which is likely to be the one most commonly implemented.

AES Parameters

<table>
<thead>
<tr>
<th>Key Size (words/bytes/bits)</th>
<th>4/16/128</th>
<th>6/24/192</th>
<th>8/32/256</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plaintext Block Size</td>
<td>4/16/128</td>
<td>4/16/128</td>
<td>4/16/128</td>
</tr>
<tr>
<td>Number of Rounds</td>
<td>10</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>Round Key Size</td>
<td>4/16/128</td>
<td>4/16/128</td>
<td>4/16/128</td>
</tr>
<tr>
<td>Expanded Key Size</td>
<td>44/176</td>
<td>52/208</td>
<td>60/240</td>
</tr>
</tbody>
</table>

AES Encryption and Decryption
AES Data Structures

Input, state array, and output

Key and expanded key

AES Encryption Round

- Substitute Bytes Transformation
- ShiftRows Transformation
- AddRoundKey Transformation
- AES Key Expansion
Double DES

- has two encryption stages and two keys
- Given a plaintext P and two encryption keys K1 and K2 and , ciphertext C is generated as C = E(K2, E(K1, P))
- Decryption requires that the keys be applied in reverse order P = D(K1, D(K2, C))
- this scheme apparently involves a key length of 56 * 2 = 112 bits, resulting in a dramatic increase in cryptographic strength

Meet-In-The-Cryptographic Attack

- It is based on the observation that, if we have C = E(K2, E(K1, P)) then X = E(K1, P) = D(K2, C)
- Given a known pair, (P, C) the attack proceeds as follows
- First, encrypt P for all 2^{56} possible values of K1
- Store these results in a table and then sort the table by the values of X
- Next, decrypt C using all 2^{56} possible values of K2
- As each decryption is produced, check the result against the table for a match.
- If a match occurs, then test the two resulting keys against a new known plaintext–ciphertext pair.
- If the two keys produce the correct ciphertext, accept them as the correct keys.
- For any given plaintext P, there are 2^{56} possible ciphertext values that could be produced by double DES
- the foregoing procedure will produce about 2^{48} false alarms on the first (P, C) pair.
- With an additional 64 bits of known plaintext and ciphertext, the false alarm rate is reduced to 2^{48-64} = 2^{-16}.
- If the meet-in-the-middle attack is performed on two blocks of known plaintext–ciphertext, the probability that the correct keys are determined is 1 – 2^{-16}.
- The result is that a known plaintext attack will succeed against double DES, which has a key size of 112 bits, with an effort on the order of 2^{56}, which is not much more than the 2^{55} required for single DES

Triple DES

- triple encryption method that uses only two keys
- The function follows an encrypt-decrypt-encrypt (EDE) sequence
- C = E(K1, D(K2, E(K1, P)))
- There is no cryptographic significance to the use of decryption for the second stage.
- advantage is that it allows users of 3DES to decrypt data encrypted by users of the older single DES:
  - C = E(K1, D(K1, E(K1, P))) = E(K1, P)

Attacks on TDES

Known-Plaintext Attack on Triple DES

Triple DES with Three Keys

- Three-key 3DES has an effective key length of 168 bits and is defined as follows:
  - C = E(K3, D(K2, E(K1, P)))
- Backward compatibility with DES is provided by putting K3 = K2 or K1 = K2.
- A number of Internet-based applications have adopted three-key 3DES, including PGP and S/MIME
Blowfish

- Symmetric block cipher that can be effectively used for encryption and safeguarding of data
- It takes a variable-length key, from 32 bits to 448 bits, making it ideal for securing data.
- fast, free alternative to existing encryption algorithms
- unpatented and license-free, and is available free for all uses
- **Blowfish Algorithm** is a Feistel Network, iterating a simple encryption function 16 times.
- The block size is 64 bits, and the key can be any length up to 448 bits.
- Although there is a complex initialization phase required before any encryption can take place, the actual encryption of data is very efficient on large microprocessors.
- Blowfish is a variable-length key block cipher.
- It is suitable for applications where the key does not change often, like a communications link or an automatic file encryptor.
- It is significantly faster than most encryption algorithms when implemented on 32-bit microprocessors with large data caches

**Feistel Networks**

- A Feistel network is a general method of transforming any function (usually called an F function) into a permutation.
- It was invented by Horst Feistel and has been used in many block cipher designs.
- The working of a Feistal Network is given below:
  - Split each block into halves
  - Right half becomes new left half
  - New right half is the final result when the left half is XOR’d with the result of applying f to the right half and the key.
  - Note that previous rounds can be derived even if the function f is not invertible

**The Blowfish Algorithm:**

- Manipulates data in large blocks
- Has a 64-bit block size.
- Has a scalable key, from 32 bits to at least 256 bits.
- Uses simple operations that are efficient on microprocessors.
  - e.g., exclusive-or, addition, table lookup, modular-multiplication.
  - It does not use variable-length shifts or bit-wise permutations, or conditional jumps.
- Employs precomputable subkeys.
  - On large-memory systems, these subkeys can be precomputed for faster operation.
  - Not precomputing the subkeys will result in slower operation, but it should still be possible to encrypt data without any precomputations.
- Consists of a variable number of iterations.
- Uses subkeys that are a one-way hash of the key.
  - This allows the use of long passphrases for the key without compromising security.
- Has no linear structures that reduce the complexity of exhaustive search.
- Uses a design that is simple to understand.

**Description Of The Algorithm**

- Blowfish is a variable-length key, 64-bit block cipher.
- The algorithm consists of two parts:
  - a key-expansion part and
  - a data-encryption part.
- Key expansion converts a key of at most 448 bits into several subkey arrays totaling 4168 bytes.
- Data encryption occurs via a 16-round Feistel network.
- Each round consists of a key-dependent permutation, and a key- and data-dependent substitution.
- All operations are XORs and additions on 32-bit words.
The only additional operations are four indexed array data lookups per round.

**Subkeys**
- Blowfish uses a large number of subkeys.
- These keys must be precomputed before any data encryption or decryption.
- The P-array consists of 18 32-bit subkeys: P1, P2, ..., P18.
- There are four 32-bit S-boxes with 256 entries each:
  - S1,0, S1,1, ..., S1,255;
  - S2,0, S2,1, ..., S2,255;
  - S3,0, S3,1, ..., S3,255;
  - S4,0, S4,1, ..., S4,255.

**Encryption**
- Blowfish has 16 rounds.
- The input is a 64-bit data element, x.
- Divide x into two 32-bit halves: xL, xR.
- Then,
  
  \[
  \text{for } i = 1 \text{ to } 16: \\
  xL = xL \text{ XOR } P_i \\
  xR = F(xL) \text{ XOR } xR \\
  \text{Swap } xL \text{ and } xR
  \]
  
  - After the sixteenth round, swap xL and xR again to undo the last swap.
  - Then, xR = xR XOR P17 and xL = xL XOR P18.
  - Finally, recombine xL and xR to get the ciphertext.

**Decryption**
- Exactly the same as encryption, except that P1, P2, ..., P18 are used in the reverse order.

**Generating the Subkeys**

The subkeys are calculated using the Blowfish algorithm:

1. Initialize first the P-array and then the four S-boxes, in order, with a fixed string.
   - This string consists of the hexadecimal digits of pi (less the initial 3): P1 = 0x243f6a88, P2 = 0x85a308d3, P3 = 0x13198a2e, P4 = 0x03707344, etc.
2. XOR P1 with the first 32 bits of the key, XOR P2 with the second 32-bits of the key, and so on for all bits of the key (possibly up to P14). Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits. (For every short key, there is at least one equivalent longer key; for example, if A is a 64-bit key, then AA, AAA, etc., are equivalent keys.)
3. Encrypt the all-zero string with the Blowfish algorithm, using the subkeys described in steps (1) and (2).
4. Replace P1 and P2 with the output of step (3).
5. Encrypt the output of step (3) using the Blowfish algorithm with the modified subkeys.
6. Replace P3 and P4 with the output of step (5).
7. Continue the process, replacing all entries of the P array, and then all four S-boxes in order, with the output of the continuously changing Blowfish algorithm.

- In total, 521 iterations are required to generate all required subkeys.
- Applications can store the subkeys rather than execute this derivation process multiple times.

**Example:**

TBD
CAST128

History
- Cast-128 is a block cipher algorithm used in a lot of products like GPG and PGP.
- It was approved for the Canadian Government use by the Communication Security Establishment.
- It was created in 1996 by Carlisle Adams and Stafford Tavares using the CAST design.
- The CAST name is based on the initials of its inventors, though Bruce Schneier reports the authors’ claim that "the name should conjure up images of randomness".

Purpose
- General purpose in the internet community wherever a strong cryptography freely-algorithm is needed.
- Although Entrust holds a patent on the CAST design procedure, CAST-128 is available worldwide on a royalty-free basis for commercial and non-commercial uses.

Algorithm
- CAST-128 is a 12- or 16-round Feistel network with a 64-bit block size and a key size of between 40 to 128 bits (but only in 8-bit increments).
- The full 16 rounds are used when the key size is longer than 80 bits.
- Components include large 8×32-bit S-boxes based on bent functions, key-dependent rotations, modular addition and subtraction, and XOR operations.
- There are three alternating types of round function(image on the right), but they are similar in structure and differ only in the choice of the exact operation (addition, subtraction or XOR) at various points.
- Also its based in the Feistel Cipher structure

Advantages
- Because its based in the Feistel Cipher structure it has the advantage that encryption and decryption operations are very similar, even identical in some cases, requiring only a reversal of the key schedule.
- Therefore the size of the code or circuitry required to implement such a cipher is nearly halved.
- Is available worldwide on a royalty-free basis for commercial and non-commercial uses.
RC5

Introduction
- a proprietary cipher owned by RSADSI
- designed by Ronald Rivest (of RSA fame)
- used in various RSADSI products
- can vary key size / data size / no rounds
- very clean and simple design
- easy implementation on various CPUs
- yet still regarded as secure

RC5 Ciphers
- RC5 is a family of ciphers RC5-w/r/b
  - w = word size in bits (16/32/64) nb data=2w
  - r = number of rounds (0..255)
  - b = number of bytes in key (0..255)
- nominal version is RC5-32/12/16
  - 32-bit words so encrypts 64-bit data blocks
  - using 12 rounds
  - with 16 bytes (128-bit) secret key

RC5 Key Expansion
- RC5 uses 2r+2 subkey words (w-bits)
- subkeys are stored in array S[i], i=0..t-1
- the key schedule consists of
  - initializing S to a fixed pseudorandom value, based on constants e and phi
  - the byte key is copied (little-endian) into a c-word array L
  - a mixing operation then combines L and S to form the final S array

RC5 Encryption
- split input into two halves A & B
- L0 = A + S[0];
- R0 = B + S[1];
- for i = 1 to r do
  - Li = ((Li-1 XOR Ri-1) <<< Ri-1) + S[2 x i];
  - Ri = ((Ri-1 XOR Li) <<< Li) + S[2 x i + 1];
- each round is like 2 DES rounds
- note rotation is main source of non-linearity
- need reasonable number of rounds (eg 12-16)

RC5 Modes
- RFC2040 defines 4 modes used by RC5
- RC5 Block Cipher, is ECB mode
- RC5-CBC, is CBC mode
- RC5-CBC-PAD, is CBC with padding by bytes with value being the number of padding bytes
- RC5-CTS, a variant of CBC which is the same size as the original message, uses ciphertext stealing to keep size same as original
Traffic Confidentiality

Points of Vulnerability

- LAN or on other LANs in the same building that are interconnected with bridges and routers
  - eavesdropping by another employee
- The wiring closet
- An attack can take place on any of the communications links
  - cable (telephone twisted pair, coaxial cable, or optical fiber), microwave links, or satellite channels.
- various processors along the path are themselves subject to attack
  - attempts to modify the hardware or software, to gain access to the memory of the processor

Link versus End-to-End Encryption
<table>
<thead>
<tr>
<th>Link Encryption</th>
<th>End-to-End Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Security within End Systems and Intermediate Systems</strong></td>
<td></td>
</tr>
<tr>
<td>Message exposed in sending host</td>
<td>Message encrypted in sending host</td>
</tr>
<tr>
<td>Message exposed in intermediate nodes</td>
<td>Message encrypted in intermediate nodes</td>
</tr>
<tr>
<td><strong>Role of User</strong></td>
<td></td>
</tr>
<tr>
<td>Applied by sending host</td>
<td>Applied by sending process</td>
</tr>
<tr>
<td>Transparent to user</td>
<td>User applies encryption</td>
</tr>
<tr>
<td>Host maintains encryption facility</td>
<td>User must determine algorithm</td>
</tr>
<tr>
<td>One facility for all users</td>
<td>Users selects encryption scheme</td>
</tr>
<tr>
<td>Can be done in hardware</td>
<td>Software implementation</td>
</tr>
<tr>
<td>All or no messages encrypted</td>
<td>User chooses to encrypt, or not, for each message</td>
</tr>
<tr>
<td><strong>Implementation Concerns</strong></td>
<td></td>
</tr>
<tr>
<td>Requires one key per (host-intermediate node) pair and (intermediate node-intermediate node) pair</td>
<td>Requires one key per user pair</td>
</tr>
<tr>
<td>Provides host authentication</td>
<td>Provides user authentication</td>
</tr>
</tbody>
</table>