Module 1

Secure Communication

Introduction

Security: Requires measures to protect data during transmission

Definitions

- **Computer Security** - generic name for the collection of tools designed to protect data.
- **Network Security** - measures to protect data during their transmission in a network.
- **Internet Security** - measures to protect data during their transmission over a collection of interconnected networks.

Examples of security violations

1. User A transmits a file to B which contains sensitive information. User C ,who is not authorized to read the file, takes a copy of it.
2. Manager D transmits a data to a computer E (to update an authorization file). User F intercepts the message and forward to E.
3. In example 2, rather than intercept the message, F creates a new message and sends to E.

4. An employee is terminated from a office. The message from the manager is intercepted by the employee. He delays the message and collect many sensitive information from the office.

Aim

- **Internet Security:** consists of measures to prevent, detect, and correct security violations that involve the transmission of information.

What do we need for a secure communication channel?

- Authentication (Who am I talking to?)
- Confidentiality (Is my data hidden?)
- Integrity (Has my data been modified?)
- Availability (Can I reach the destination?)

Three Key Objectives

- Confidentiality
  - Data confidentiality
  - Privacy
- Integrity
  - Data integrity
  - System integrity
- Availability
- Additional concepts
  - Authenticity
  - Accountability

Examples of Security Requirements

Confidentiality

1. Student grade information
- highly important by students
- available to students, their parents, and employees that require the information .

2. Student enrollment information
- moderate confidentiality rating.

3. Directory information (lists of students or faculty or departmental)
- a low confidentiality rating or indeed no rating.
- freely available to the public and published on a school's Web site.

- **Integrity**

1. Patient's allergy information stored in a database.
- Information should be correct and current.
- high requirement for integrity.
- Inaccurate information could result in serious harm or death to a patient

- **Availability**

1. An interruption of service results in the inability for customers to access computing resources and staff to access the resources they need to perform critical tasks.
2. The loss of the service translates into a large financial loss ie employee productivity loss and potential customer loss.

OSI Security Architecture

- To assess effectively the security needs of an organization and to evaluate security products and policies, a systematic way of defining the requirements for security is needed.
- X.800 “Security Architecture for OSI”
- Defines a systematic way of defining and providing security requirements

- The OSI security architecture focuses on

1. Security attacks
2. Security mechanisms
3. Security services.

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

Threat and Attack

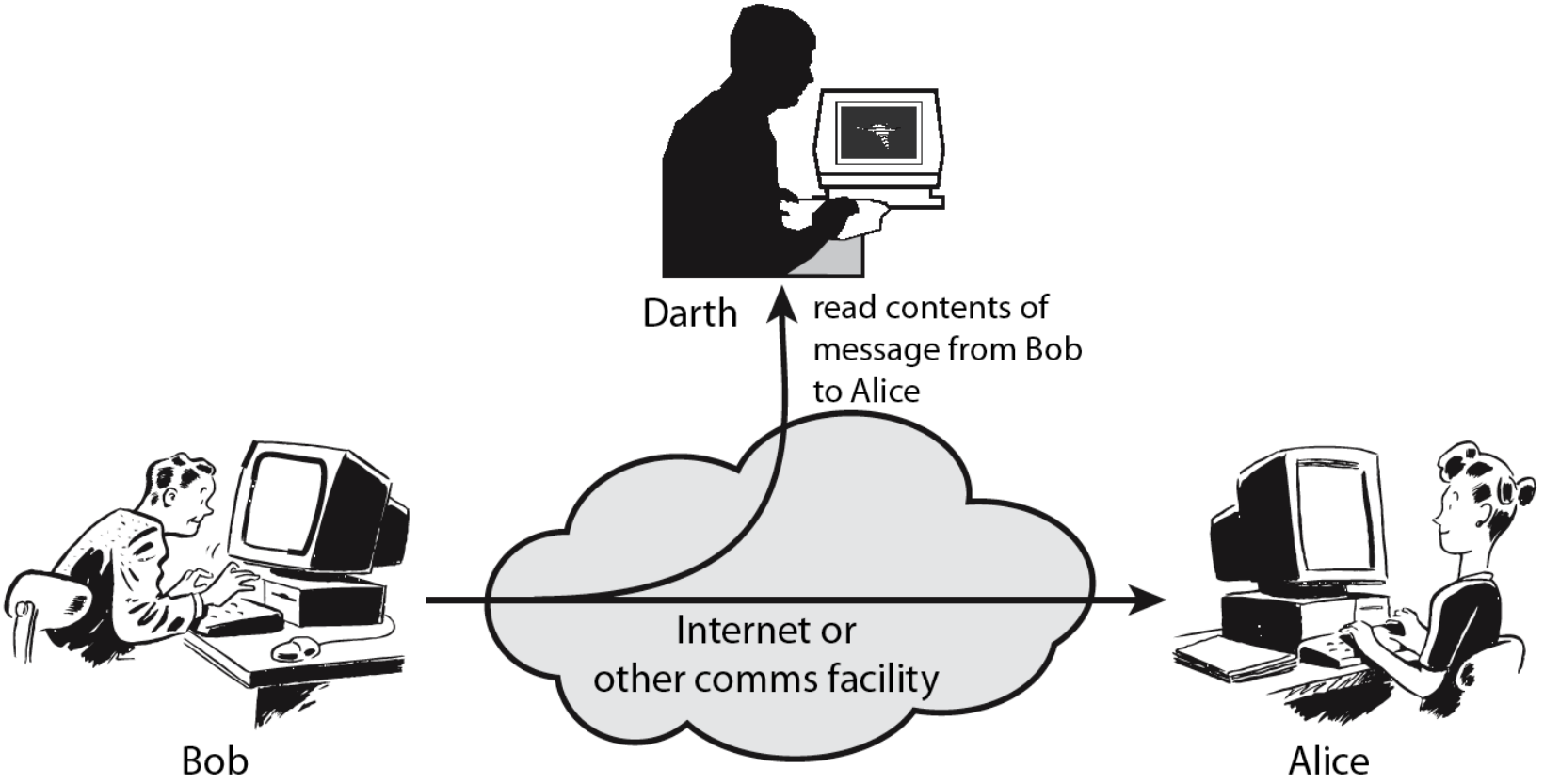
- **Threat -** A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. A threat is a possible danger that might exploit the vulnerability.
- **Attack -** An assault on system security that derives from an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

Passive attacks

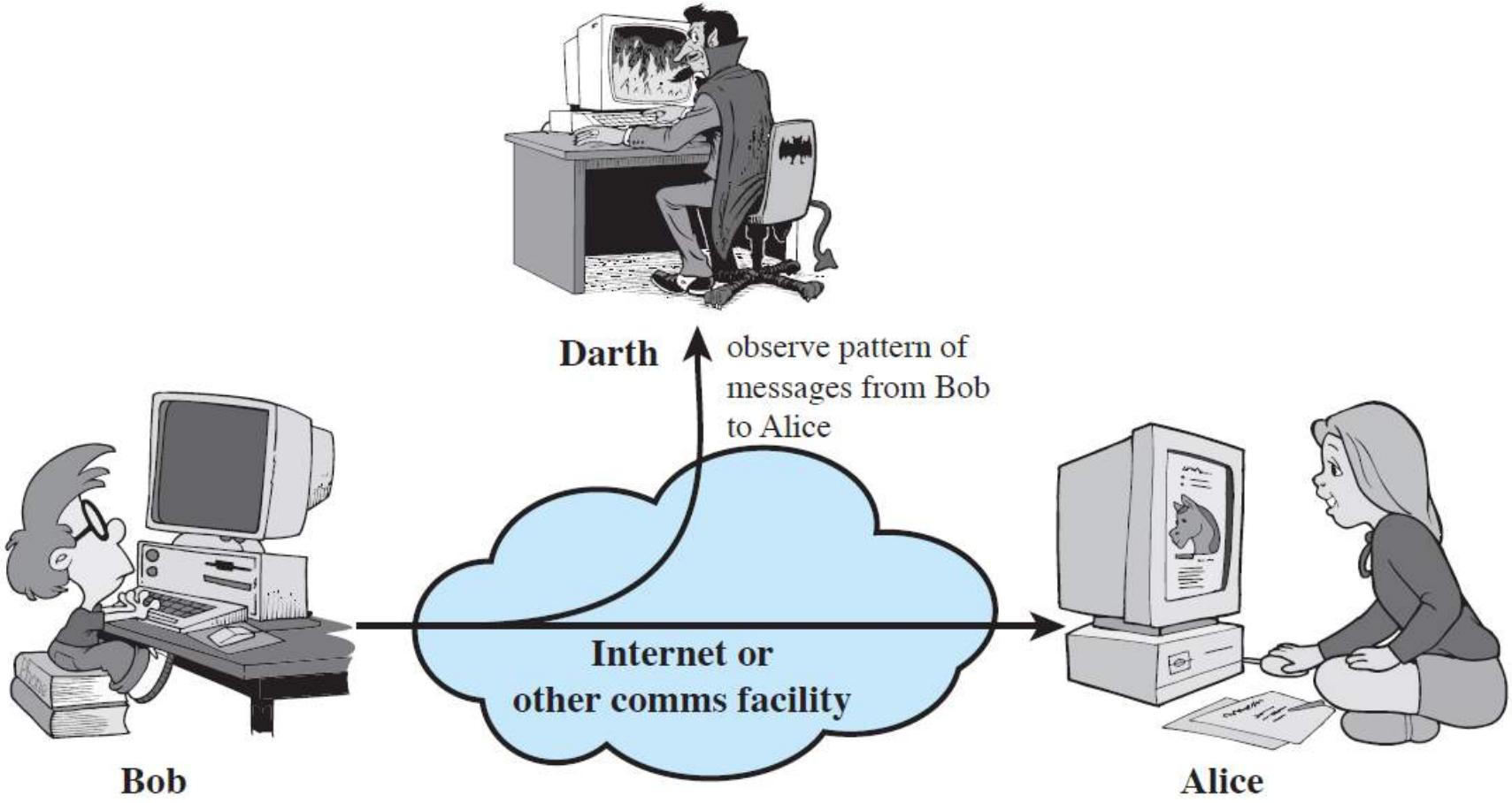
- A passive attack attempts to learn or make use of information from the system but does not affect system resources.
- *Passive attacks* are in the nature of monitoring the transmissions. The goal of the opponent is to obtain information that is being transmitted.

- Two types of passive attacks
  - Release of message contents
  - Traffic analysis
- Passive attacks are very difficult to detect
  - Message transmission apparently normal
    - No alteration of the data
  - Emphasis on prevention rather than detection
    - By means of encryption

Passive Attacks (1)
Release of Message Contents



Passive Attacks (2)
Traffic Analysis



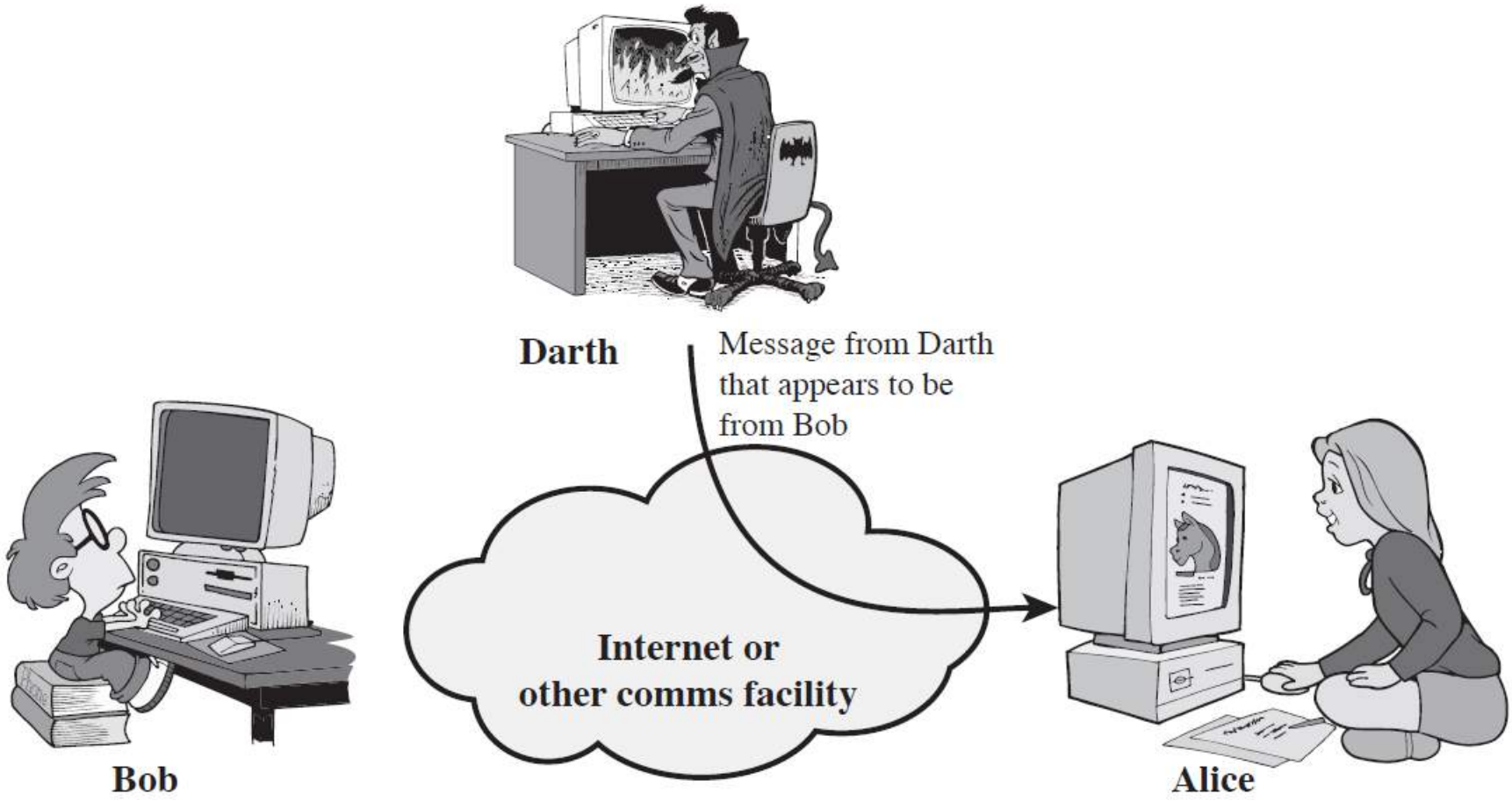
Active Attacks

- Active attacks try to alter system resources or affect their operation
  - Modification of data, or creation of false data
- Four categories
  - Masquerade
  - Replay
  - Modification of messages
  - Denial of service

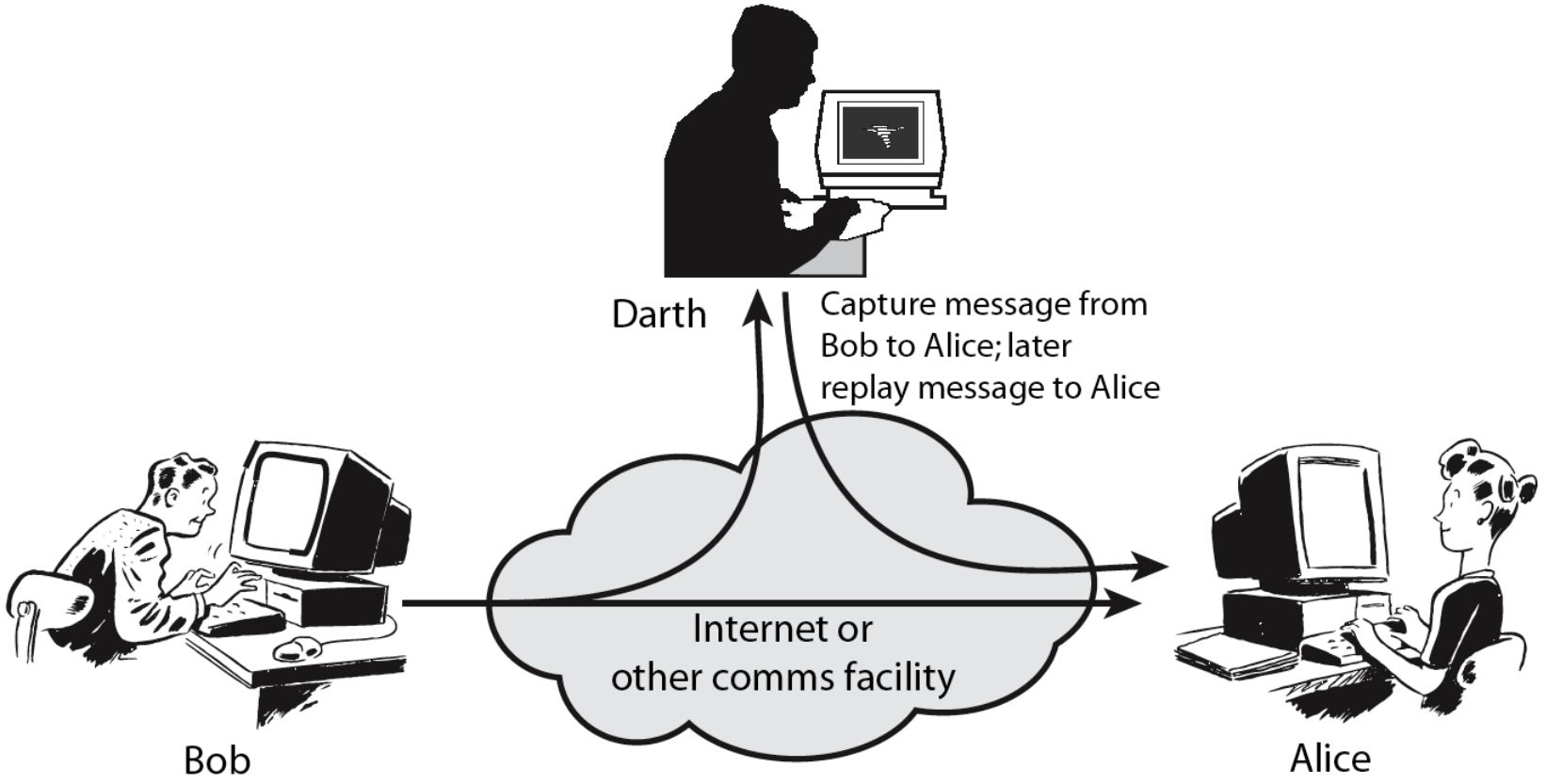
Difficult to prevent

- The goal is to detect and recover

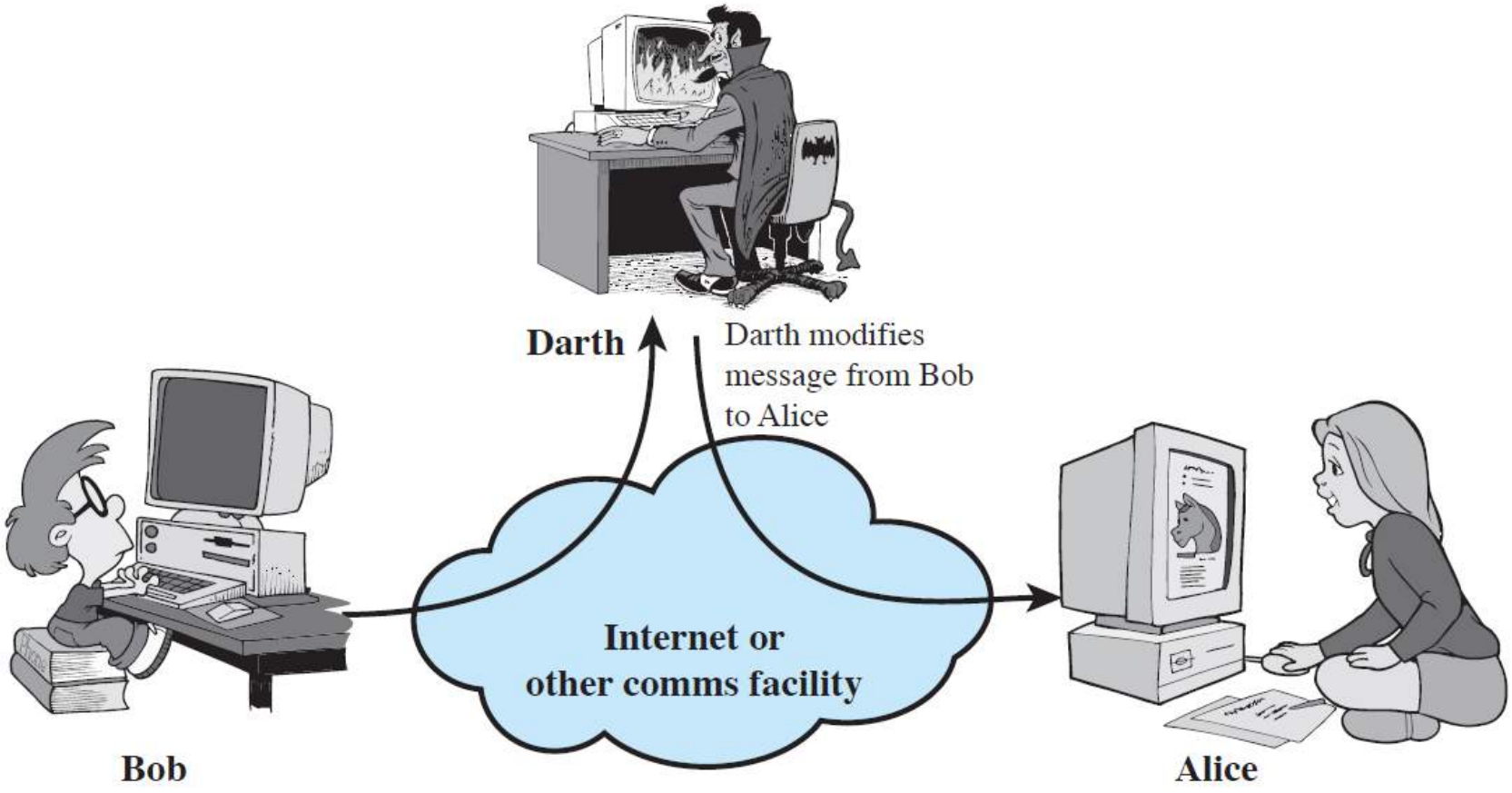
Active Attacks (1)
Masquerade



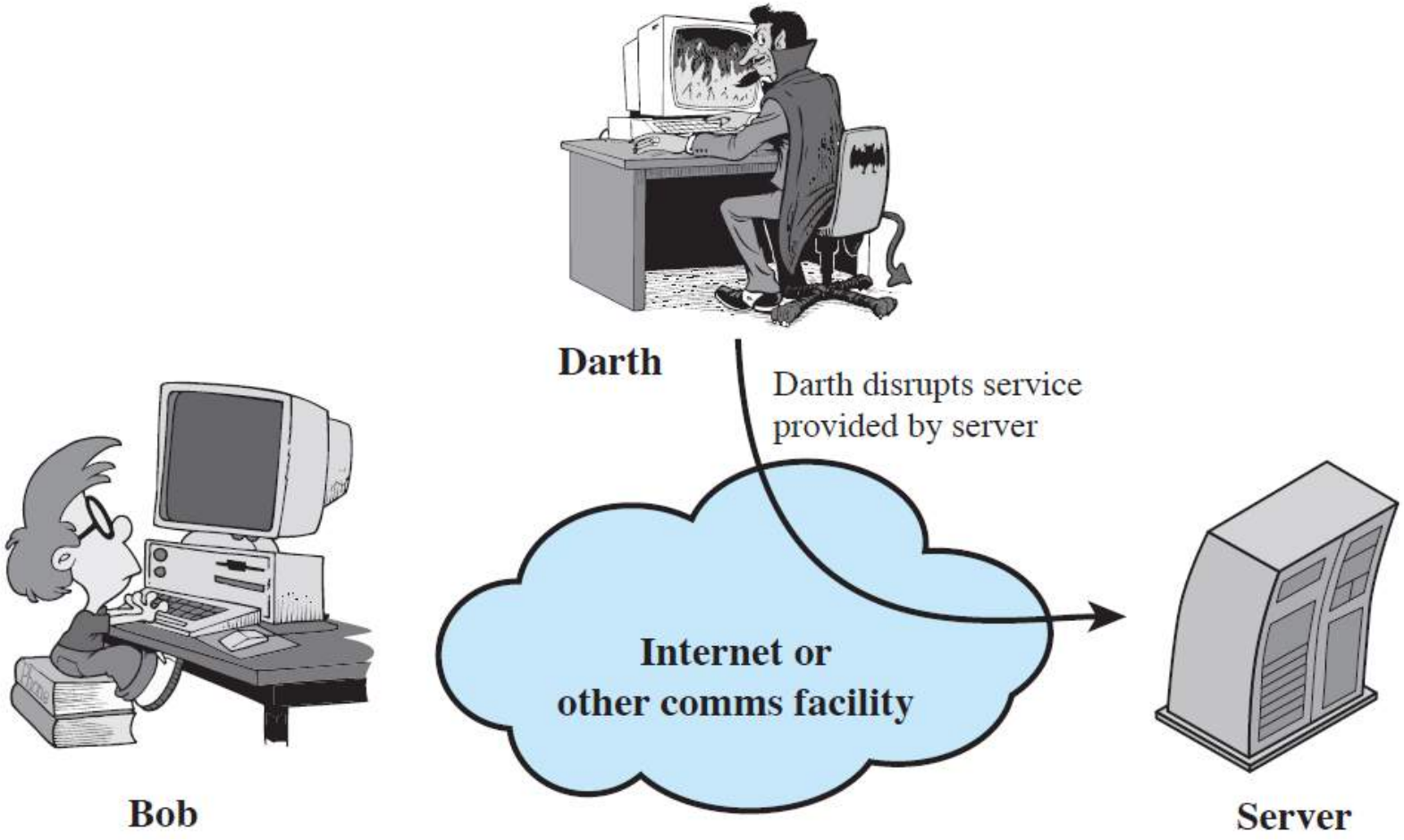
Active Attacks (2)
Replay



Active Attacks (3)
Modification of Messages



Active Attacks (4)
Denial of Service



Security Service

- enhance security of data processing systems and information transfers of an organization
- intended to counter security attacks
- using one or more security mechanisms
- often replicates functions normally associated with physical documents

Security Services

- X.800:

  "a service provided by a protocol layer of communicating open systems, which ensures adequate security of the systems or of data transfers"

- RFC 2828:

  "a processing or communication service provided by a system to give a specific kind of protection to system resources"

1. **Authentication** is concerned with assuring that a communication is authentic. Two specific authentication services are defined in X.800:

- **Peer entity authentication:** provides identity of a peer entity in an association.
- **Data origin authentication:** provides corroboration of the source of a data unit. Provide protection against the duplication or modification of data.

2. **Access control** is the ability to limit and control the access to host systems and applications via communications links.

3. **Confidentiality** is the protection of transmitted data from passive attacks, and the protection of traffic flow from analysis.
4. **Integrity** assures that messages are received as sent, with no duplication, insertion, modification, reordering, replay, or loss.
5. **Availability** is the property of a system / resource being accessible and usable upon demand by an authorized system entity, according to performance specifications for the system.

Security Mechanisms(X.800)

- Security mechanisms speccified in X.800 are

- 1. Speccific security mechanisms:

  Incorporated in to the appropriate protocol layer in order to provide some of the security services

  Encipherment(Encryption)

  Use of mathematical algorithm to transform data in to a form that is not readily intelligible

Digital Signature

Data appended to 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

Data Integrity

A variety of mechanisms used to assure the integrity of data unit

Authentication Exchange

A mechanism intended to ensure the identity of an entity

Traffic padding

The insertion of bits in to 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

Notarization

The use of a trusted third party

2. **Pervasive security mechanisms:**

These mechanisms are not specific to any particular OSI security service or protocol layer.

Trusted Functionality

Perceived to be correct with respect to some criteria

Security Label

The marking bound to a resource that names or designates the security attributes of that resource

Event detection

Detection of security relevant events

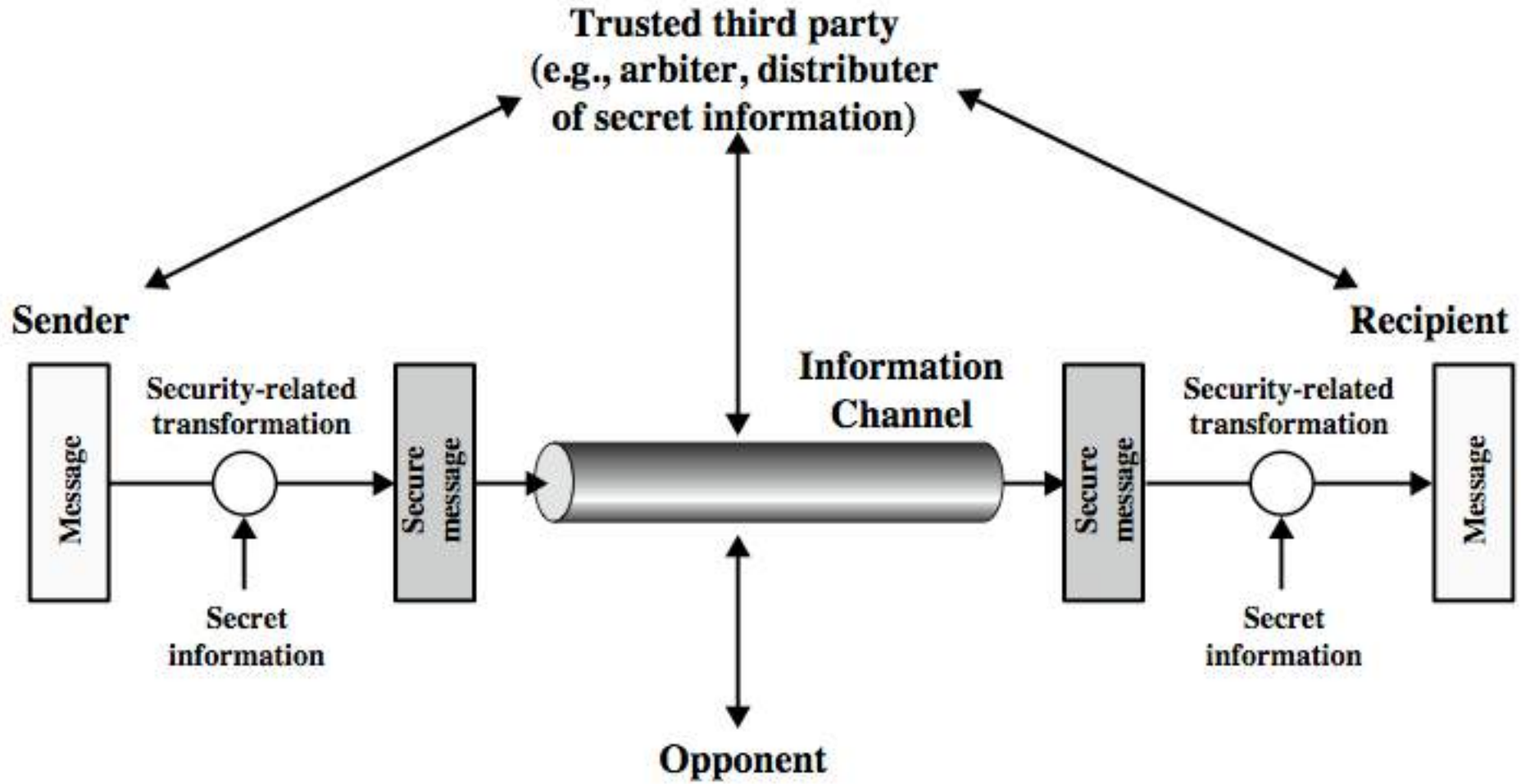
Security Audit Trail

Data collected and potentially used to facilitate a security audit

Security Recovery

Deals with requests from mechanisms

Model for Network Security



Model for Network Security

1. design a suitable algorithm for the security transformation
2. generate the secret information (keys) used by the algorithm
3. develop methods to distribute and share the secret information
4. specify a protocol enabling the principals to use the transformation and secret information for a security service

MODULE 2

MODULAR ARITHMETIC

- Modular arithmetic : Groups, Ring, Fields. The Euclidean algorithm-Finite fields of the form GF(p). Polynomial arithmetic: Finite fields of the form GF(2n).

Abstract algebra (Modern algebra)

- Elements are operated algebraically.
- Two elements can be combined in different ways to obtain a third element of the set.
- Not limited to ordinary arithmetic operations.
- Fundamental elements – groups, rings, fields.

Groups

- A group G is denoted by {G, ∘ } is a set of elements with binary operation denoted by ∘
- Each ordered pairs (a,b) of elements in G is associated with an element (a . b) in G, which follows the axioms:

A1. Closure : if a and b belong to G, then a.b is also in G.

A2. Associative : a.(b.c) = (a.b).c for all a,b,c in G

A3. Identity element: there is an element e in G such that a.e = e.a = a for all ‘a’ in G.

A4. Inverse element: For each ‘a’ in G there is an element a^{I} in G such that a. a^{I} = a^{I} . a = e.

Finite group: a group with finite number of elements.

Order of the group: number of elements

A group is said to be ‘abelian’ if it satisfies the following additional condition

A5. Commutative : a.b = b.a for all a,b in G

Cyclic Groups

- A Group is cyclic if every element b

(b $\in$ G) is a power of some fixed element ‘a’

ie $b = a^k$

- K is an integer
- ‘a’ is the generator of G
- Cyclic group is always abelian and may be finite or infinite

Rings

- A ring R denoted by {R, +, x}, is a set of elements with two binary operations called addition and multiplication.
- R satisfies the axioms A1 – A5, so R is abelian.

M1. Closure under multiplication: if a and b belongs to R, then ab is also in R.

M2. Associativity of multiplication: (ab)c = a(bc) for all a,b,c in R.

M3. Distributive laws: $a(b+c) = ab + ac$ for all a,b,c in R.

- A ring is a set in which we can do addition, subtraction and multiplication without leaving the set.

M4. Commutativity of multiplication: $ab = ba$ for all a,b in R.

M5: Multiplicative identity: there is an element ‘I’ in R such that aI = Ia =a for all ‘a’ in R.

Integral domain: Ring that satisfies A1 – A5 and M1 – M5

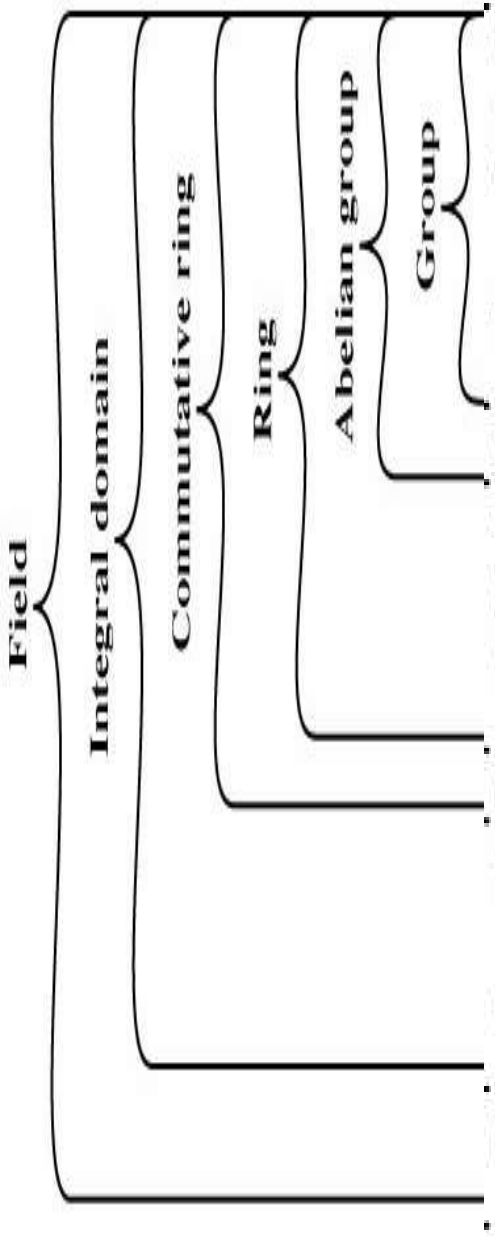
M6. Non zero divisors: if a,b in R and ab = 0, then either a = 0 or b = 0

Fields

- A field is denoted by {F, +, x} is a set of elements with two binary operations addition and multiplication.
- F satisfies A1 – A5, M1- M6.

M7. Multiplicative inverse: for each ‘a’ in F except 0, there is an element a^{-1} in F such that $a\ a^{-1} = a^{-1}\ a = 1$

- A field is a set in which addition, subtraction, multiplication and division without leaving the set.
- $a^3 = a.a.a$
- $a^0 = e$
- $a^{-1} = (a^l)^n$
- $a / b = a\ (b^{-1})$



(A1) Closure under addition: If a and b belong to S, then $a + b$ is also in S

(A2) Associativity of addition: $a + (b + c) = (a + b) + c$ for all a, b, c in S

(A3) Additive identity: There is an element 0 in R such that $a + 0 = 0 + a = a$ for all a in S

(A4) Additive inverse: For each a in S there is an element $-a$ in S such that $a + (-a) = (-a) + a = 0$

(A5) Commutativity of addition: $a + b = b + a$ for all a, b in S

(M1) Closure under multiplication: If a and b belong to S, then ab is also in S

(M2) Associativity of multiplication: $a(bc) = (ab)c$ for all a, b, c in S

(M3) Distributive laws: $a(b + c) = ab + ac$ for all a, b, c in S

$(a + b)c = ac + bc$ for all a, b, c in S

(M4) Commutativity of multiplication: $ab = ba$ for all a, b in S

(M5) Multiplicative identity: There is an element 1 in S such that $a1 = 1a = a$ for all a in S

(M6) No zero divisors: If a, b in S and $ab = 0$, then either $a = 0$ or $b = 0$

(M7) Multiplicative inverse: If a belongs to S and $a \neq 0$, there is an element a^{-1} in S such that $aa^{-1} = a^{-1}a = 1$

Modular Arithmetic

- n – positive integer
- a – any integer
- Dividing a by n, q is quotient, r is integer remainder, then
- $a = qn + r \qquad 0 \leq r < n$
- $q = \lfloor a/n \rfloor$

$\lfloor x \rfloor$ is the largest integer less than or equal to x

- The remainder is also referred to as residue.
- a = 11, n=7
- 11 = 1 x 7 + 4
- r = 4
- a mod n is the remainder when a is divided by n
- $a = qn + a \bmod n$

- a = -11 n = 7
- -11 = (-2) x 7 + 3
- r = 3
- 11 mod 7 = 4
- -11 mod 7 = 3
- Two integers a and b are said to be congruent modulo n if

  a mod n = b mod n, then a ≡ b mod n

Divisors

- A non zero ‘b’ divides ‘a’ if a = mb for some ‘m’
- ‘b’ divides ‘a’ if there is no remainder on division.
- Positive divisors of 12 ?

Relations

1. If a/1, then $a = \pm 1$
2. If a/b and b/a, then $a = \pm b$
3. If b/g and b/h , then b/ (mg + nh) for arbitary integers m and n

Properties of modular operator

1. $a \equiv b \bmod n$, if $n / (a - b)$
2. $a \equiv b \bmod n$ implies $b \equiv a \bmod n$
3. $a \equiv b \bmod n$ and $b \equiv c \bmod n$ implies $a \equiv c \bmod n$

Modular arithmetic operations

1. $[(a \bmod n) + (b \bmod n)] \bmod n =$

$$(a + b) \bmod n$$

2. $[(a \bmod n) - (b \bmod n)] \bmod n =$

$$(a - b) \bmod n$$

3. $[(a \bmod n) \times (b \bmod n)] \bmod n =$

$$(a \times b) \bmod n$$

Example $11 \bmod 8, 15 \bmod 8$

Properties of Modular Arithmetic for Integers in Z_n

Z_n is a set of non negative integers less than n

$Z_n = \{0,1,\ldots\ldots\ldots, (n - 1)\}$

1. Commutative laws

$$(w + x) \bmod n = (x + w) \bmod n$$
$$(w \times x) \bmod n = (x \times w) \bmod n$$

2. Associative laws

$$[(w + x) + y] \bmod n = [w + (x + y)] \bmod n$$
$$[(w \times x) \times y] \bmod n = [w \times (x \times y)] \bmod n$$

3. Distributive laws

$[w \times (x + y)] \bmod n = [(w \times x) + (w \times y)] \bmod n$

$[w + (x \times y)] \bmod n = [(w + x) \times (w + y)] \bmod n$

4. Identities

$(0 + w) \bmod n = w \bmod n$

$(1 \times w) \bmod n = w \bmod n$

5. Additive inverse (-w)

For each $w \in Z_n$, there exists a z such that

$w + z \equiv 0 \bmod n$

- Two integers are **relatively prime** if their only common positive integer factor is 1.

Greatest Common Divisor

- The greatest common divisor of a and b is denoted as **gcd (a,b)**
- **‘c’** is gcd(a,b) if

1.c is a divisor of a and of b

2.Any divisor of a and b is a divisor of c

gcd (a,b) =max [k, such that k/a and k/b]

gcd (a,b) = gcd (a,-b) = gcd (-a,-b)

In general gcd (a,b) = gcd ($|a|, |b|$)

- For any non negative integer ‘a’ and any positive integer ‘b’

  gcd (a,b) = gcd (b, a mod b)

Euclid’s Algorithm : simple procedure to find greatest common divisor of two positive integers.

Euclid Algorithm to find gcd (a,b)

- Euclid (a,b)

1. A← a; B ← b
2. If B = 0, return A = gcd (a,b)
3. R = A mod B
4. A ← B
5. B ← R
6. Goto step 2

gcd (55,22)

gcd (55,22) = gcd (22, 55 mod 22)

gcd (22, 11) = gcd (11, 22 mod 11)

gcd (11,0) = 11

- gcd (18,12) ?
- gcd (11, 10) ?

- Find gcd (1970, 1066) ?
- Find gcd (24140,16762)?

Finite fields of the form GF(p)

- The order of a finite field must be a power of a prime p^n where n is a positive integer.
- The finite field of order p^n is written as $GF(p^n)$ GF stands for Galois Field

Finite Fields of order p

- For a given prime, p, the finite field of order p, GF(p) is defined as the set Z_p of integers {0, 1,..., p - 1}, together with the arithmetic operations modulo p.
- **Multiplicative inverse** (w^{-1}): For each $w \in Z_p$, $w \neq 0$, there exists a $w \in Z_p$, such that $w \times z \equiv 1 \pmod{p}$.

GF(2)

+	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

- Addition modulo 8

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

(a) Addition modulo 8

- Multiplication modulo 8

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

(b) Multiplication modulo 8

Multiplication modulo 7

×	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

(b) Multiplication modulo 7

Extended Euclid's Algorithm

- Euclid algorithm can be extended so that, in addition to finding gcd(m,b), if the gcd is 1, the algorithm returns the multiplicative inverse of b.

Extended Euclid (a,b)

1. $(A1,A2,A3) \leftarrow (1,0,a)$;
   $(B1,B2,B3) \leftarrow (0,1,b)$;
2. if B3=0; return A3=gcd(a,b);
   //no inverse
3. if B3=1; return B3 = gcd(a,b);
   B2 = $b^{-1} \bmod a$;
4. Q = $\lfloor A3/B3 \rfloor$;
5. $[T1,T2,T3] \leftarrow [A1-QB1,\ A2-QB2,\ A3-QB3]$;
6. $[A1,A2,A3] \leftarrow [B1,B2,B3]$;

7. [B1,B2,B3]←[T1,T2,T3];
8. Goto step 2.

- Find the multiplicative inverse of 550 in GF(1759)0r 550 Mod 1759
- 1234 mod 4321

Polynomial Arithmetic

- Three classes of polynomial arithmetic:

1. Ordinary polynomial arithmetic – using the basic rules of algebra.
2. Polynomial arithmetic with coefficients in Z_p

Ordinary polynomial arithmetic

- A polynomial of degree n is an expression of the form,

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$$

- a_i are elements of some designated set of numbers S called the coefficient set and $a_n \neq 0$

- A zeroth degree polynomial is called a constant polynomial.
- A nth degree polynomial is said to be a monic polynomial if $a_n = 1$
- The variable x is referred as indeterminate.

- Addition and subtraction is performed by adding or subtracting corresponding coefficients. $n \geq m$

$$f(x) = \sum_{i=0}^{m} a_i x^i \qquad g(x) = \sum_{i=0}^{n} b_i x^i$$

$$f(x) + g(x) = \sum_{i=0}^{m} (a_i + b_i) x^i + \sum_{i=m+1}^{n} b_i x^i$$

Find f(x)+g(x), f(x)- g(x), f(x)xg(x),

$$f(x) = x^3 + x^2 + 2$$

$$g(x) = x^2 - x + 1$$

Polynomial Arithmetic with coefficients in Z_p

- Polynomial in which the coefficients are elements of some field *F,* is referred as *a polynomial over the field F*.
- Such polynomials set is referred to as a **polynomial ring**.
- Within a field, two elements a and b, the quotient a/b is also an element of the field. However, given a ring R that is not a field, division will result in a quotient and a remainder; this is not exact division.
-

$$\frac{f(x)}{g(x)} = q(x) + \frac{r(x)}{g(x)}$$

$$f(x) = q(x)g(x) + r(x)$$

- degree of f(x) is n, and of g(x) is m, n ≥ m, then degree of the quotient q(x), is (n-m) and of remainder is at most (m–1).

Polynomial arithmetic over GF(2)

$$f(x) = x^7 + x^5 + x^4 + x^3 + x + 1$$

$$g(x) = x^3 + x + 1$$

$$f(x) + g(x) = x^7 + x^5 + x^4$$

$$f(x) - g(x) = x^7 + x^5 + x^4$$

$$f(x) \times g(x) = x^{10} + x^4 + x^2 + 1$$

$$f(x) \Big/ g(x) = x^4 + 1$$

Greatest common divisor

- The polynomial c(x) is said to be the greatest common divisor of a(x) and b(x) if

1. c(x) divides both a(x) and b(x)
2. Any divisor of a(x) and b(x) is a divisor of c(x)

- gcd[a(x), b(x)]
- Euclid[$a(x)$, $b(x)$]

1. A(x) ← $a(x)$; B(x) ← $b(x)$
2. **if** B(x) = 0 **return** A(x) = gcd[$a(x)$, $b(x)$]
3. R(x) = A(x) mod B(x)
4. A(x) ← B(x)
5. B(x) ← R(x)
6. **goto** 2

Find gcd [a(x), b(x)]

$$a(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$$

$$b(x) = x^4 + x^2 + x + 1$$

$$\gcd = x^3 + x^2 + 1$$

Finding multiplicative inverse

Extended Euclid [m(x), b(x)]

1. $[A_1(x), A_2(x), A_3(x)] \leftarrow [1,0,m(x)]$;
   $[B_1(x), B_2(x), B_3(x)] \leftarrow [0,1,b(x)]$;
2. If $B_3(x)=0$ return $A_3(x) = \gcd [m(x), b(x)]$;
   no inverse
3. If $B_3(x)=1$ return $B_3(x) = \gcd [m(x), b(x)]$;
   $B_2(x) = b(x)^{-1} \bmod m(x)$

4. $Q(x)$ = quotient of $A_3(x) / B_3(x)$
5. $[T_1(x), T_2(x), T_3(x)] \leftarrow [A_1(x) - Q(x)\, B_1(x), A_2(x) - Q(x)\, B_2(x), A_3(x) - Q(x)\, B_3(x)]$
6. $[A_1(x), A_2(x), A_3(x)] \leftarrow [B_1(x), B_2(x), B_3(x)]$
7. $[B_1(x), B_2(x), B_3(x)] \leftarrow [T_1(x), T_2(x), T_3(x)]$
8. Goto 2

Find multiplicative inverse of

$(x^7 + x + 1)$ in $(x^8 + x^4 + x^3 + x + 1)$

- In GF(2), addition is equivalent to the XOR operation, and multiplication is equivalent to the logical AND operation.
- A polynomial f(x) over a field is called **irreducible if and only if it cannot** be expressed as a product of two polynomials.
- An irreducible polynomial is also called a **prime polynomial.**

Primitive Element

- *In GF(q), a nonzero element 'a' is said to be primitive if the order* of *a is q − 1.*
- *The powers of a primitive element generate all the nonzero* elements of *GF(q).*
- *Every finite field has a primitive element.*

- *f(x) is said to be irreducible if it is not divisible by any* polynomial over *GF(2) of degree less than n but greater than* zero.
- *x^2, $x^2 + 1$, $x^2 + x$ are reducible over GF(2).*
- *$x + 1$, $x^2 + x + 1$, $x^3 + x + 1$ are irreducible over GF(2).*
- *For any m > 1,, there exists an irreducible polynomial of degree m.*

- A **primitive polynomial** is the minimal polynomial of a primitive element of the finite extension field GF(p^m).
- In other words, a polynomial with coefficients in GF(p) is a primitive polynomial if its degree is m and it has a root in GF(p^m) such that is the entire field GF(p^m).

- Initially, we have two elements 0 and 1 from *GF(2) and a new* symbol *α.*

EXAMPLE 4: There are two polynomials over GF(2) with degree 1: X and $1+X$.

EXAMPLE 5: There are four polynomials over GF(2) with degree 2: X^2, $1 + X^2$, $X + X^2$, and $1 + X + X^2$.

EXAMPLE 9: Among the four polynomials of degree 2 ; X^2, $X^2 + 1$ and $X^2 + X$ are not irreducible since they are either divisible by X or $X + 1$. However, $X^2 + X + 1$ does not have either "0" or "1" as a root and so is not divisible by any polynomial of degree 1.Therefore, $X^2 + X + 1$ is an irreducible polynomial of degree 2.

Construction of $GF(2^m)$

- GF(2) has two elements 0,1 and a new element α.

$$
\begin{aligned}
0 \cdot 0 &= 0, \\
0 \cdot 1 &= 1 \cdot 0 = 0, \\
1 \cdot 1 &= 1, \\
0 \cdot \alpha &= \alpha \cdot 0 = 0, \\
1 \cdot \alpha &= \alpha \cdot 1 = \alpha,
\end{aligned}
$$

$$\alpha^2 = \alpha \cdot \alpha,$$

$$\alpha^3 = \alpha \cdot \alpha \cdot \alpha,$$

$$\vdots$$

$$\alpha^j = \alpha \cdot \alpha \cdot \cdots \cdot \alpha \quad (j \text{ times}),$$

$$\vdots$$

Let $p(X)$ be a primitive polynomial of degree m over GF(2). We assume that $p(\alpha) = 0$. Since $p(X)$ divides $X^{2^m-1} + 1$,we have:-

$$X^{2^m-1} + 1 = q(X)p(X).$$

If we replace X by α in above equation, we obtain

$$\alpha^{2^m-1} + 1 = q(\alpha)p(\alpha).$$

Since $p(\alpha) = 0$, we have

$$\alpha^{2^m-1} + 1 = q(\alpha) \cdot 0.$$

$$\alpha^{2^m - 1} + 1 = 0.$$

Adding 1 to both sides of above equation (use modulo-2 addition) results in the following equality:

$$\alpha^{2^m - 1} = 1.$$

Therefore, under the condition that $p(\alpha) = 0$, the set F becomes finite and contains the following elements:

$$F^* = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^m-2}\}.$$

The construction of GF(4)

Because $4 = 2^2$, we seek a primitive polynomial in $GF(2)[x]$ of degree 2. Let $p(x) = x^2 + x + 1$. Let α be a root of $p(x)$. This implies that $\text{ord}(\alpha) = 3$ and $\alpha^2 + \alpha + 1 = 0$, i.e., $\alpha^2 = \alpha + 1$. Then,

Exp. Rep.	Poly. Rep.	Vector-space Rep. $(1,\alpha)$
α^0	1	(1, 0)
α^1	α	(0, 1)
α^2	$\alpha+1$	(1, 1)
0	0	(0, 0)

The construction of GF(8)

Because $8 = 2^3$, we seek a primitive polynomial in $GF(2)[x]$ of degree 3. Let $p(x) = x^3 + x + 1$. Let α be a root of $p(x)$. This implies that $\text{ord}(\alpha) = 7$ and $\alpha^3 + \alpha + 1 = 0$, i.e., $\alpha^3 = \alpha + 1$. Then,

$$\alpha^4 = \alpha^3 \cdot \alpha = \alpha^2 + \alpha$$

$$\alpha^5 = \alpha^4 \cdot \alpha = \alpha^3 + \alpha^2 = \alpha + 1 + \alpha^2$$

$$\alpha^6 = \alpha^5 \cdot \alpha = \alpha^3 + \alpha^2 + \alpha = \alpha + 1 + \alpha^2 + \alpha = \alpha^2 + 1.$$

Exp. Rep.	Poly. Rep.	Vector-space Rep. $(1, \alpha, \alpha^2)$
α^0	1	(1, 0, 0)
α^1	α	(0, 1, 0)
α^2	α^2	(0, 0, 1)
α^3	$1+\alpha$	(1, 1, 0)
α^4	$\alpha+\alpha^2$	(0, 1, 1)
α^5	$1+\alpha+\alpha^2$	(1, 1, 1)
α^6	$1+\alpha^2$	(1, 0, 1)
0	0	(0, 0, 0)

The construction of GF(16)

Let $p(x) = x^4 + x + 1$.

Exp. Rep.	Poly. Rep.	Vector-space Rep. $(1, \alpha, \alpha^2, \alpha^3)$	Order	$\log_\alpha(x)$	$\log_\alpha(x+1)$
0	0	(0, 0, 0, 0)	-	*	0
α^0	1	(1, 0, 0, 0)	1	0	*
α^1	α	(0, 1, 0, 0)	15	1	4

α^2	α^2	(0, 0, 1, 0)	15	2	8
α^3	α^3	(0, 0, 0, 1)	5	3	14
α^4	$\alpha+1$	(1, 1, 0, 0)	15	4	1
α^5	$\alpha^2+\alpha$	(0, 1, 1, 0)	3	5	10
α^6	$\alpha^3+\alpha^2$	(0, 0, 1, 1)	5	6	13
α^7	$\alpha^3+\alpha+1$	(1, 1, 0, 1)	15	7	9
α^8	α^2+1	(1, 0, 1, 0)	15	8	2
α^9	$\alpha^3+\alpha$	(0, 1, 0, 1)	5	9	7
α^{10}	$\alpha^2+\alpha+1$	(1, 1, 1, 0)	3	10	5
α^{11}	$\alpha^3+\alpha^2+\alpha$	(0, 1, 1, 1)	15	11	12
α^{12}	$\alpha^3+\alpha^2+\alpha+1$	(1, 1, 1, 1)	5	12	11
α^{13}	$\alpha^3+\alpha^2+1$	(1, 0, 1, 1)	15	13	6
α^{14}	α^3+1	(1, 0, 0, 1)	15	14	3

Module 3

Secure Communication

Model for Network Security

1. design a suitable algorithm for the security transformation
2. generate the secret information (keys) used by the algorithm
3. develop methods to distribute and share the secret information
4. specify a protocol enabling the principals to use the transformation and secret information for a security service

KEY POINTS

- Plain Text:

  Original message is known as plain text
- Cipher text:

  Coded message is called cipher text .
- Enciphering or Encryption: The process of converting from plaintext to cipher text is known as enciphering or encryption

- Deciphering or decryption: Restoring the plain text from the cipher text is known as deciphering or decryption
- Cryptography: The many schemes used for encryption constitute the area of study known as cryptography
- Cipher: The scheme for cryptography is known as cipher

ENCRYPTION

- 2 Types

1) Symmetric Encryption
2)Asymmetric Encryption

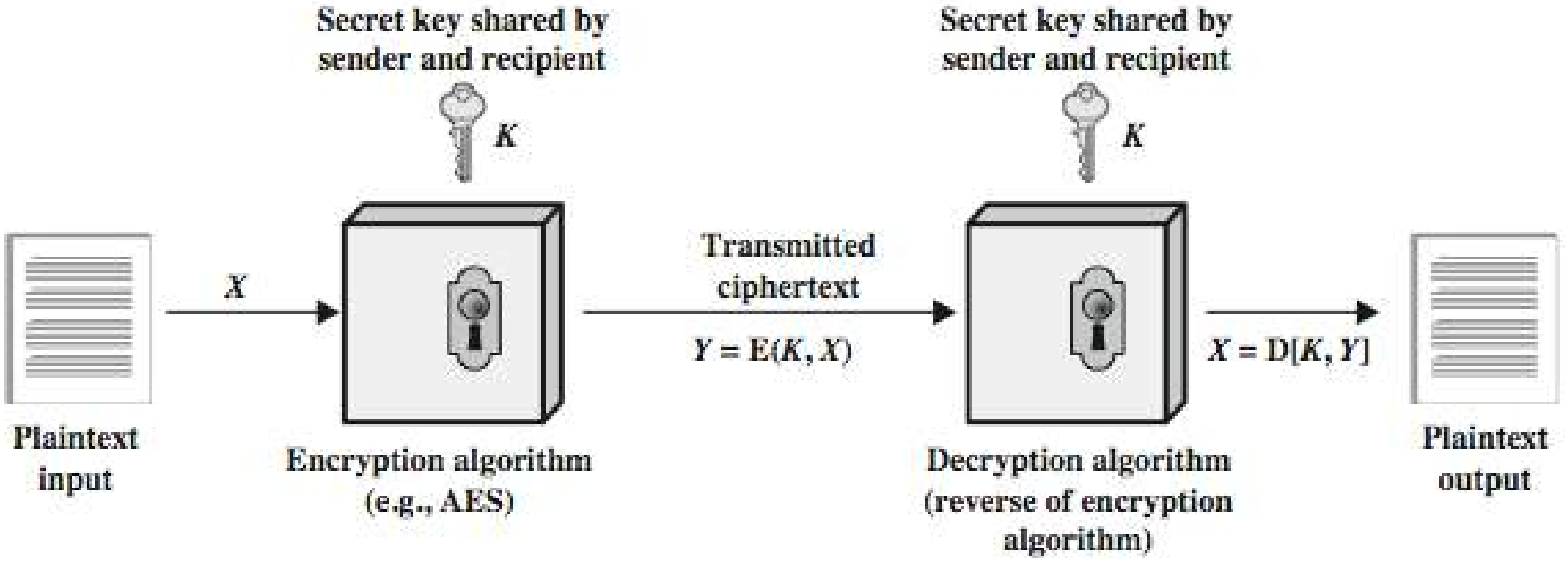
Symmetric Encryption

- conventional / private-key / single-key
- sender and recipient share a common key
- all classical encryption algorithms are private-key.

Some Basic Terminology

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

Symmetric Cipher Model



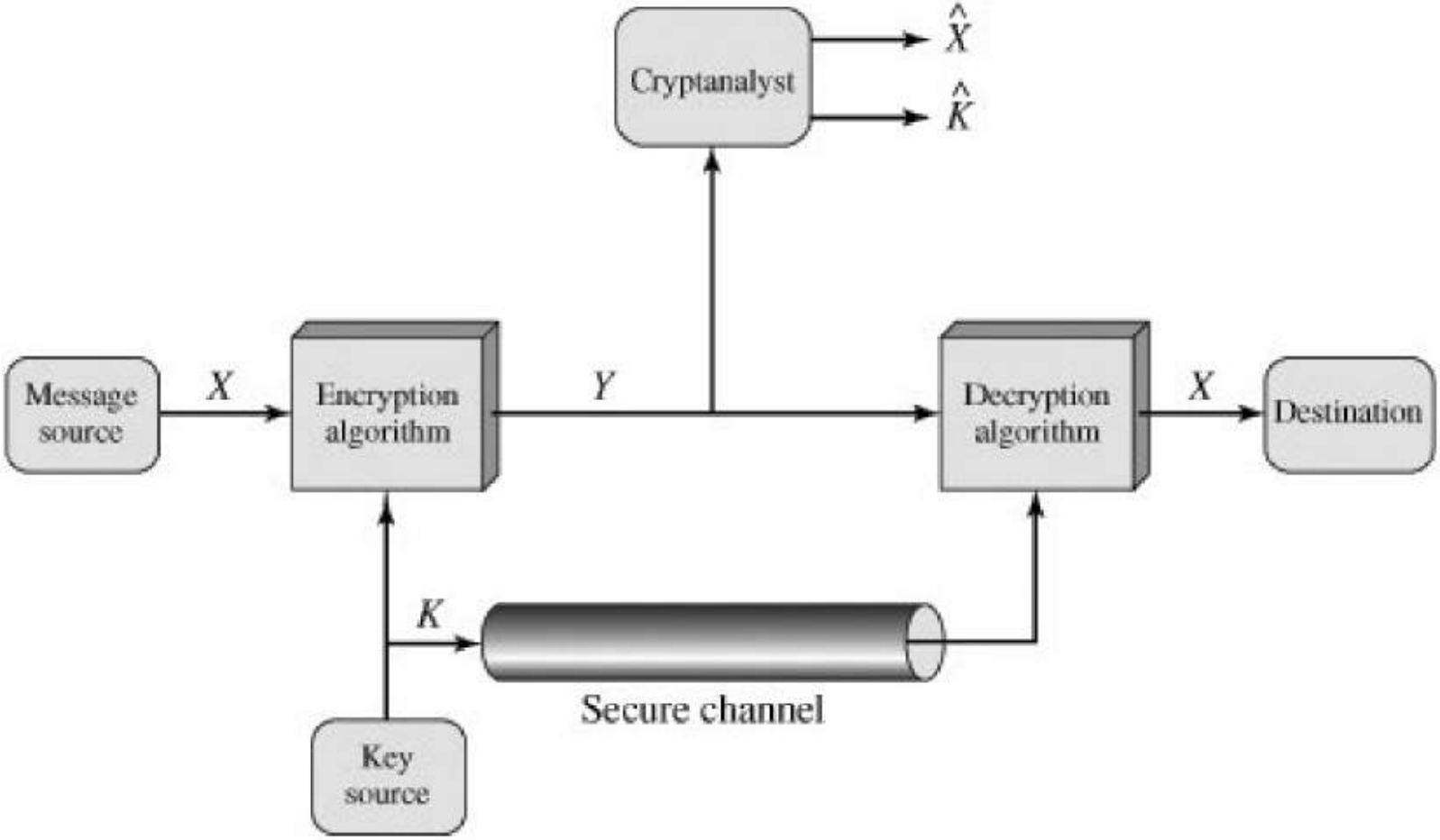
Requirements

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

  $Y = E(K, X)$

  $X = D(K, Y)$

Model of conventional cryptosystem



Cryptography

- Characterize cryptographic system by:
  - type of encryption operations used(plaintext to ciphertext)
    - Substitution- each element in plain text is mapped into another element
    - Transposition – elements in plain text are rearranged
    - Product – multiple stages of substitution and transposition

– number of keys used
  - single-key or private – both sender and receiver use the same key.
  - two-key or public - sender and receiver uses different key.
– way in which plaintext is processed
  - Block – process one block of input at a time
  - Stream – process input continuously

Cryptanalysis

- objective to recover key not just message
- general approaches:
  - cryptanalytic attack (cryptanalysis) – rely on nature of the algorithm plus some knowledge of the characteristics of the plain text.
  - brute-force attack – tries every possible key on a piece of cipher until an intelligent translation into plain text is obtained.

- computationally secure
  - The cost of breaking the cipher exceeds the value of information
  - The time required to break the cipher exceeds the lifetime of information
- **unconditional security**

– no matter how much computer power is available, the cipher cannot be broken

– no matter how much time is available for the opponent – code breaking is impossible

Brute Force Search

- always possible to simply try every key
- most basic attack, proportional to key size

Key Size (bits)	Number of Alternative Keys	Time required at 1 decryption/μs
32	$2^{32} = 4.3 \times 10^9$	2^{31} μs = 35.8 minutes
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs = 1142 years
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} μs = 5.4×10^{24} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs = 5.9×10^{36} years

Substitution Techniques

- The letters of the plain text are replaced by other letters, numbers or symbols.
- If plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns

Caesar Cipher

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

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
```

- example:

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

- For each plain text letter p, the cipher text letter C is

$$C = E(p) = (p + 3) \bmod (26)$$

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

- The decryption algorithm is

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

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
- eg. break ciphertext “JRRG PRUQLQJ"

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
```

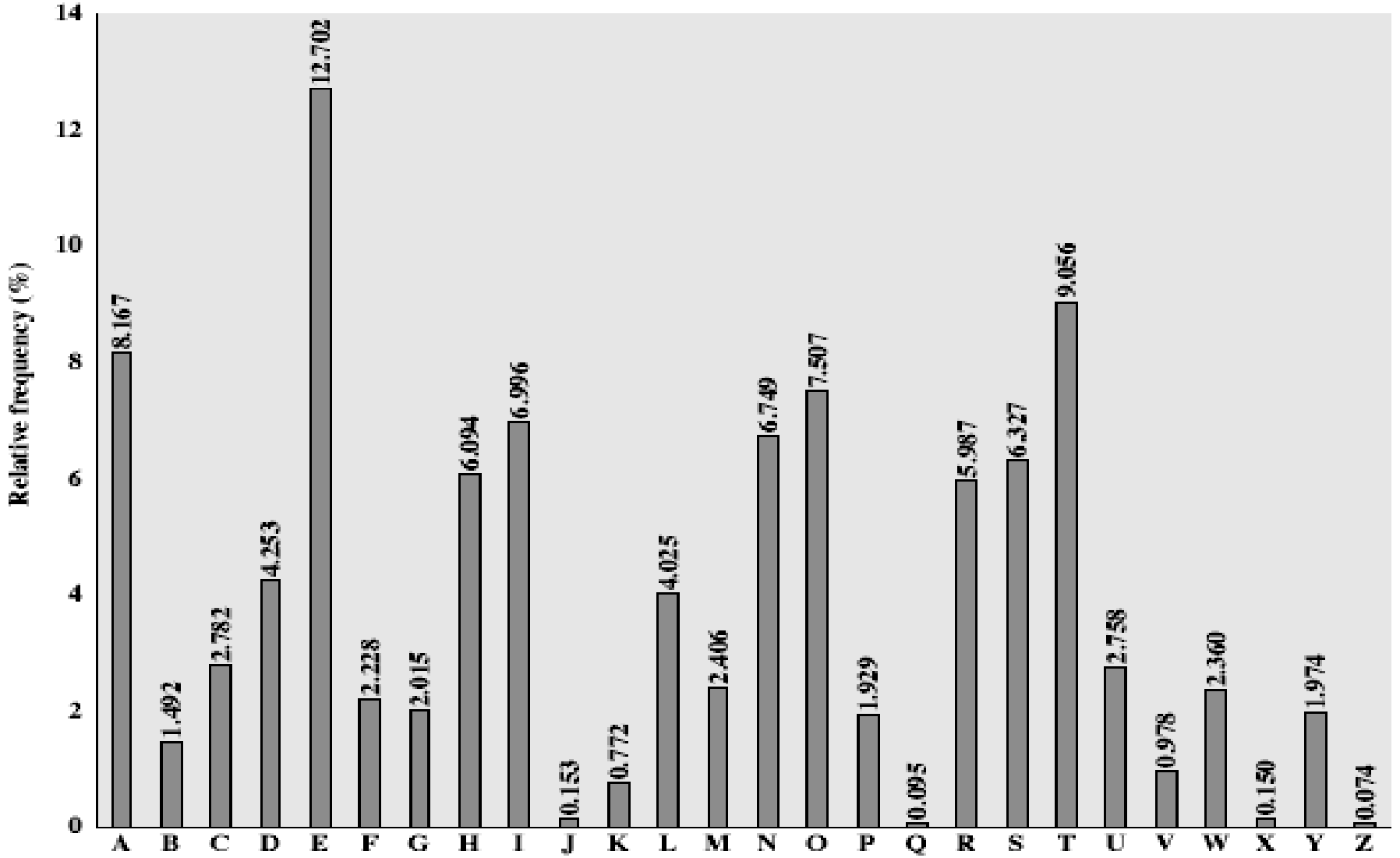
Monoalphabetic Cipher Security

- now have a total of 26! = 4 x 1026 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**
- letters are not equally commonly used
- in English **e** is by far the most common letter than T,R,N,I,O,A,S
- other letters are fairly rare
- Z,J,K,Q,X
- have tables of single, double & triple letter frequencies

English Letter Frequencies



Use in Cryptanalysis

- key concept - monoalphabetic substitution ciphers do not change relative letter frequencies
- calculate letter frequencies for ciphertext
- compare counts/plots against known values

Example Cryptanalysis

Ciphertext:

UZ`QSO`VUOHXMOPV`GPOZPEVSG`ZWSZ

`OPFPESX`

UDBMETSX`AIZ`VUEPHZ`HMDZSHZO`WSF

P`APPD`TSVP`QUZW`YMXUZUHSX`

EPYEPOPDZSZUFPO`MB`ZWP`FUPZ`HMDJ

`UD`TMOHMQ

P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	T 2.50	I 0.83	N 0.00
O 7.50	X 4.17	A 1.67	J 0.83	R 0.00
M 6.67				

- guess P & Z are e and t
- guess ZW is ‘th’ and hence ZWP is ‘the’
- proceeding with trial and error finally get:

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

- Even the large number of keys in a mono alphabetic cipher not providing security.
- Two principle methods are used

1. To encrypt multiple letters of plain text.
2. To use multiple cipher alpbabet.

Playfair Cipher

- It is multiple letter encryption cipher.
- It is based on the use of a 5 x 5 matrix of letters constructed using a keyword.
- Example
- The key word is monarchy.
- The matrix is constructed by filling the keyword followed by remaining letters in alphabetic order. The letters I and J are counted as one letter.

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Encrypting and Decrypting

- Plaintext is encrypted two letters at a time:

1. if a pair is a repeated letter, insert a filler like 'X', eg. "balloon" encrypts as "ba lx lox on"

2. if both letters fall in the same row of the matrix, replace each with letter to right (wrapping back to start from end), eg. “ar" encrypts as "RM"

3. if both letters fall in the same column, replace each with the letter below it (again wrapping to top from bottom), eg. “mu" encrypts to "CM"

4. otherwise each letter is replaced by the letter that lies in its row and the column occupied by the other plain text letter, eg. “hs" encrypts to "BP", and “ea" to "IM" or "JM" (as desired)

Security of the Playfair Cipher

- security much improved over monoalphabetic
- since have 26 x 26 = 676 diagrams
- would need a 676 entry frequency table to analyse (verses 26 for a monoalphabetic)
- and correspondingly more ciphertext
- was widely used for many years (eg. US & British military in WW1)

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
- Features

1. A set of related monoalphabetic substitution rules is used.
2. A key determines which particular rule is chosen for a given transformation

Vigenère Cipher

- Simplest polyalphabetic substitution cipher is the **Vigenère Cipher**
- Effectively multiple caesar ciphers.
- Caesar ciphers with shifts 0 through 25.
- The caesar cipher with shift of 3 is denoted by the key value d.

Example

Process:

1. write the plaintext out
2. write the keyword repeated above it
3. use each key letter as a caesar cipher key
4. encrypt the corresponding plaintext letter

- eg using keyword *deceptive*

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

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

Security of Vigenère Ciphers

- have multiple ciphertext letters for each plaintext letter.
- hence letter frequencies information is obscured.
- The length of the keyword…….
- To defend the cryptanalysis a keyword as long as the plain text is selected.

Autokey Cipher

- ideally want a key as long as the message
- Vigenère proposed the **autokey** cipher
- with keyword is prefixed to message as key
- knowing keyword can recover the first few letters
- use these in turn on the rest of the message
- but still have frequency characteristics to attack
- eg. given key *deceptive*

```
key:        deceptivewearediscoveredsav
plaintext:  wearediscoveredsaveyourself
ciphertext:ZICVTWQNGKZEIIGASXSTSLVVWLA
```

One-Time Pad

- if a truly random key as long as the message is used, the cipher will be secure --- One-Time pad
- It 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

Problems

1. Making large quantity of random keys.
2. Distribution and protection of keys.

Hill Cipher

- Another interesting multiletter cipher is the Hill cipher, developed by the mathematician Lester Hill in 1929.
- *This 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 ... z = 25

- For *m = 3, the system can be described as*

  c1 = (*k11P1 + k12P2 + k13P3) mod 26*

  c2 = (*k21P1 + k22P2 + k23P3) mod 26*

  c3 = (*k31P1 + k32P2 + k33P3) mod 26*

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

C = KP mod 26

Example

- plaintext "paymoremoney“

$$K = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix}$$

- The first 3 letters of the text gives

$$p = \begin{bmatrix} 15 \\ 0 \\ 24 \end{bmatrix}$$

$$K \times P = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix} \begin{bmatrix} 15 \\ 0 \\ 24 \end{bmatrix} = \begin{bmatrix} 375 \\ 819 \\ 486 \end{bmatrix} \bmod 26 = \begin{bmatrix} 11 \\ 13 \\ 18 \end{bmatrix} = LNS$$

- LNSHDLEWMTRW

- Plain text “friday”
- K = 2 x 2 = $\begin{bmatrix} 7 & 8 \\ 19 & 3 \end{bmatrix}$

- PQCFKU

Simplified DES (S-DES)

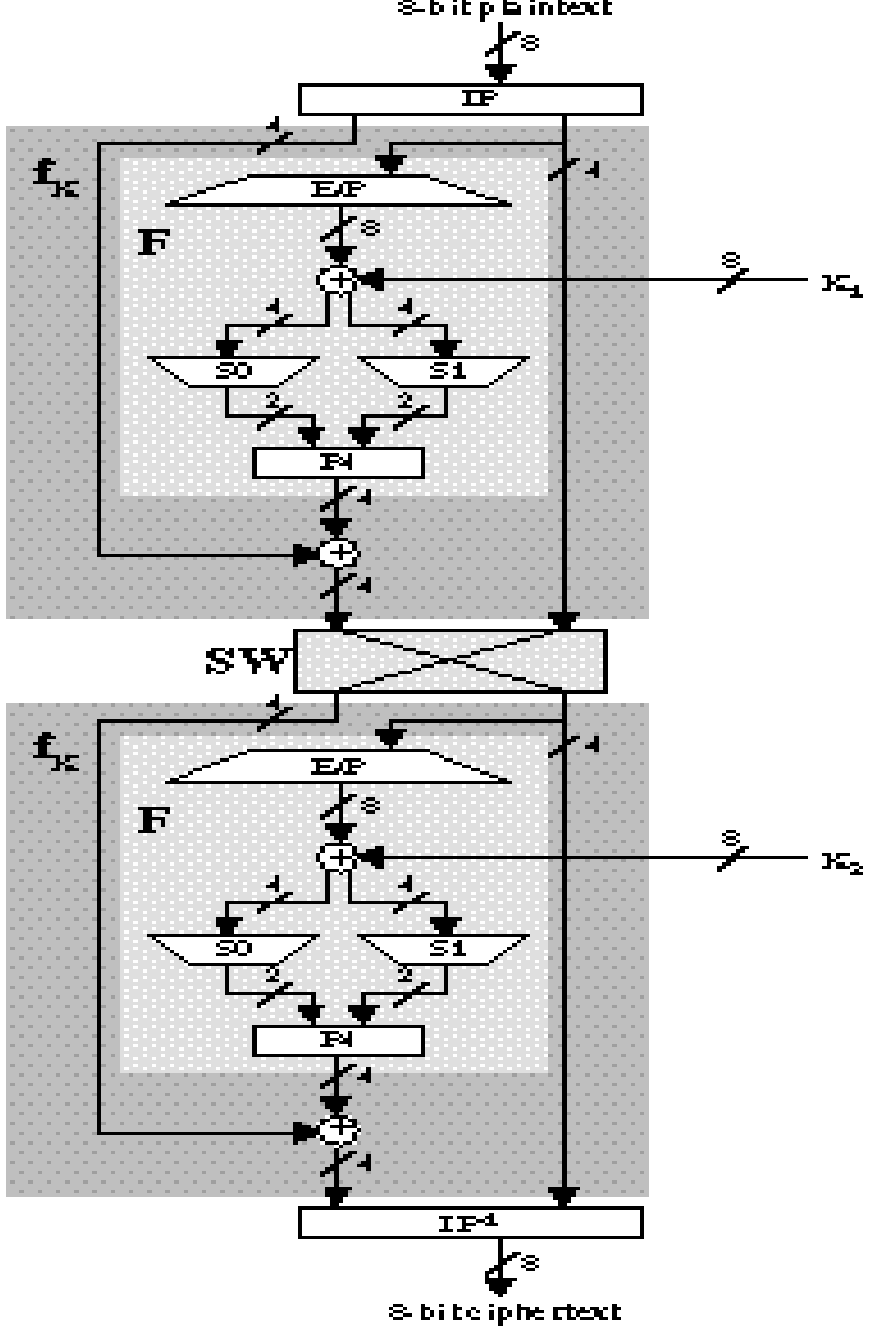
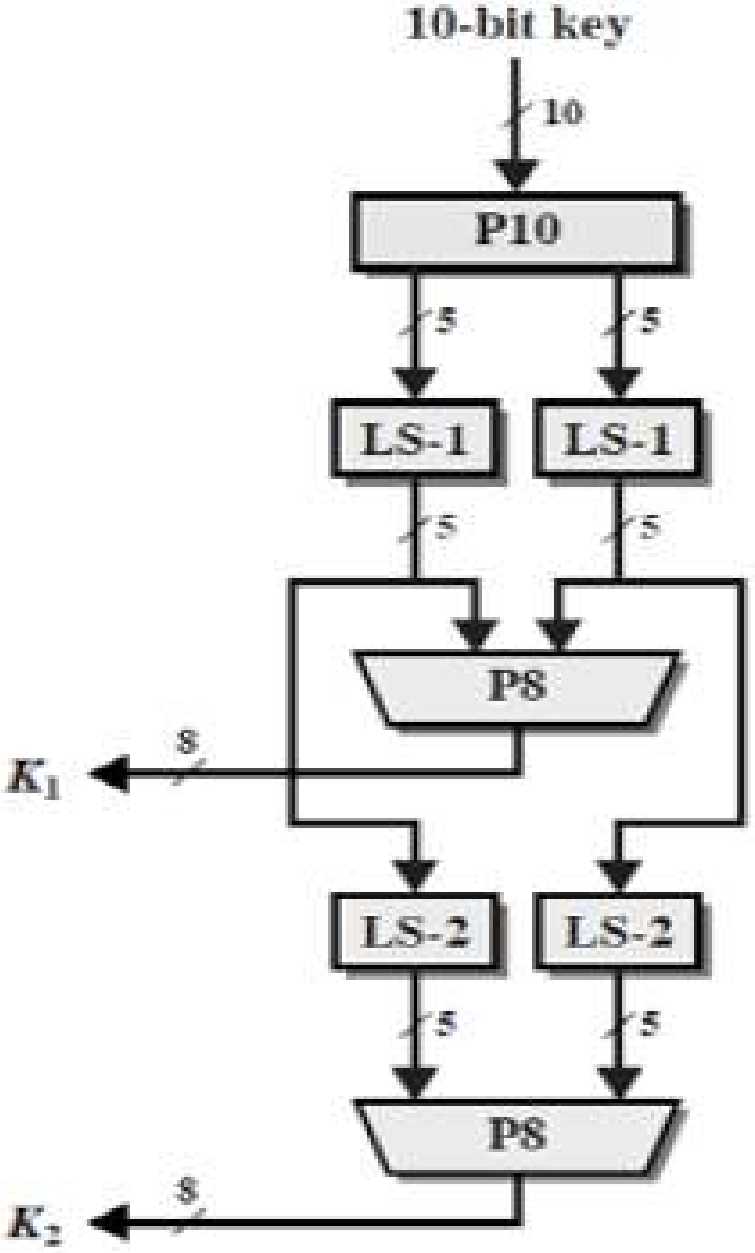


Figure 3.3 Simplified DES Encryption Detail


10-bit key
10
P10
5
5
LS-1
LS-1
5
5
P8
8
K_1
LS-2
LS-2
5
5
P8
8
K_2

Key – 1010000010
Initial permutation is P10

P10									
3	5	2	7	4	10	1	9	8	6

After permutation- 1000001100
Split into two 5 bits

L- 10000

R- 01100

Perform circular shift to L and R

After shift

L-00001

R-11000

Now key is 0000111000

Apply P8 which permutes 8 out of 10 bits

P8							
6	3	7	4	8	5	10	9

Now sub key K1 is obtained

K1 – 10100100

K2- 01000011

- Input - 0 1 1 0 1 1 0 1

To the input (plaintext), apply initial permutation IP:

IP							
2	6	3	1	4	8	5	7

In the next steps, we will develop 4 bits with which to replace the left half of this "blue" result.

Input:
0 1 1 0 1 1 0 1

1 1 1 0 0 1 1 0

To right 4 bits of above result, apply expansion/permutation E/P (generating 8 bits from 4). The bit numbering is that of the 4-bit right-nibble, not of the 8-bit byte (e.g., indicated bit 2 refers to byte's bit 6).

E/P							
4	1	2	3	2	3	4	1

0 0 1 1 1 1 0 0

To right 4 bits of above result, apply expansion/permutation E/P (generating 8 bits from 4). The bit numbering is that of the 4-bit right-nibble, not of the 8-bit byte (e.g., indicated bit 2 refers to byte's bit 6).

E/P							
4	1	2	3	2	3	4	1

0 0 1 1 1 1 0 0

Upon above result, perform binary XOR operation with subkey K1:

K1							
1	0	1	0	0	1	0	0

1 0 0 1 1 0 0 0

Determine a row and a column from above XOR result. For the row, combine bits 1 and 4 and convert to decimal. For the column, combine bits 2 and 3 and convert to decimal.

Determine another row and column. For this second row, combine bits 5 and 8; for this second column, bits 6 and 7.

Identify the entry in s-box S0 at the first row/first column you determined. S0 shows it in decimal; convert it to binary (two bits). Enter those bits as the first half of the 4-bit number at right. Identify the entry in s-box S1 at the second row/second column you determined. Convert it to binary; enter those two bits as the second half of the number at right.

S0 =

	c0	c1	c2	c3
r0	1	0	3	2
r1	3	2	1	0
r2	0	2	1	3
r3	3	1	3	2

S1 =

	c0	c1	c2	c3
r0	0	1	2	3
r1	2	0	1	3
r2	3	0	1	0
r3	2	1	0	3

left nibble:
bits 1 & 4 -> 11 -> 3
bits 2 & 3 -> 00 -> 0
therefore, get from S0 row 3 col 0
result is 3 -> 11

right nibble:
bits 1 & 4 -> 10 -> 2
bits 2 & 3 -> 00 -> 0
therefore, get from S1 row 2 col 0
result is 3 -> 11

1 1 1 1

To above result, apply permutation P4:

P4			
2	4	3	1

1 1 1 1

Upon the above P4 result, perform binary XOR operation, combining it with the left 4-bits of our first result (application of IP to original plaintext input, blue cell above).	XOR with 1110
We are trying to replace the left half of that first result. These XOR result bits are the replacement bits for it.	0 0 0 1

Upon the above P4 result, perform binary XOR operation, combining it with the left 4-bits of our first result (application of IP to original plaintext input, blue cell above). We are trying to replace the left half of that first result. These XOR result bits are the replacement bits for it.	XOR with 1110 0 0 0 1
Rewrite that "blue" first result with its left half replaced. (Look it up, keep/copy its right half, use the preceding result as the new left half.)	0 0 0 1 0 1 1 0

Rewrite that "blue" first result with its left half replaced. (Look it up, keep/copy its right half, use the preceding result as the new left half.)	0 0 0 1 0 1 1 0
Swap the two 4-bit halves of the above (previous) result. In the next steps, we will again develop 4 replacement bits, and with them replace the left half of this "green" swap result. The steps will be the same ones used for that purpose already.	0 1 1 0 0 0 0 1

To right 4 bits of above swap result, apply expansion/permutation E/P (generating 8 bits from 4):

E/P							
4	1	2	3	2	3	4	1

1 0 0 0 0 0 1 0

Upon above result, perform binary XOR operation with subkey K2:

K2							
0	1	0	0	0	0	1	1

1 1 0 0 0 0 0 1

Determine a row and a column from above result. For the row, combine bits 1 and 4 and convert to decimal. For the column, combine bits 2 and 3 and convert to decimal.

Determine another row and column. For this second row, combine bits 5 and 8; for this second column, bits 6 and 7.

Identify the entry in s-box S0 at the first row/first column you determined. It's given in decimal; convert it to binary (two bits). Enter those bits as the first half of the 4-bit number at right. Identify the entry in s-box S1 at the second row/second column you determined. Convert it to binary; enter those two bits as the second half of the number at right.

S0 =

	c0	c1	c2	c3
r0	1	0	3	2
r1	3	2	1	0
r2	0	2	1	3
r3	3	1	3	2

S1 =

	c0	c1	c2	c3
r0	0	1	2	3
r1	2	0	1	3
r2	3	0	1	0
r3	2	1	0	3

left nibble:
bits 1 & 4 -> 10 -> 2
bits 2 & 3 -> 10 -> 2
therefore, get from S0 row 2 col 2
result is 1 -> 01

right nibble:
bits 1 & 4 -> 01 -> 1
bits 2 & 3 -> 00 -> 0
therefore, get from S1 row 1 col 0
result is 2 -> 10

0 1 1 0

To above result, apply permutation P4:

P4			
2	4	3	1

1 0 1 0

Upon the above P4 result, perform binary XOR operation, combining it with the left 4-bits of the earlier swap result (green cell above). We are trying to replace the left half of that swap result. These XOR result bits are the replacement bits for it.	XOR with 0110 1 1 0 0

Upon the above P4 result, perform binary XOR operation, combining it with the left 4-bits of the earlier swap result (green cell above). We are trying to replace the left half of that swap result. These XOR result bits are the replacement bits for it.	XOR with 0110 1 1 0 0
Rewrite that "green" swap result with its left half replaced. (Look it up, keep/copy its right half, use the preceding result as the new left half.)	1 1 0 0 0 0 0 1

Rewrite that "green" swap result with its left half replaced. (Look it up, keep/copy its right half, use the preceding result as the new left half.)	1 1 0 0 0 0 0 1
To above result, apply reverse of initial permutation IP, which is IP^{-1}: IP^{-1} 4 1 3 5 7 2 8 6 This result is ciphertext. It is the S-DES encryption of the plaintext input.	0 1 0 0 0 1 1 0