Subject Name:Network and Cyber Security
Subject Code:18KP1CSELCS1:A

Mrs.G.Muthamiz Selvi
Mrs.K.Hemalatha

Semester – I
MBE – 1

Hours - 6
Credit - 4

Network and Cyber Security (18KP1CSELCS1:A)

Objective: To provide application level knowledge in security concepts.

Unit - I : Introduction: The OSI Security Architecture – Security Attacks – Security Services – Security Mechanisms -A Model for Network Security. Classical Encryption Techniques: Symmetric Cipher Model – Steganography, Block Ciphers and the Data Encryption Standard: The Data Encryption Standard - Finite Fields: The Euclidean Algorithm.

Unit - II : Introduction to Number Theory : Prime Numbers - Fermat's and Euler's Theorems. Public Key Cryptography and RSA: Principles of Public-Key Cryptosystems : The RSA Algorithm. Key Management: Others Public-Key Cryptosystems. Key Management -Diffie-Hellman key exchange.

Unit - III : Message Authentication and Hash Functions: Authentication Requirements - Authentication functions – Message Authentication Codes - Hash functions. Digital Signatures and Authentication Protocols : Digital signatures :

Unit – IV: IP Security : IP Security Overview –IP security Architecture: Security Associations – SA Parameters – SA Selectors – Transport and Tunnel Modes . Authentication Header-Anti-Replay service. Intruders: Intrusion Detection – Honeypots - Password management – Password Protection - Malicious Software - Viruses and related threats.

Unit - V : - Cyber Security: Cyber Crime: Introduction – Various Cyber Crimes – National Cyber Security Policy (NCSP): Introduction – Analysis – Possible Impact and opportunities – Indian Cyber Space – Cyber Security Initiatives – Recommendations – Trends and Developments – IT Act: Introduction – IT Act Amendment 2008.

Text : Unit I-IV: "Cryptography and Network Security" Principles and Practices - William Stallings - Pearson Education – Fourth Edition- First Impression, 2006.

Chapters : 1, 2.1, 2.5, 3.2, 4.3, 8.1 - 8.2, 9, 10.1 - 10.2, 11.1-11.4 , 13.1 , 16.1 - 16.3, 18, 19.1. (Algorithms only. Problems should not be given from these topics)

Unit V: "Cyber Security" – Course Material Compiled by Dr. P.Cynthia Selvi, Head & Associate Professor, Department of Computer Science, Kunthavai Naacchiyar Government Arts College for Women (Autonomous), Thanjavur.

Reference

1. "Introduction to Cryptography and Network Security" – Behrouz A.Forouzan – McGraw Hill Higher Education – Special Indian Edition – 2008.
2. "Cryptography and Network Security" – Atul Kahate - Tata McGraw Hill – 2008.
3. "Cryptography: Origin to Recent Advancement" – Lambert Academic Publishing – 2011.

Head, Dept. of Computer Science,
Govt. Arts College (W)

UNIT - I

Introduction

The OSI Security Architecture

The Security Architecture for OSI, defines such a systematic approach.The OSI security architecture is useful to managers as a way of organizing the task of providing security.

The OSI security architecture focuses on:

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

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

● Security service: A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization.

Security Attacks

Two types:

Passive attacks - A passive attack attempts to learn or make use of information from the system but does not affect system resources.

Active attacks - An active attack attempts to alter system resources or affect their operation.

Passive Attacks

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted.

Two types of passive attacks:

1.Release of message contents

2.Traffic analysis.

Release of message contents:

The release of message contents is easily understood. A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information.

Traffic Analysis:

A way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message.

Active Attacks

Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories:

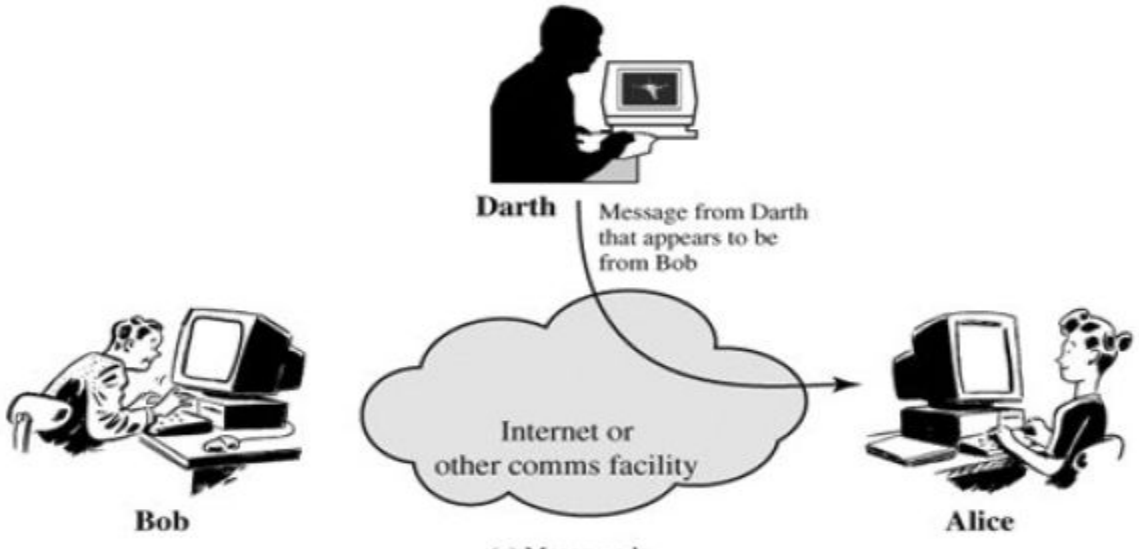
1.Masquerade

2.Replay

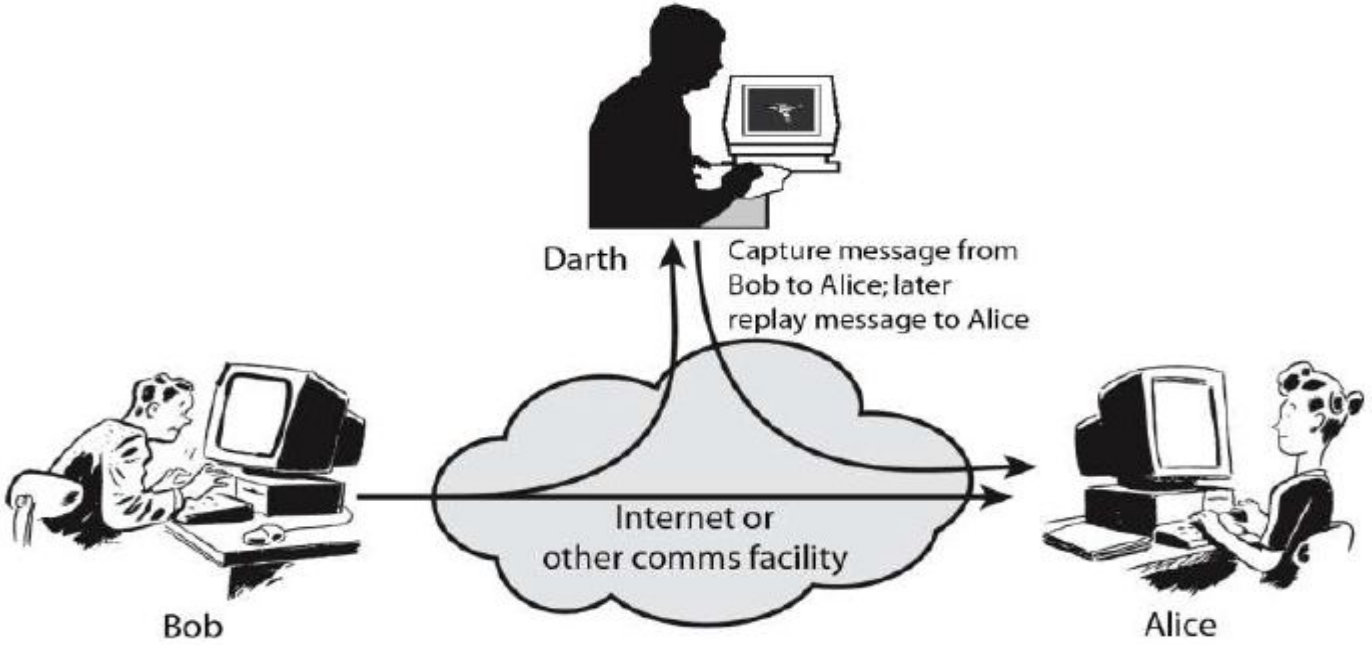
3.Modification of messages

4.Denial of service.

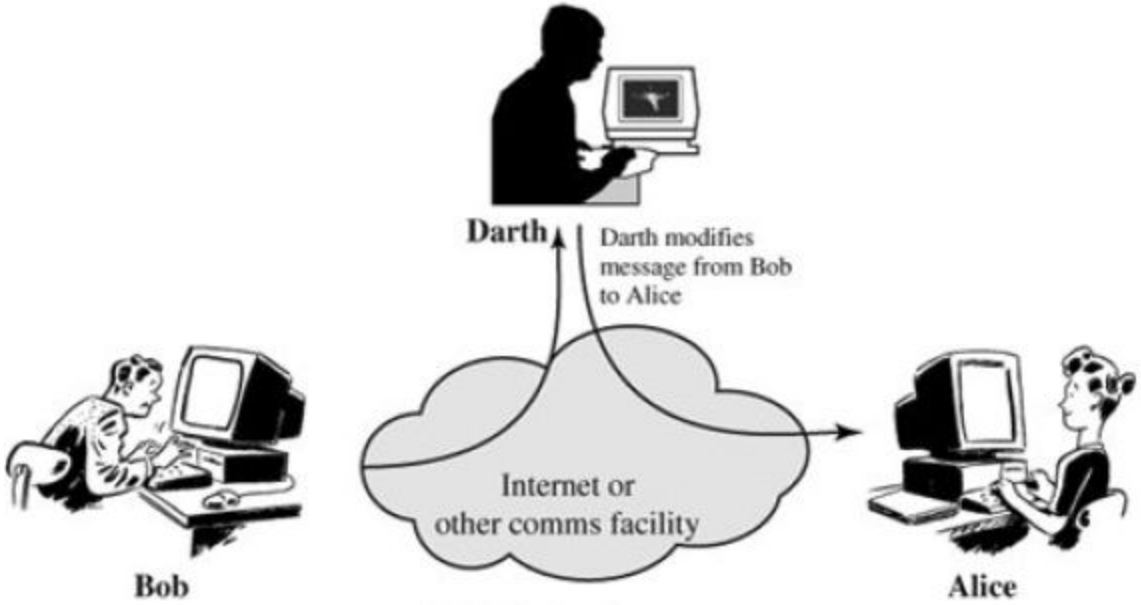
Masquerade: A **masquerade attack** is an **attack** that uses a fake identity, such as a network identity, to gain unauthorized access to personal computer information through legitimate access identification.



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

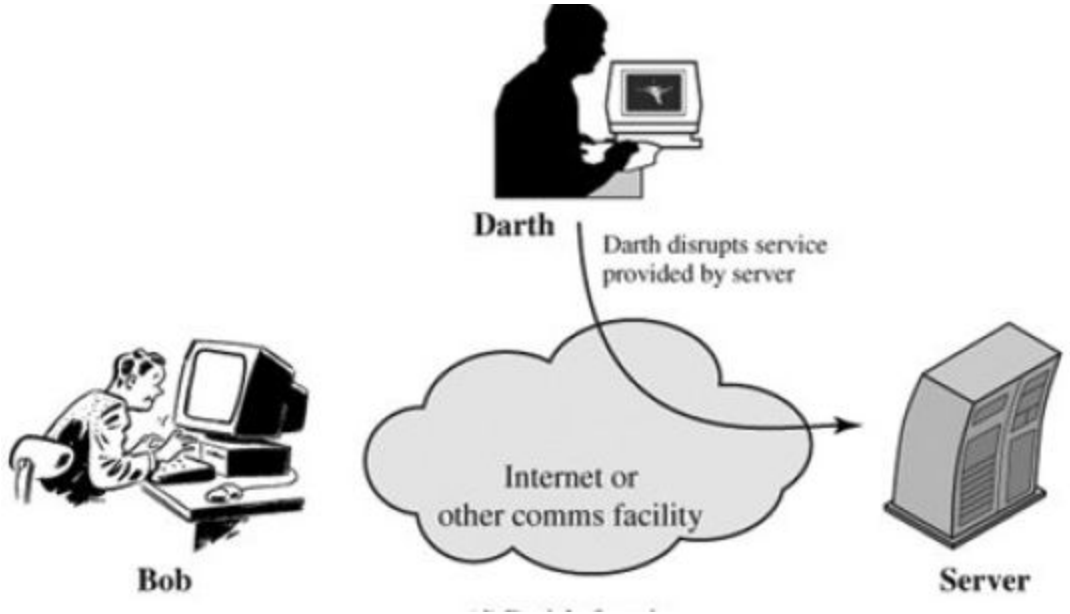


Modification of messages: The portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect.



Denial of service : Prevents or inhibits the normal use or management of communications facilities.This attack may have a specific target.

Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.



Security Services

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:

1.AUTHENTICATION

2.ACCESS CONTROL

3.DATA CONFIDENTIALITY

4.DATA INTEGRITY

5.NONREPUDIATION

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

ACCESS CONTROL-The prevention of unauthorized use of a resource.

DATA CONFIDENTIALITY-The protection of data from unauthorized disclosure.

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

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

Security Mechanisms

SPECIFIC SECURITY MECHANISMS

May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.

- Encipherment
- Routing Control
- Digital Signature
- Notarization
- Access Control
- Authentication Exchange
- Traffic Padding
- Data Integrity

PERVASIVE SECURITY MECHANISMS

Mechanisms that are not specific to any particular OSI security service or protocol layer.

1.Trusted Functionality 2.Security Audit Trail

3.Security Recovery 4.Security Label

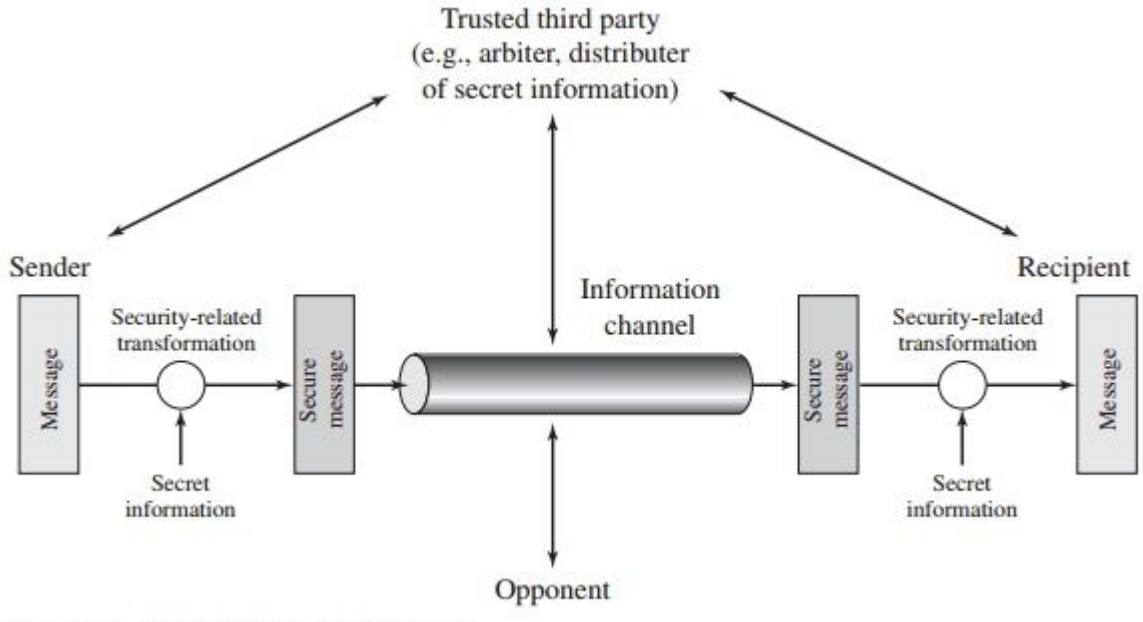
5.Event Detection

A Model for Network Security

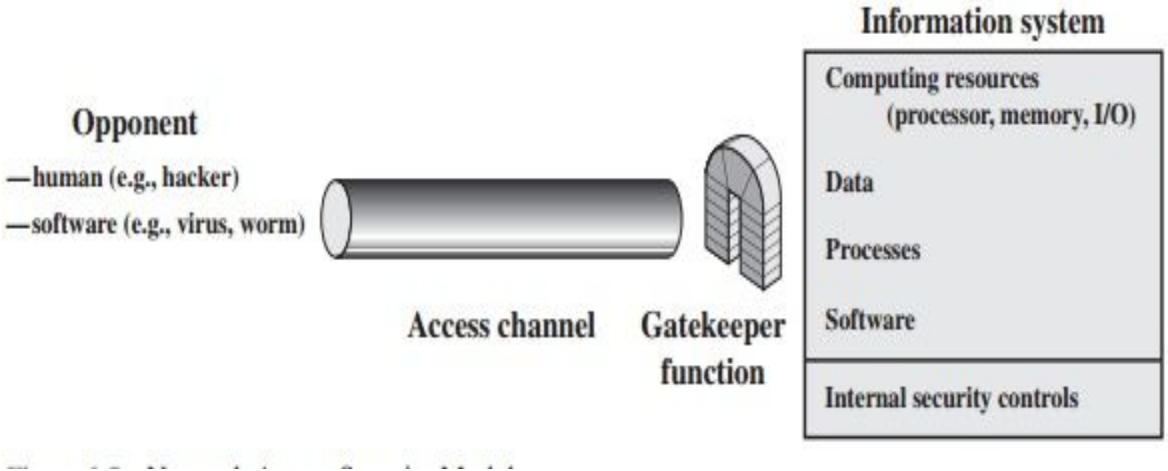
A security-related transformation on the information to be sent.

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

Model for Network Security



Network Access Security Model



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

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

Programs can present two kinds of threats:

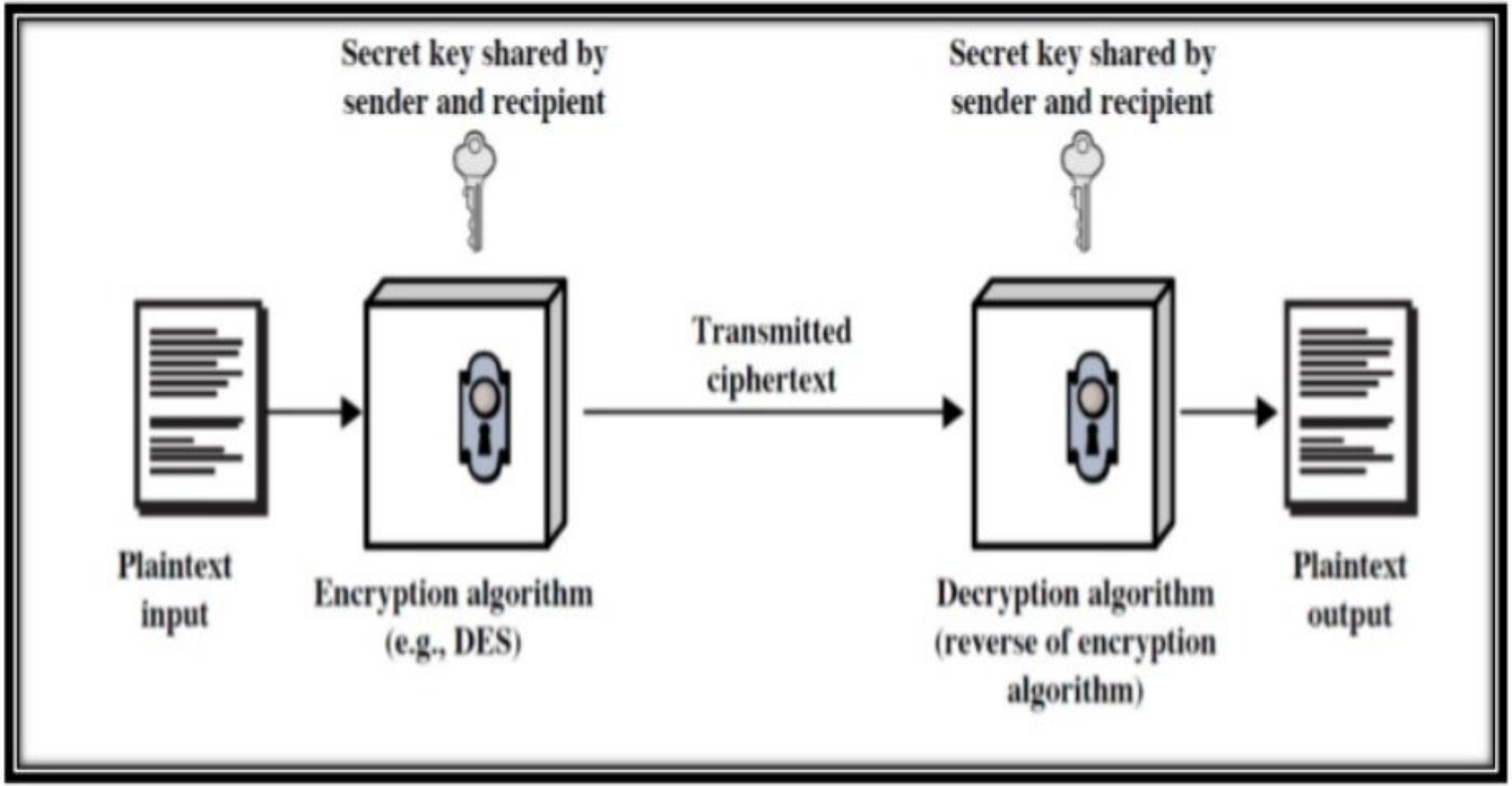
- **Information access threats:** Intercept or modify data on behalf of users who should not have access to that data.

- **Service threats:** Exploit service flaws in computers to inhibit use by legitimate users.

Symmetric Cipher Model

- A symmetric encryption scheme has five ingredients:

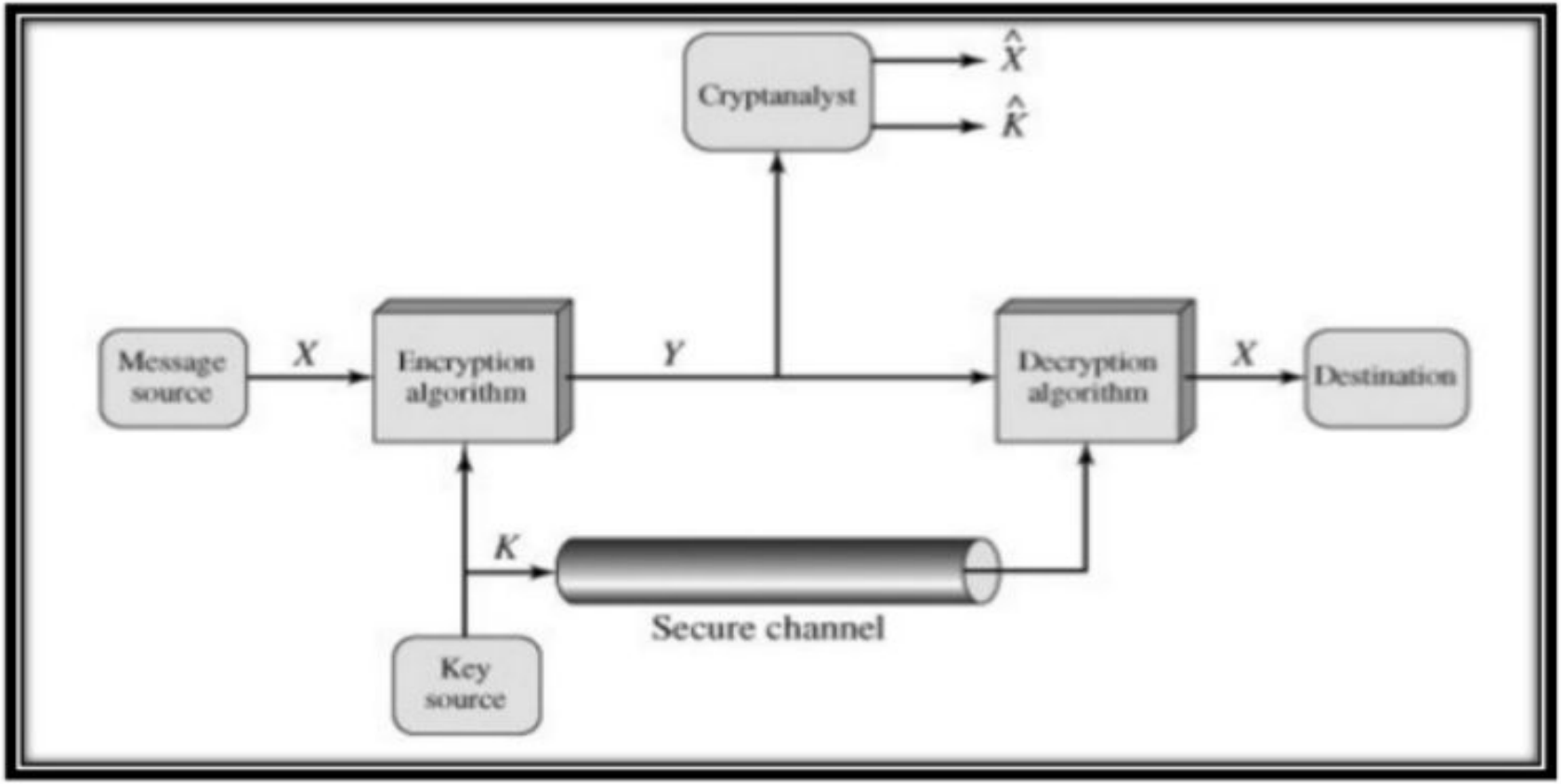
- Plaintext: original message to be encrypted.
- Ciphertext: the encrypted message.
- Encryption algorithm: The encryption algorithm performs various substitutions and transformations on the plaintext.
- Decryption algorithm: This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.
- Secret key: A secret key is the piece of information or parameter that is used to encrypt and decrypt messages in a symmetric, or secret-key, encryption.



Simplified Model of Symmetric encryption

There are two requirements for secure use of symmetric encryption:

- a strong encryption algorithm.
- a secret key known only to sender / receiver.



Model of Conventional Cryptosystem

Symmetric Encryption

Mathematically:

$Y = EK(X)$ or $Y = E(K, X)$

$X = DK(Y)$ or $X = D(K, Y)$

- X = plaintext
- Y = ciphertext
- K = secret key
- E = encryption algorithm
- D = decryption algorithm
- Both E and D are known to public

Cryptography

Cryptographic systems are characterized along three independent dimensions:

- The type of operations used for transforming plaintext to ciphertext.
- The number of keys used.
- The way in which the plaintext is processed.

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.

Kirchhoff's principle: the adversary knows all details about a cryptosystem except the secret key.

Two general approaches:

- brute-force attack
- non-brute-force attack (cryptanalytic attack)

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.

Cryptanalytic Attacks

Type of Attack	Known to Cryptanalyst
Ciphertext Only	• Encryption algorithm • Ciphertext
Known Plaintext	• Encryption algorithm • Ciphertext • One or more plaintext–ciphertext pairs formed with the secret key
Chosen Plaintext	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	• Encryption algorithm • Ciphertext • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Unconditional Security

- An encryption scheme is unconditionally secure if the ciphertext generated by the scheme does not contain enough information to determine uniquely the corres-ponding plaintext, no matter how much ciphertext is available.
- Therefore, all that the users of an encryption algorithm can strive for is an algorithm that meets one or both of the following criteria:

  (1) The cost of breaking the cipher exceeds the value of the encrypted information.

  (2) The time required to break the cipher exceeds the useful lifetime of the information.
- An encryption scheme is said to be computationally secure if either of the foregoing two criteria are met.

Steganography

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

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

- 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 to the paper.
- Pin punctures: Small pin punctures on selected letters are ordinarily not visible unless the paper is held up in front of a light.

➢Typewriter correction ribbon: Used between lines typed with a black ribbon, the results of typing with the correction tape are visible only under a strong light.

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

DATA ENCRYPTION STANDARD

The most widely used encryption scheme.

Adopted in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST).

The algorithm itself is referred to as the Data Encryption Algorithm (DEA).

For DES, 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.

HISTORY

In 1971 ,IBM developed an algorithm, named LUCIFER which operates on a block of 64 bits ,using a 128 -bit key.

Walter Tuchman, an IBM researcher,refined LUCIFER and reduced the key size to 56 -bit,to fit on a chip.

In 1973, the National Bureau of Standards (NBS) issued a request for proposals for a national cipher standard. This was by far the best algorithm proposed and was adopted in 1977 as the Data Encryption Standard.

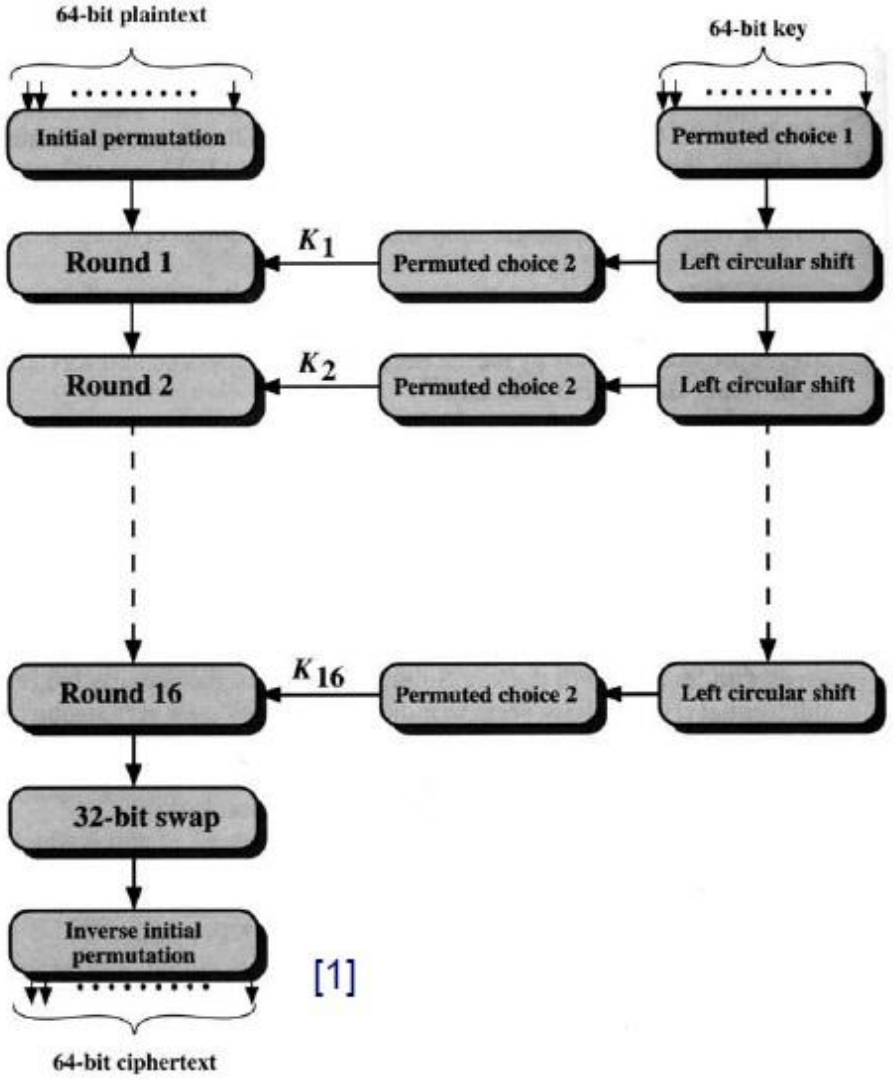
DES Encryption

Any encryption scheme, 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.

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 last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key.

Encryption



[1]

Cont...

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.

The permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits.

Initial Permutation:

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.

Encryption (IP, IP^{-1})

- IP

Bit	0	1	2	3	4	5	6	7
1	58	50	42	34	26	18	10	2
9	60	52	44	36	28	20	12	4
17	62	54	46	38	30	22	14	6
25	64	56	48	40	32	24	16	8
33	57	49	41	33	25	17	9	1
41	59	51	43	35	27	19	11	3
49	61	53	45	37	29	21	13	5
57	63	55	47	39	31	23	15	7

- IP^{-1}

Bit	0	1	2	3	4	5	6	7
1	40	8	48	16	56	24	64	32
9	39	7	47	15	55	23	63	31
17	38	6	46	14	54	22	62	30
25	37	5	45	13	53	21	61	29
33	36	4	44	12	52	20	60	28
41	35	3	43	11	51	19	59	27
49	34	2	42	10	50	18	58	26
57	33	1	41	9	49	17	57	25

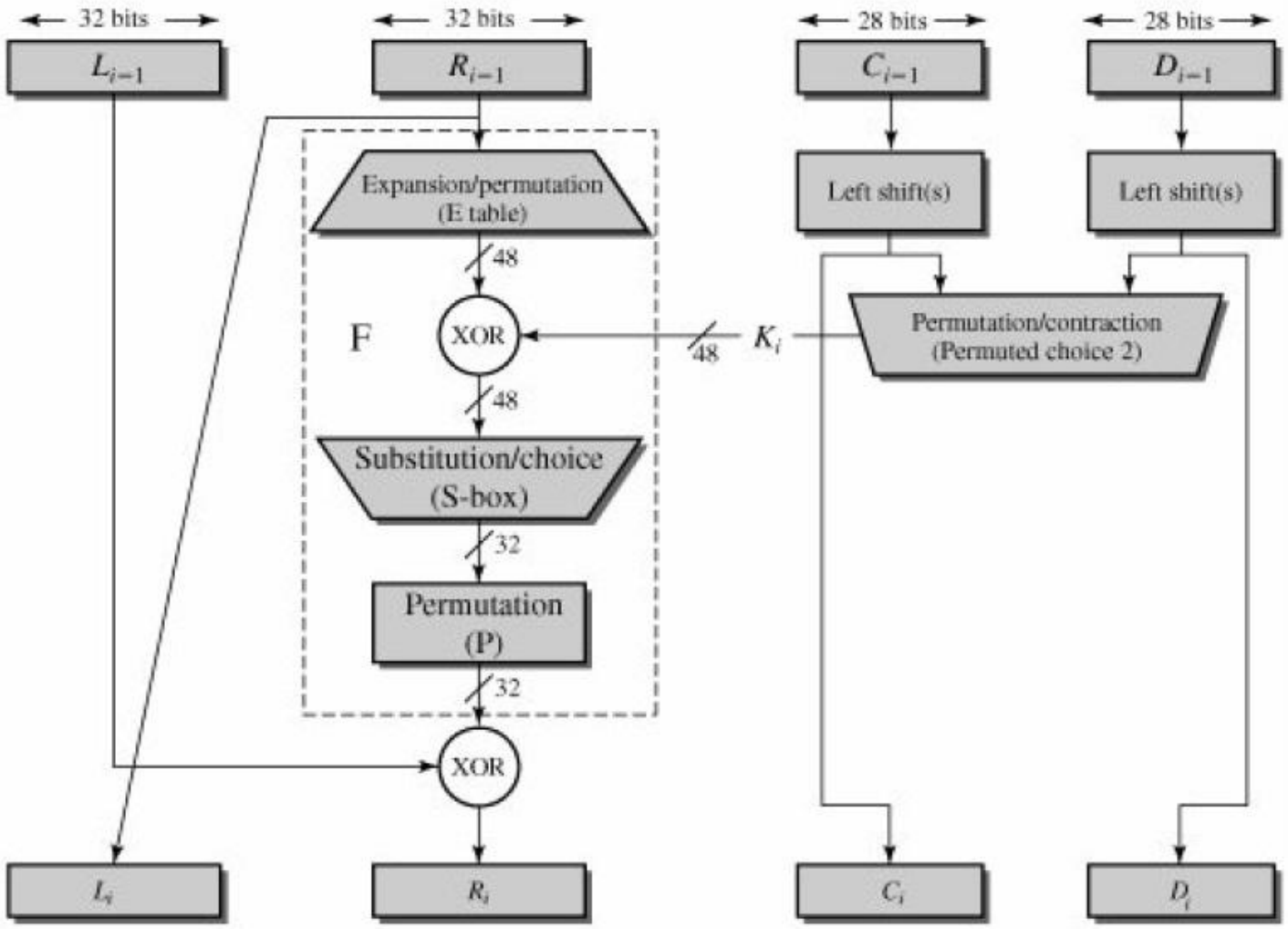
- Note: $IP(IP^{-1}) = IP^{-1}(IP) = I$

Details of Single Round:

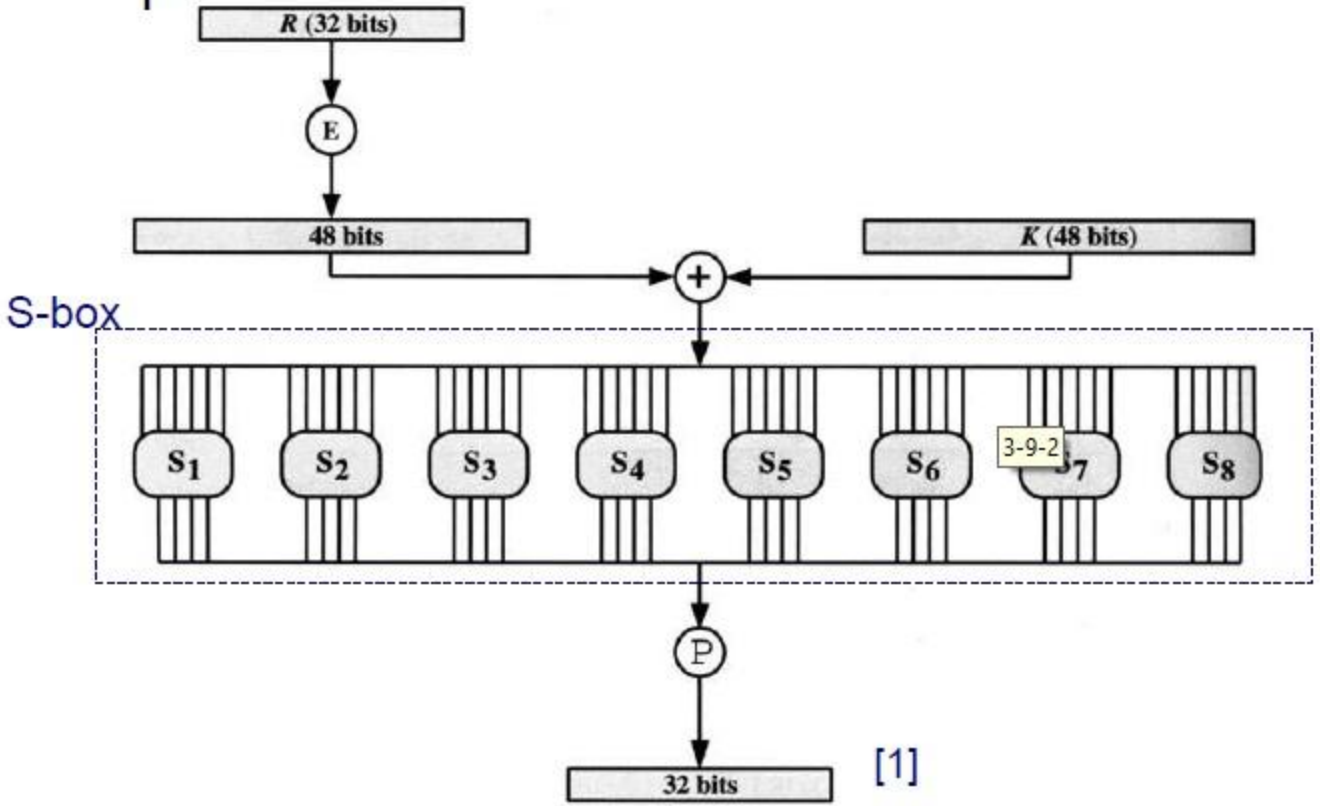
The left and right halves of each 64-bit intermediate value are treated as separate 32-bit quantities, labeled L (left) and R (right). The overall processing at each round can be summarized in the following formulas:

$L_i = R_{i-1}$

$R_i = L_{i-1} \times F(R_{i-1}, K_i)$


32 bits
32 bits
28 bits
28 bits
L_{i-1}
R_{i-1}
C_{i-1}
D_{i-1}
Expansion/permutation
(E table)
Left shift(s)
Left shift(s)
48
F
XOR
48
K_i
Permutation/contraction
(Permuted choice 2)
48
Substitution/choice
(S-box)
32
Permutation
(P)
32
XOR
L_i
R_i
C_i
D_i

F


R (32 bits)
E
48 bits
K (48 bits)
S-box
S1
S2
S3
S4
S5
S6
S7
S8
3-9-2
P
32 bits

[1]

S-box

S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Key Generation

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 ChoiceOne.

The resulting 56-bit key is then treated as two 28-bit quantities, labeled C0 and D0. At each round, Ci-1 and Di-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.

DES Decryption

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

this might provide a way to reduce the size of the plaintext or key space to be searched

The Avalanche Effect

A desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the ciphertext.

In particular, a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the ciphertext.

The Euclidean Algorithm

The Euclidean Algorithm finds the greatest common divisor of two integers a and b.

For example, If we want to find gcd(287, 91), we divide 287 by 91:

$$287 = 91*3 + 14$$

We know that for integers a, b and c, if a | b and a | c, then a | (b + c).

Therefore, any divisor of 287 and 91 must also be a divisor of 287 - 91*3 = 14.

Consequently, gcd(287, 91) = gcd(14, 91).

In the next step, we divide 91 by 14:

$91 = 14*6 + 7$

This means that gcd(14, 91) = gcd(14, 7).

So we divide 14 by 7:

$14 = 7*2 + 0$

We find that 7 | 14, and thus gcd(14, 7) = 7.

Therefore, gcd(287, 91) = 7.

The Euclidean algorithm can be based on the following theorem: For any nonnegative integer a and any positive integer b,

$$\gcd(a, b) = \gcd(b, a \bmod b)$$

$$\gcd(55, 22) = \gcd(22, 55 \bmod 22) = \gcd(22, 11) = 11$$

Euclidean Algorithm	
Calculate	**Which satisfies**
$r_1 = a \bmod b$	$a = q_1 b + r_1$
$r_2 = b \bmod r_1$	$b = q_2 r_1 + r_2$
$r_3 = r_1 \bmod r_2$	$r_1 = q_3 r_2 + r_3$
•	•
•	•
•	•
$r_n = r_{n-2} \bmod r_{n-1}$	$r_{n-2} = q_n r_{n-1} + r_n$
$r_{n+1} = r_{n-1} \bmod r_n = 0$	$r_{n-1} = q_{n+1} r_n + 0$ $d = \gcd(a, b) = r_n$

UNIT - II

Prime Numbers

- prime numbers only have divisors of 1 and self
  - they cannot be written as a product of other numbers
  - note: 1 is prime, but is generally not of interest
- eg. 2,3,5,7 are prime, 4,6,8,9,10 are not
- prime numbers are central to number theory
- list of prime number less than 200 is:

```
2  3  5  7  11  13  17  19  23  29  31  37  41  43  47  53  59
61  67  71  73  79  83  89  97  101  103  107  109  113  127
131  137  139  149  151  157  163  167  173  179  181  191
193  197  199
```

Prime Factorisation

- to **factor** a number n is to write it as a product of other numbers: n=a x b x c
- note that factoring a number is relatively hard compared to multiplying the factors together to generate the number
- the **prime factorisation** of a number n is when its written as a product of primes
  - eg. $91=7x13$; $3600=2^4x3^2x5^2$

$$a = \prod_{p \in \mathrm{P}} p^{a_p}$$

Relatively Prime Numbers & GCD

- two numbers a, b are **relatively prime** if have **no common divisors** apart from 1
  - eg. 8 & 15 are relatively prime since factors of 8 are 1,2,4,8 and of 15 are 1,3,5,15 and 1 is the only common factor
- conversely can determine the greatest common divisor by comparing their prime factorizations and using least powers
  - eg. $300=2^1 \times 3^1 \times 5^2$ $18=2^1 \times 3^2$ hence $GCD(18,300)=2^1 \times 3^1 \times 5^0=6$

Fermat's Theorem

- $a^{p-1} = 1 \pmod{p}$
  - where p is prime and $\gcd(a,p)=1$
- also known as Fermat's Little Theorem
- also have: $a^{p} = a \pmod{p}$
- useful in public key and primality testing

Euler Totient Function ø(n)

- when doing arithmetic modulo n
- **complete set of residues** is: 0..n-1
- **reduced set of residues** is those numbers (residues) which are relatively prime to n
  - eg for n=10,
  - complete set of residues is {0,1,2,3,4,5,6,7,8,9}
  - reduced set of residues is {1,3,7,9}
- number of elements in reduced set of residues is called the **Euler Totient Function ø(n)**

Euler Totient Function ø(n)

- to compute ø(n) need to count number of residues to be excluded
- in general need prime factorization, but
  - for p (p prime) ø(p)=p-1
  - for p.q (p,q prime) ø(p.q)=(p-1)x(q-1)
- eg.

  ø(37) = 36

  ø(21) = (3-1)x(7-1) = 2x6 = 12

Private-Key Cryptography

- traditional **private/secret/single key** cryptography uses **one** key
- shared by both sender and receiver
- if this key is disclosed communications are compromised
- also is **symmetric**, parties are equal
- hence does not protect sender from receiver forging a message and claiming it's sent by sender (repudiation problem)

Public-Key Cryptography

- probably most significant advance in the 3000 year history of cryptography
- uses **two** keys – a public & a private key
- **asymmetric** since parties are **not** equal
- uses clever application of number theoretic concepts to make it work
- complements **rather than** replaces private key cryptography (efficiency reasons)

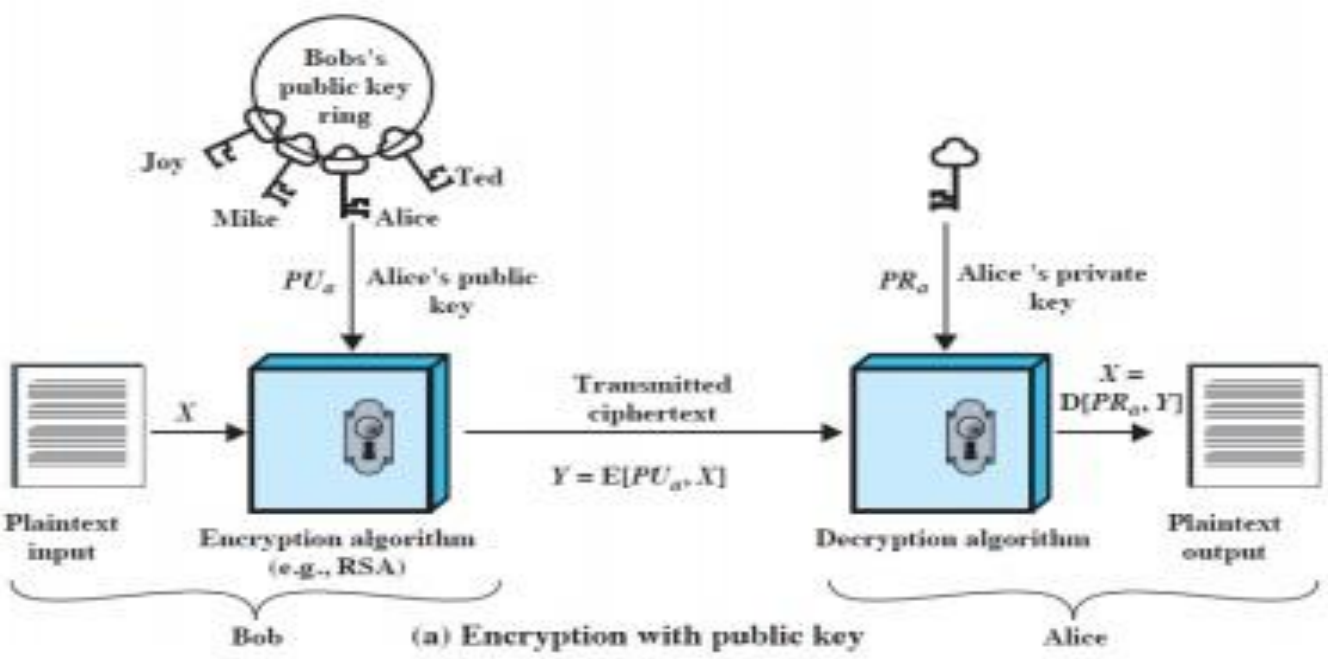
Why Public-Key Cryptography?

- developed to address two key issues:
  - **key distribution** – how to have secure communications in general without having to trust a KDC with your key
  - **digital signatures** – how to verify a message comes intact from the claimed sender
- public invention due to Whitfield Diffie & Martin Hellman at Stanford Uni in 1976
  - known earlier in classified community (NSA (60's (claimed)), CESG (1970 (documented)))

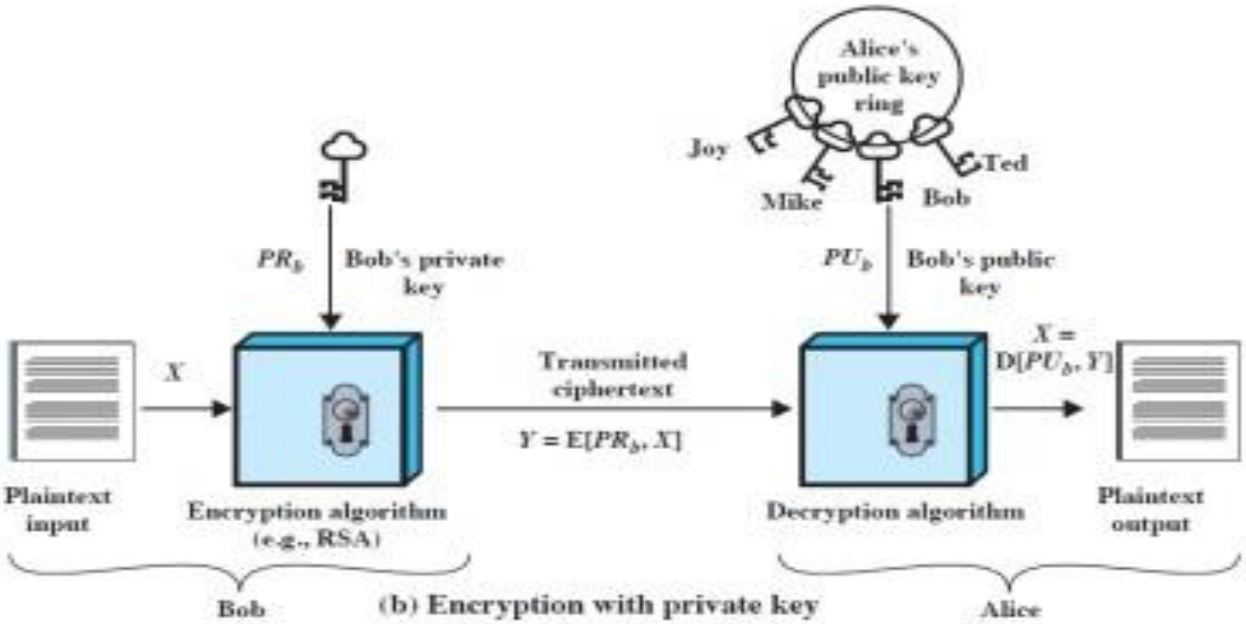
Public-Key Cryptography

- **public-key/two-key/asymmetric** cryptography involves the use of **two** keys:
  - a **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
  - a related **private-key**, known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures**
- **infeasible to determine private key from public (requires solving a hard problem)**
- is **asymmetric** because
  - those who **encrypt** messages or **verify** signatures ***cannot*** **decrypt** messages or **create** signatures

Public-Key Cryptography



Public-Key Cryptography

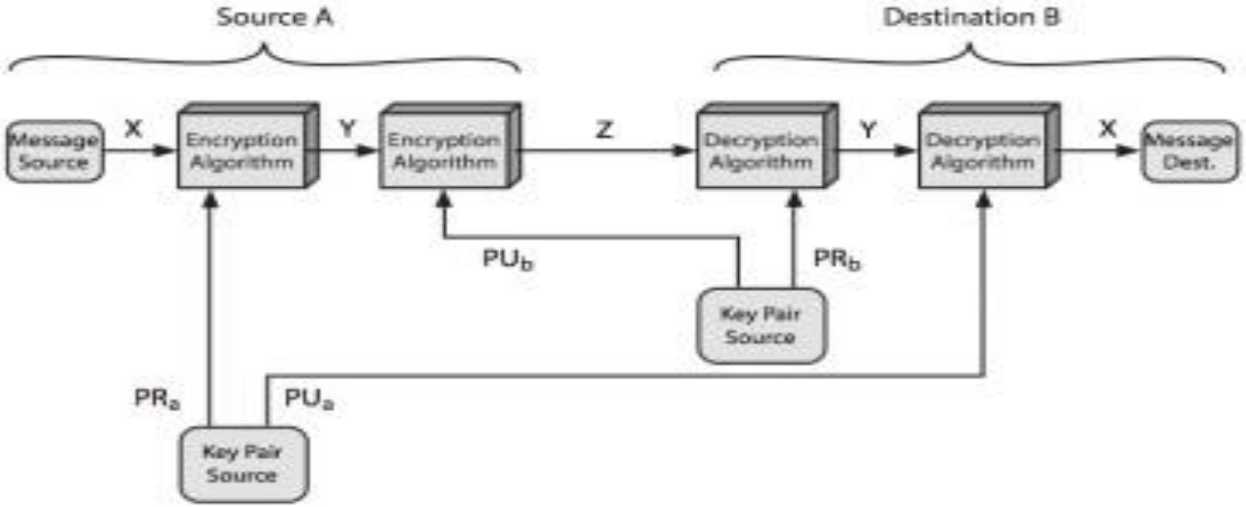


(b) Encryption with private key

Symmetric vs Public-Key

Conventional Encryption	Public-Key Encryption
Needed to Work: 1. The same algorithm with the same key is used for encryption and decryption. 2. The sender and receiver must share the algorithm and the key. *Needed for Security:* 1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	*Needed to Work:* 1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption. 2. The sender and receiver must each have one of the matched pair of keys (not the same one). *Needed for Security:* 1. One of the two keys must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

Public-Key Cryptosystems



Combining secrecy and authentication

Public-Key Applications

- can classify uses into 3 categories:
  - **encryption/decryption** (provide secrecy)
  - **digital signatures** (provide authentication)
  - **key exchange** (of session keys)
- some algorithms are suitable for all uses, others are specific to one

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

Public-Key Requirements

- Public-Key algorithms rely on two keys where:
  - it is computationally infeasible to find decryption key knowing only algorithm & encryption key
  - it is computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
  - either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms)
- these are formidable requirements which only a few algorithms have satisfied

Public-Key Requirements

- need a trapdoor one-way function
- one-way function has
  - $Y = f(X)$ easy
  - $X = f^{-1}(Y)$ infeasible
- a trap-door one-way function has
  - $Y = f_k(X)$ easy, if k and X are known
  - $X = f_k^{-1}(Y)$ easy, if k and Y are known
  - $X = f_k^{-1}(Y)$ infeasible, if Y known but k not known
- a practical public-key scheme depends on a suitable trap-door one-way function

RSA

- by Rivest, Shamir & Adleman of MIT in 1977
- best known & widely used public-key scheme
- based on exponentiation in a finite (Galois) field over integers modulo a prime
  - nb. exponentiation takes $O((\log n)^3)$ operations (easy)
- uses large integers (eg. 1024 bits, or 2048 bits)
- security due to cost of factoring large numbers
  - nb. factorization takes $O(e^{\log n \log \log n})$ operations (superpolynomial, hard)

RSA En/decryption

- to encrypt a message M the sender:
  - obtains **public key** of recipient $PU = \{e, n\}$
  - computes: $C = M^e \bmod n$, where $0 \leq M < n$
- to decrypt the ciphertext C the owner:
  - uses their private key $PR = \{d, n\}$
  - computes: $M = C^d \bmod n$
- note that the message M must be smaller than the modulus n (block if needed)

RSA Key Setup

- each user generates a public/private key pair by:
- selecting two large primes at random: p, q
- computing their system modulus $n = p.q$
  - note $\phi(n) = (p-1)(q-1)$
- selecting at random the encryption key e
  - where $1 < e < \phi(n)$, $\gcd(e, \phi(n)) = 1$
- solve following equation to find decryption key d
  - $e.d = 1 \mod \phi(n)$ and $0 \leq d \leq n$
- publish their public encryption key: $PU = \{e, n\}$
- keep secret private decryption key: $PR = \{d, n\}$

Key Generation	
Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$d = e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption	
Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption	
Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

The RSA Algorithm

RSA Example - Key Setup

1. Select primes: $p = 17$; $q = 11$
2. Calculate $n = pq = 17 \times 11 = 187$
3. Calculate $\phi(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. Select e: $\gcd(e, 160) = 1$; choose $e = 7$
5. Derive d: $de = 1 \bmod 160$ and $d < 160$
   Get $d = 23$ since $23 \times 7 = 161 = 10 \times 160 + 1$
6. Publish public key: $PU = \{7, 187\}$
7. Keep private key secret: $PR = \{23, 187\}$

RSA Example - En/Decryption

- sample RSA encryption/decryption is:
- given message $M = 88$ (NB. $88 < 187$)
- encryption:

  $C = 88^7 \bmod 187 = 11$
- decryption:

  $M = 11^{23} \bmod 187 = 88$

Key Management:

In cryptography it is a very tedious task to distribute the public and private key between sender and receiver. If key is known to the third party (forger/eavesdropper) then the whole security mechanism becomes worthless. So, there comes the need to secure the exchange of keys.

There are 2 aspects for Key Management:

1. Distribution of public keys.
2. Use of public-key encryption to distribute secret.

Distribution of Public Key:

Public key can be distributed in 4 ways: These are explained as following below.

1. **Public Announcement**
2. **Publicly Available Directory**
3. **Public Key Authority**
4. **Public Key Certificates**

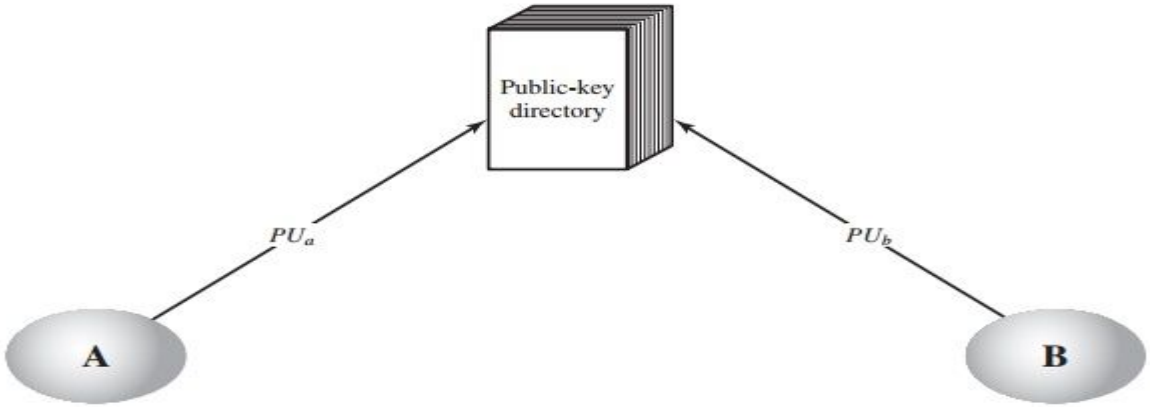
Public Announcement:

Here the public key is broadcasted to everyone. Major weakness of this method is forgery. Anyone can create a key claiming to be someone else and broadcast it. Until forgery is discovered can masquerade as claimed user.

Public Available Directory:

In this type, the public key is stored at a public directory. Directories are trusted here, with properties like Participant Registration, access and allow to modify values at any time, contains entries like {name, public-key}.

Directories can be accessed electronically still vulnerable to forgery or tampering.



Public Key Authority:

It is similar to the directory but, improve security by tightening control over distribution of keys from directory. It requires users to know public key for the directory. Whenever the keys are needed, a real-time access to directory is made by the user to obtain any desired public key securely.

Public Key Certificates

This time authority provides a certificate (which binds identity to the public key) to allow key exchange without real-time access to the public authority each time. The certificate is accompanied with some other info such as period of validity, rights of use etc. All of this content is signed by the trusted Public-Key or Certificate Authority (CA) and it can be verified by anyone possessing the authority’s public-key.

Diffie-Hellman Key Exchange

- first public-key type scheme proposed
- by Diffie & Hellman in 1976 along with the exposition of public key concepts
  - note: now know that Williamson (UK CESG) secretly proposed the concept in 1970
- is a practical method for public exchange (really **creation**) of a secret key
- used in a number of commercial products

Diffie-Hellman Key Exchange

- a public-key distribution scheme
  - cannot be used to exchange an arbitrary message
  - rather it can establish a common key
  - known only to the two participants
- value of key depends on the participants (and their private and public key information)
- based on exponentiation in a finite (Galois) field (modulo a prime or a polynomial) – easy
- security relies on the difficulty of computing discrete logarithms (similar to factoring) – hard

Diffie-Hellman Setup

- all users agree on global parameters:
  - large prime integer or polynomial q
  - a being a primitive root mod q
- each user (eg. A) generates their key
  - chooses a secret key (number): $x_A < q$
  - compute their **public key**: $y_A = a^{x_A} \bmod q$
- each user makes public that key y_A

Diffie-Hellman Key Exchange

- shared session key for users A & B is K_{AB}:

  $K_{AB} = a^{x_A \cdot x_B} \bmod q$

  $= y_A^{x_B} \bmod q$ (which **B** can compute)

  $= y_B^{x_A} \bmod q$ (which **A** can compute)
- K_{AB} is used as session key in private-key encryption scheme between Alice and Bob
- if Alice and Bob subsequently communicate, they will have the **same** key as before, unless they choose new public-keys
- attacker needs an x, must solve discrete log

Global Public Elements

q	prime number
α	$\alpha < q$ and α a primitive root of q

User A Key Generation

Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{X_A} \bmod q$

User B Key Generation

Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{X_B} \bmod q$

Calculation of Secret Key by User A

$K = (Y_B)^{X_A} \bmod q$

Calculation of Secret Key by User B

$K = (Y_A)^{X_B} \bmod q$

Diffie-Hellman Example

- users Alice & Bob who wish to swap keys:
- agree on prime $q = 353$ and $a = 3$
- select random secret keys:
  - A chooses $x_A = 97$, B chooses $x_B = 233$
- compute respective public keys:
  - $y_A = 3^{97} \bmod 353 = 40$ (Alice)
  - $y_B = 3^{233} \bmod 353 = 248$ (Bob)
- compute shared session key as:
  - $K_{AB} = y_B^{x_A} \bmod 353 = 248^{97} = 160$ (Alice)
  - $K_{AB} = y_A^{x_B} \bmod 353 = 40^{233} = 160$ (Bob)

UNIT - III

Message Authentication

- protecting message content (ie secrecy) by encrypting the message
- now consider
  - how to protect message integrity (ie protection from modification)
  - confirming the identity of the sender
- then three alternative functions used:
  - message encryption (the ciphertext itself is the authenticator)
  - message authentication code (MAC)
  - hash function

Security Attacks

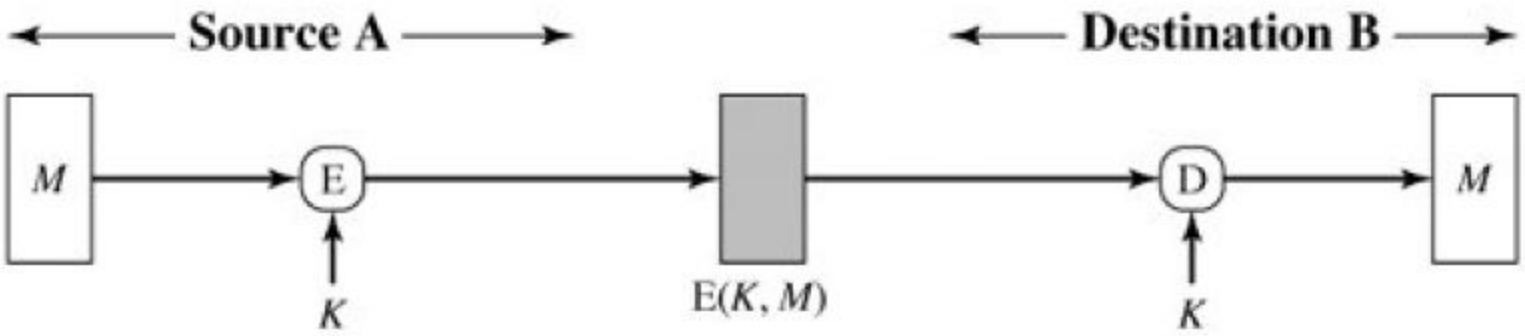
- disclosure of message contents
- traffic analysis (discover the pattern)
- Masquerade (insert a msg from a fraudulent source)
- content modification
- sequence modification (insert, delete, reorder)
- timing modification (delay or replay)
- source repudiation (denial of a transmission)
- destination repudiation (denial of a receipt)

Message Encryption

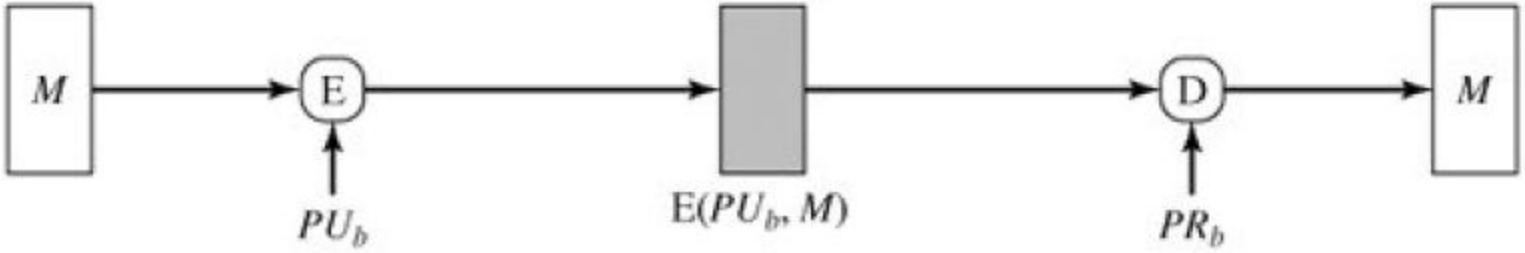
- message encryption by itself also provides a measure of authentication
- if symmetric encryption is used then:
  - receiver know sender must have created it
  - since only sender and receiver now key used
  - know content cannot of been altered
  - if message has suitable structure, redundancy or a checksum to detect any changes

Message Encryption

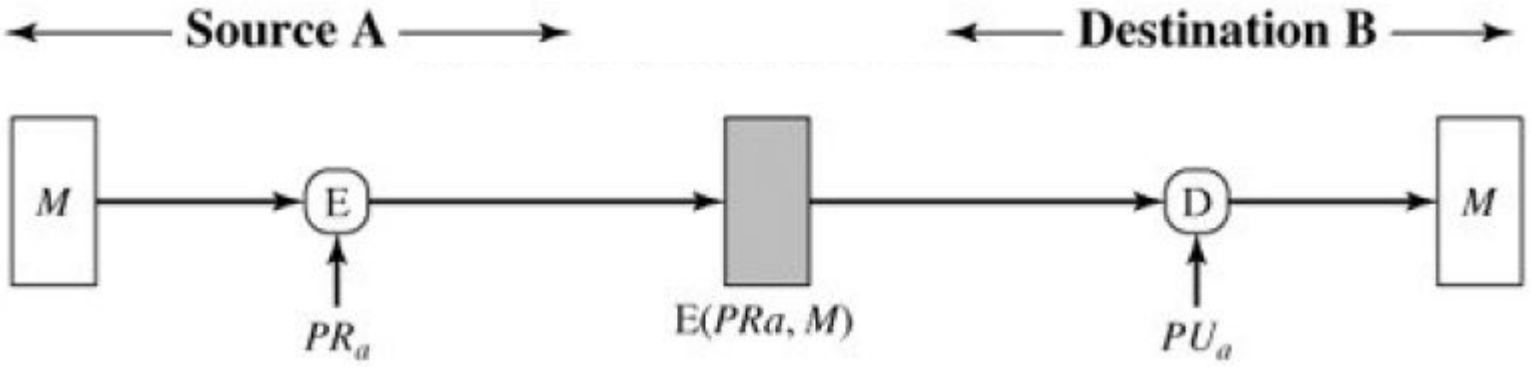
- if public-key encryption is used:
  - encryption provides no confidence of sender
  - since anyone potentially knows public-key
  - however if
    - sender **signs** message using their private-key
    - then encrypts with recipients public key
    - have both secrecy and authentication
  - again need to recognize corrupted messages
  - but at cost of two public-key uses on message



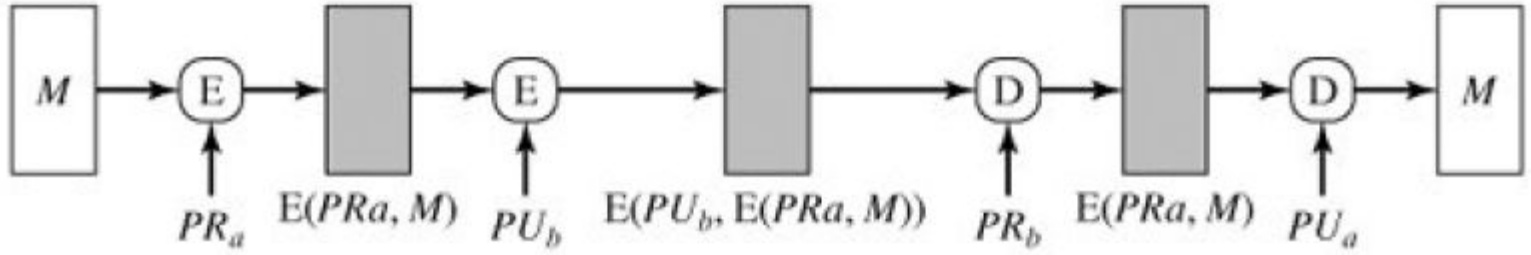
(a) Symmetric encryption: confidentiality and authentication



(b) Public-key encryption: confidentiality



(c) Public-key encryption: authentication and signature

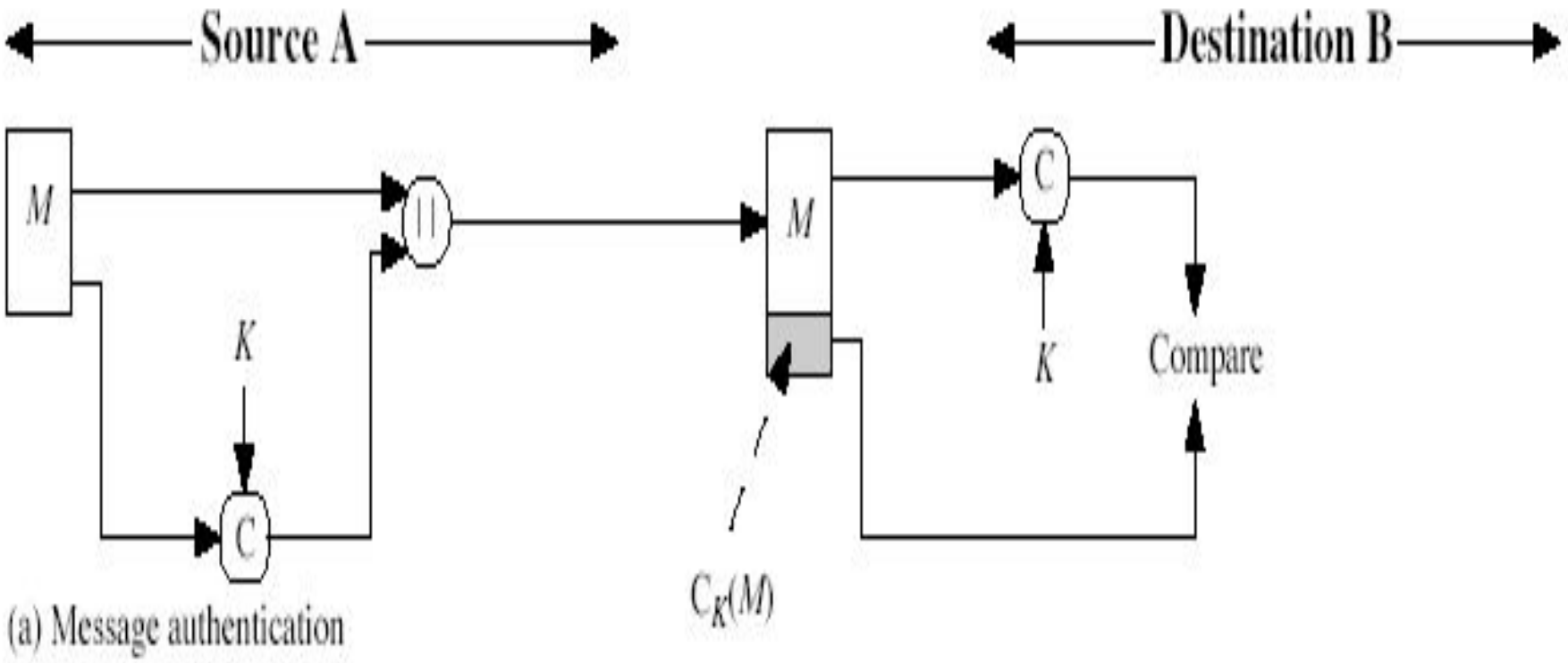


(d) Public-key encryption: confidentiality, authentication, and signature

Message Authentication Code (MAC)

- generated by an MAC function C that creates a small fixed-sized block
  - depending on both message M and a shared secret key K, $MAC=C_K(M)$
  - MAC is appended to the message M
- receiver performs same computation on message and checks it matches the MAC
- provides assurance that message is unaltered and comes from sender

Message Authentication Code



(a) Message authentication

Message Authentication Codes

- can also use encryption for secrecy
  - generally use separate keys for each
  - can compute MAC either before or after encryption
  - is generally regarded as better done before
- why use a MAC?
  - MAC is much less expensive than en/decryption
  - sometimes only authentication is needed
  - One end with a heavy load, check MAC selectively

MAC Properties

- a MAC is a cryptographic checksum

  $MAC = C_K(M)$

  – condenses a variable-length message M
  – using a secret key K
  – to a fixed-sized authenticator
- is a many-to-one function
  – potentially many messages have same MAC
    - 100-bit M, and 20-bit MAC

Requirements for MACs

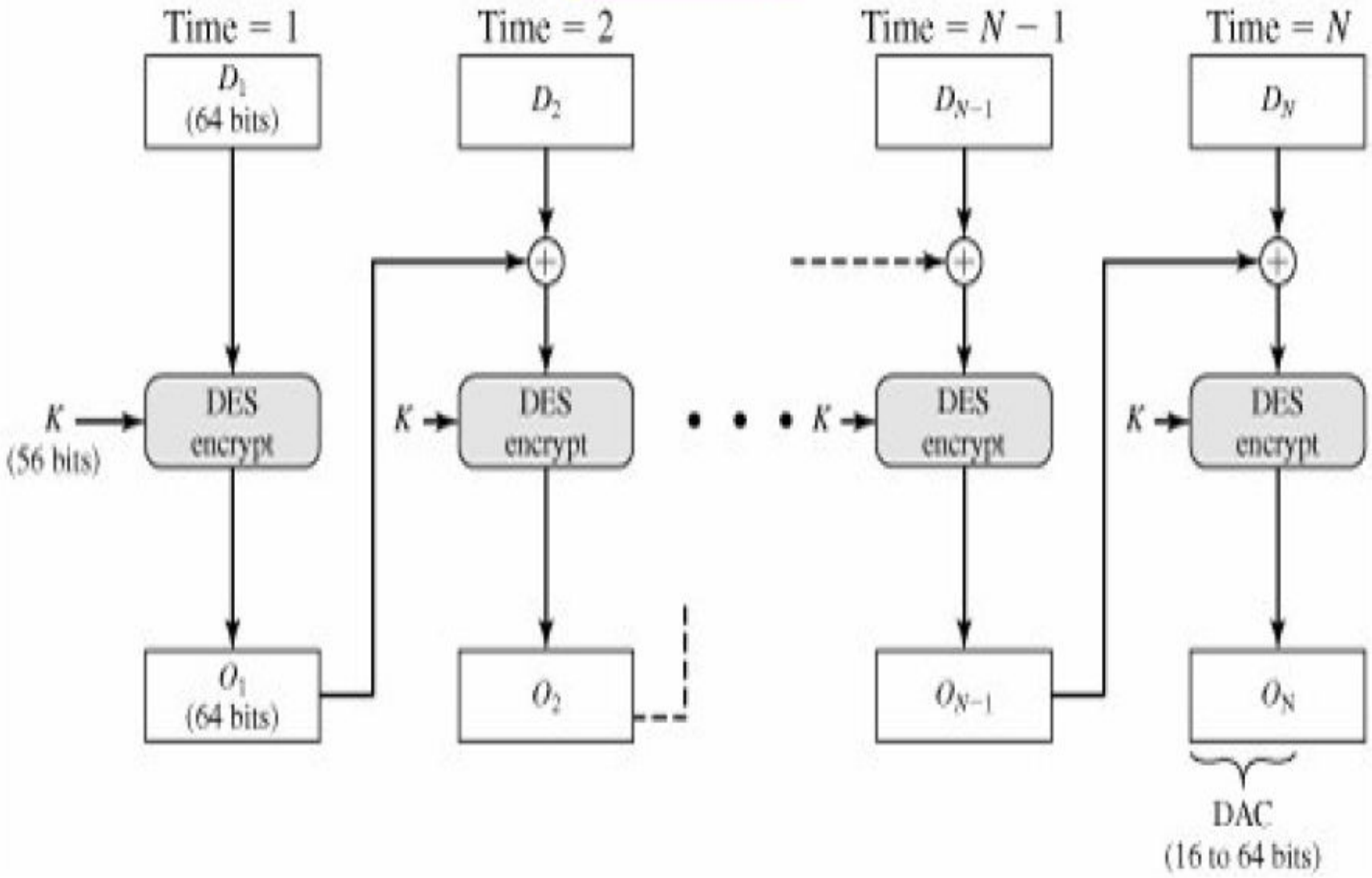
- taking into account the types of attacks
- need the MAC to satisfy the following:
  1. knowing a message and MAC, is infeasible to find another message with same MAC
  2. MACs should be uniformly distributed
  3. MAC should depend equally on all bits of the message

Using Symmetric Ciphers for MACs

- can use any block cipher chaining mode and use final block as a MAC
- **Data Authentication Algorithm (DAA)** is a widely used MAC based on DES-CBC
  - using IV=0 and zero-pad of final block
  - encrypt message using DES in CBC mode
  - and send just the final block as the MAC
    - or the leftmost M bits (16≤M≤64) of final block
- but final MAC is now too small for security

Data Authentication Algorithm (FIPS PUB 113)

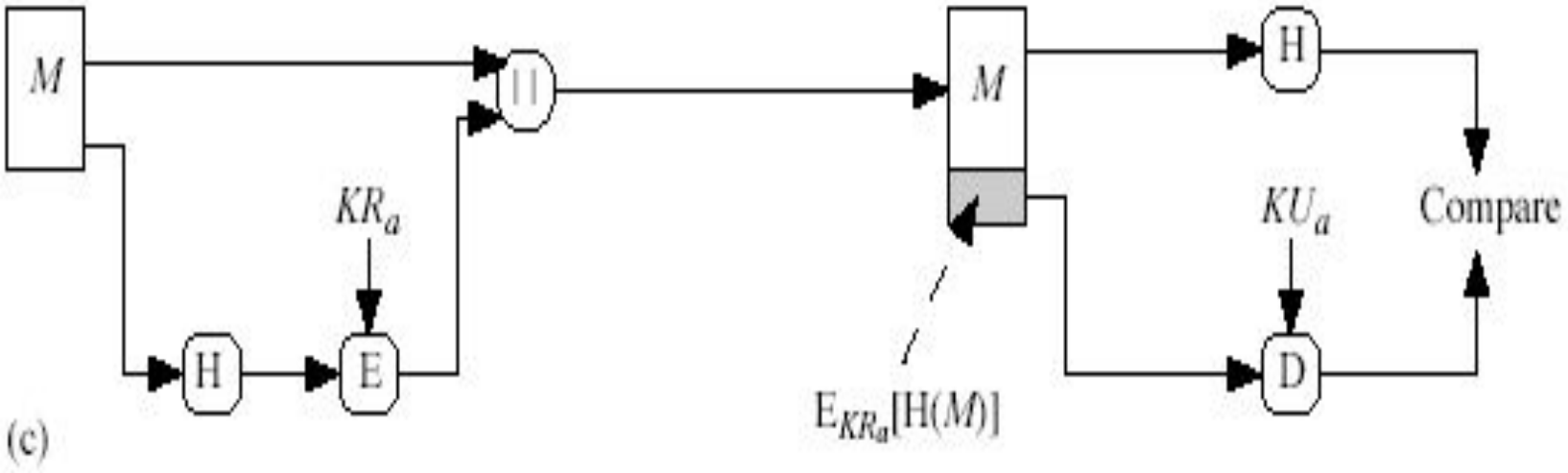
[View full size image]



Hash Functions

- condenses arbitrary message to fixed size
- usually assume that the hash function is public and not keyed
  - cf. MAC which is keyed
- used to detect changes to message
- can use in various ways with message
- most often to create a digital signature

Hash Functions & Digital Signatures



Hash Function Properties

- a Hash Function produces a fingerprint of some file/message/data

  `h = H(M)`

  – condenses a variable-length message M
  – to a fixed-sized fingerprint
- assumed to be public

Requirements for Hash Functions

1. can be applied to any sized message `M`
2. produces fixed-length output `h`
3. is easy to compute `h=H(M)` for any message `M`
4. given `h` is infeasible to find `x` s.t. `H(x)=h`
   - one-way property
5. given `x` is infeasible to find `y` s.t. `H(y)=H(x)`
   - weak collision resistance
6. is infeasible to find any `x,y` s.t. `H(y)=H(x)`
   - strong collision resistance

Simple Hash Functions

- are several proposals for simple functions
- based on XOR of message blocks
- not secure since can manipulate any message to produce a given hash
- need a stronger cryptographic function (next chapter)

	bit 1	bit 2	• • •	bit n
block 1	b_{11}	b_{21}		b_{n1}
block 2	b_{12}	b_{22}		b_{n2}
	•	•	•	•
	•	•	•	•
	•	•	•	•
block m	b_{1m}	b_{2m}		b_{nm}
hash code	C_1	C_2		C_n

Figure 11.7 Simple Hash Function Using Bitwise XOR

Birthday Attacks

- might think a 64-bit hash is secure
- but by **Birthday Paradox** is not
- **birthday attack** works thus:
  - opponent generates $2^{m/2}$ variations of a valid message all with essentially the same meaning
  - opponent also generates $2^{m/2}$ variations of a desired fraudulent message
  - two sets of messages are compared to find pair with same hash (probability > 0.5 by birthday paradox)
  - have user sign the valid message, then substitute the forgery which will have a valid signature
- conclusion is that need to use larger MACs

Dear Anthony,

{This letter is / I am writing} to introduce {you to / to you} {Mr. / --} Alfred {P. / --}

Barton, the {new / newly appointed} {chief / senior} jewellery buyer for {our / the}

Northern {European / Europe} {area / division} . He {will take / has taken} over {the / --}

responsibility for {all / the whole of} our interests in {watches and jewellery / jewellery and watches}

in the {area / region} . Please {afford / give} him {every / all the} help he {may need / needs}

to {seek out / find} the most {modern / up to date} lines for the {top / high} end of the

market. He is {empowered / authorized} to receive on our behalf {samples / specimens} of the

{latest / newest} {watch and jewellery / jewellery and watch} products, {up / subject} to a {limit / maximum}

of ten thousand dollars. He will {carry / hold} a signed copy of this {letter / document}

as proof of identity. An order with his signature, which is {appended / attached}

{authorizes / allows} you to charge the cost to this company at the {above / head office}

address. We {fully / --} expect that our {level / volume} of orders will increase in

the {following / next} year and {trust / hope} that the new appointment will {be / prove}

{advantageous / an advantage} to both our companies.

Figure 11.9 A Letter in 2^{3} ^{7} Variations [DAVI89]

Block Ciphers as Hash Functions

- can use block ciphers as hash functions
  - using H_0=0 and zero-pad of final block
  - compute: $H_i = E_{Mi} [H_{i-1}]$
  - and use final block as the hash value
  - similar to CBC but without a key
- resulting hash is too small (64-bit)
  - due to direct birthday attack and variants

Hash Functions & MAC Security

- like block ciphers have:
- **brute-force** attacks exploiting
  - strong collision resistance hash have cost $2^{m/2}$
    - 128-bit hash looks vulnerable, 160-bits better
  - MACs with known message-MAC pairs
    - can either attack keyspace (cf key search) or MAC
      - Min(2^k, 2^n)
    - at least 128-bit MAC and 128-bit key is needed for security

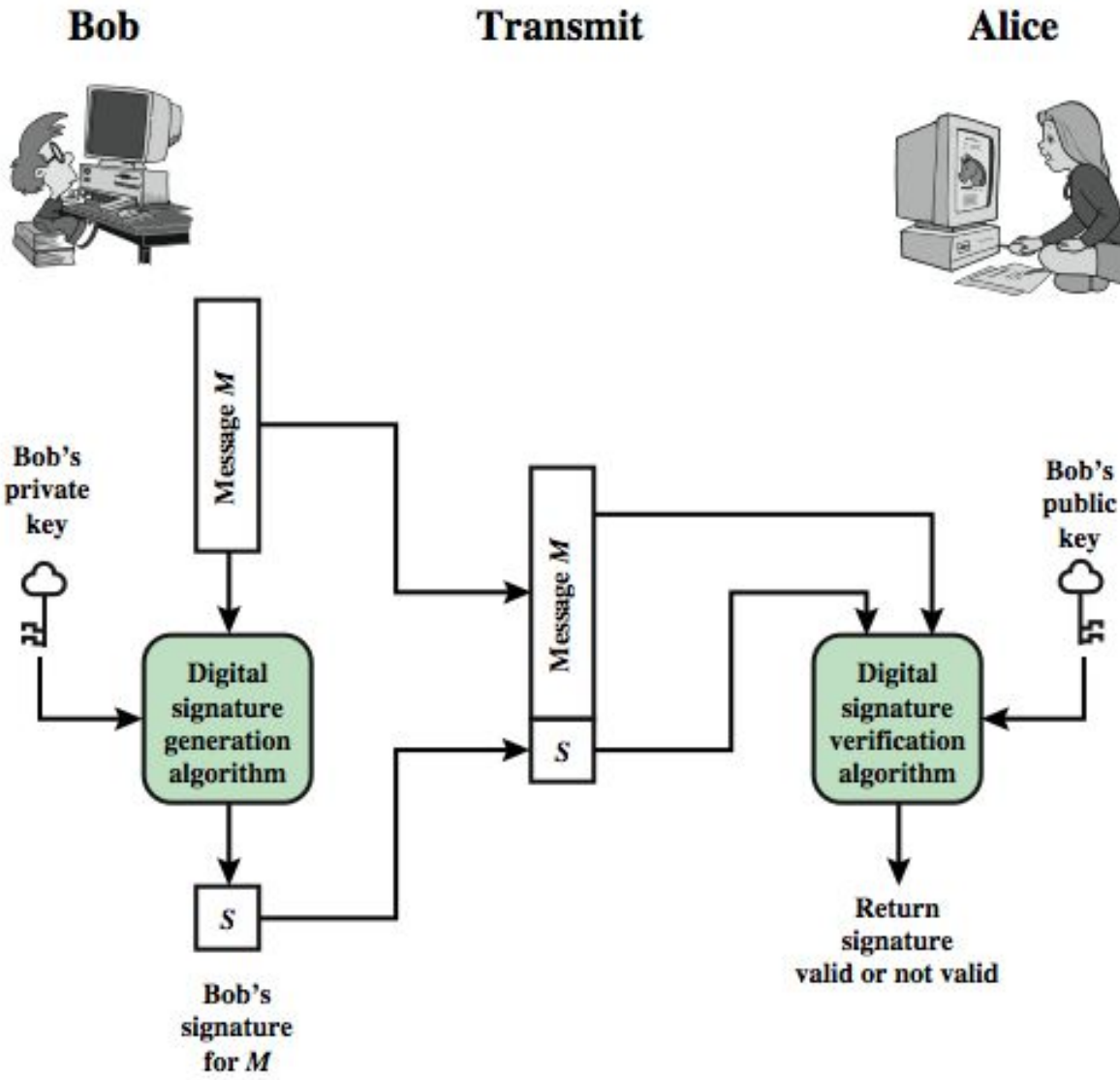
Hash Functions & MAC Security

- **cryptanalytic attacks** exploit structure
  - like block ciphers want brute-force attacks to be the best alternative
- have a number of analytic attacks on iterated hash functions
  - $CV_i = f[CV_{i-1}, M_i]$; $H(M)=CV_N$
  - typically focus on collisions in function f
  - like block ciphers is often composed of rounds
  - attacks exploit properties of round functions

Digital Signatures

- have looked at message authentication
  - but does not address issues of lack of trust
- digital signatures provide the ability to:
  - verify author, date & time of signature
  - authenticate message contents
  - be verified by third parties to resolve disputes
- hence include authentication function with additional capabilities

Digital Signature Model



Digital Signature Requirements

- must depend on the message signed
- must use information unique to sender
  - to prevent both forgery and denial
- must be relatively easy to produce
- must be relatively easy to recognize & verify
- be computationally infeasible to forge
  - with new message for existing digital signature
  - with fraudulent digital signature for given message
- be practical save digital signature in storage

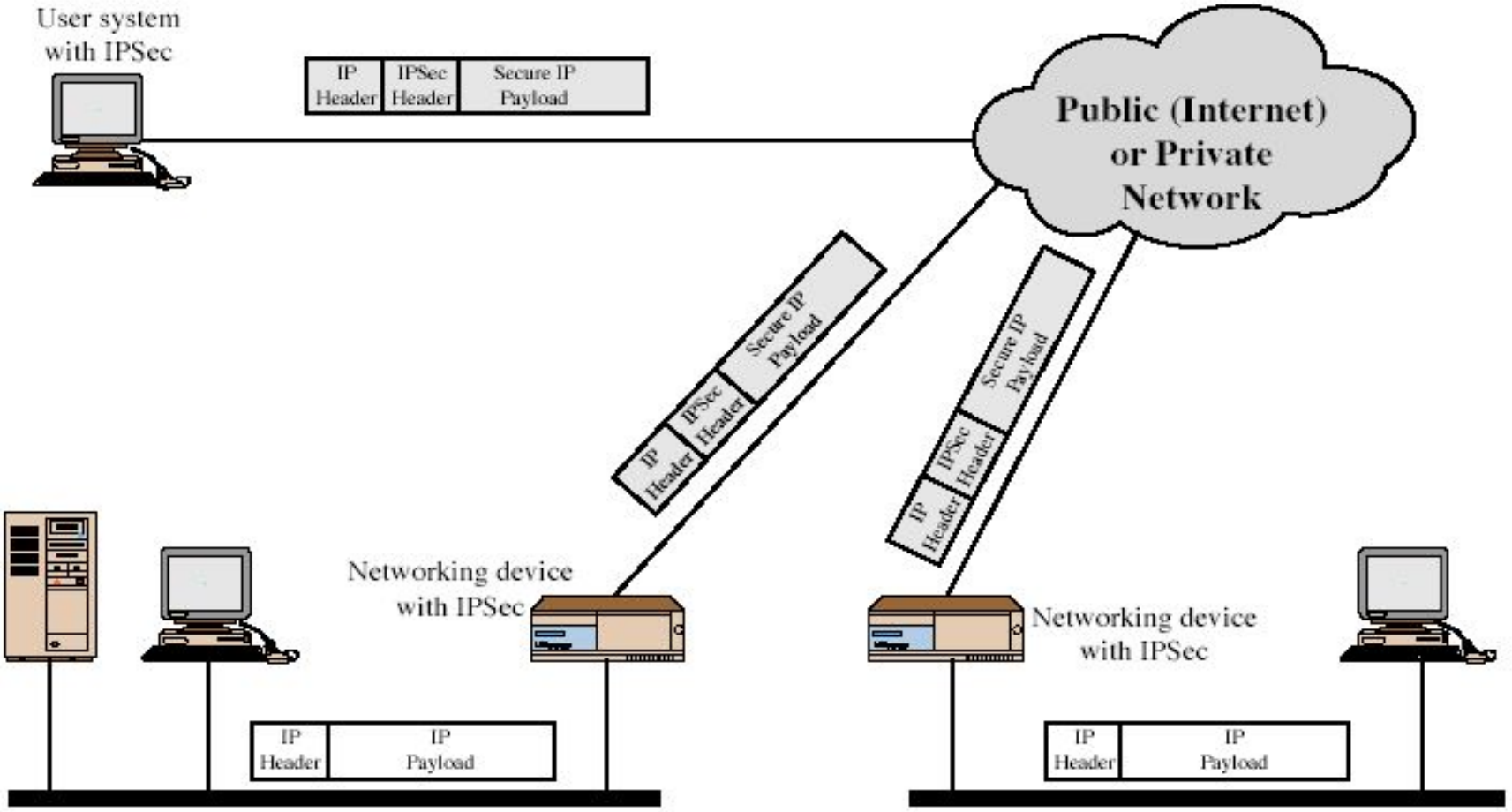
Direct Digital Signatures

- involve only sender & receiver
- assumed receiver has sender's public-key
- digital signature made by sender signing entire message or hash with private-key
- can encrypt using receivers public-key
- important that sign first then encrypt message & signature
- security depends on sender's private-key

IP Security

- have considered some application specific security mechanisms
  - eg. S/MIME, PGP, Kerberos, SSL/HTTPS
- however there are security concerns that cut across protocol layers
  - would like security implemented by the network for all applications

IPSec Uses


User system with IPSec
IP Header
IPSec Header
Secure IP Payload
Public (Internet) or Private Network
Secure IP Payload
IPSec Header
IP Header
Secure IP Payload
IPSec Header
IP Header
Networking device with IPSec
Networking device with IPSec
IP Header
IP Payload
IP Header
IP Payload

IPSec

- general IP Security mechanisms
- provides
  - authentication
  - confidentiality
  - key management
- applicable to use over LANs, across public & private WANs, & for the Internet

Benefits of IPSec

- in a firewall/router provides strong security to all traffic crossing the perimeter
- is resistant to bypass
- is below transport layer, hence transparent to applications
- can be transparent to end users

IP Security Architecture

- specification is quite complex
- defined in numerous RFC's
  - incl. RFC 2401/2402/2406/2408
  - many others, grouped by category
- mandatory in IPv6, optional in IPv4

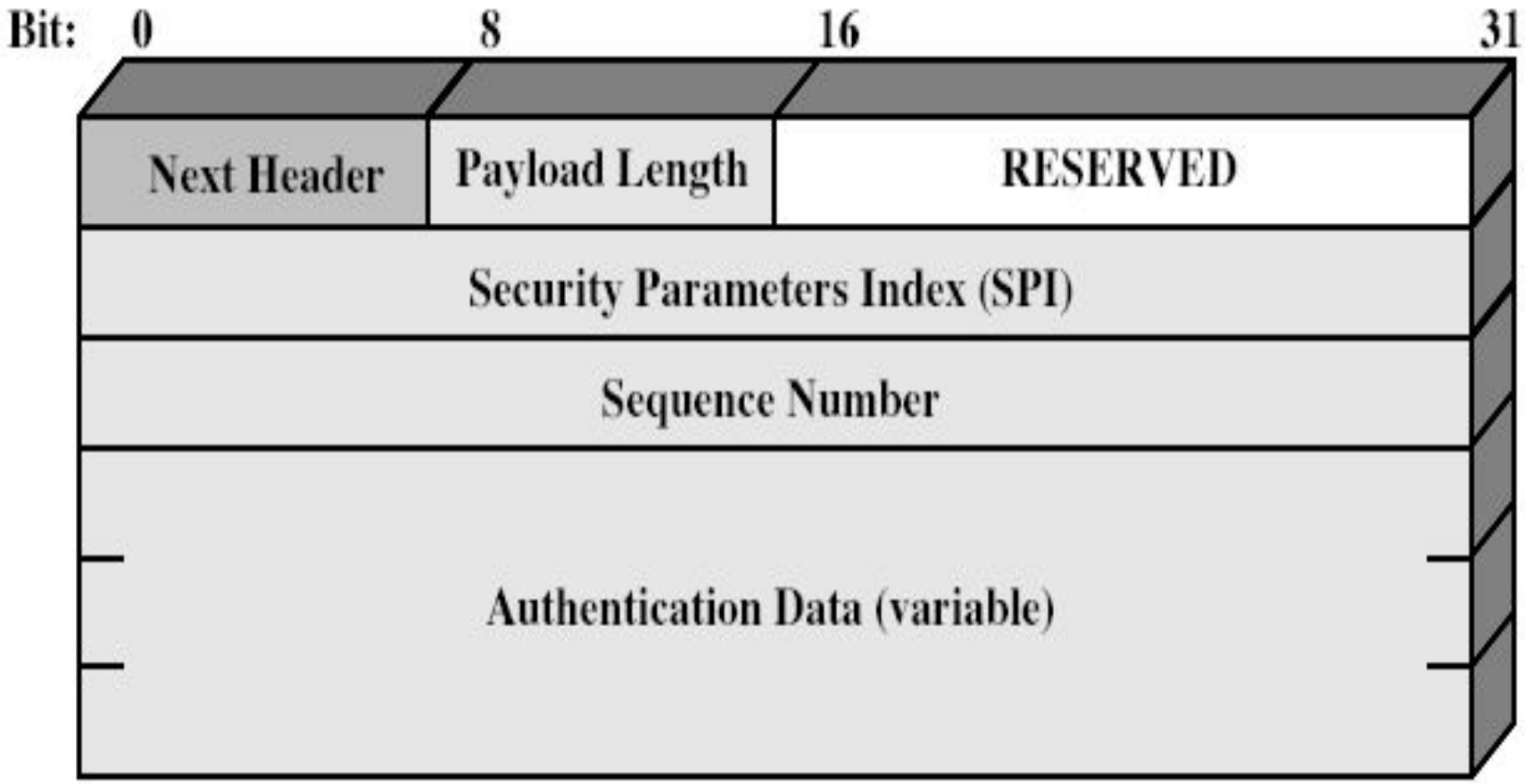
IPSec Protocols

- Authentication Header (AH)
  - Authentication
- Encapsulating Security Payload (ESP)
  - Confidentiality only
  - OR both

Security Associations

- a one-way relationship between sender & receiver that affords security for traffic flow
- defined by 3 parameters:
  - Security Parameters Index (SPI)
  - IP Destination Address
  - Security Protocol Identifier (AH or ESP?)
- has a number of other parameters
  - seq no, AH & EH info, lifetime etc
- have a database of Security Associations

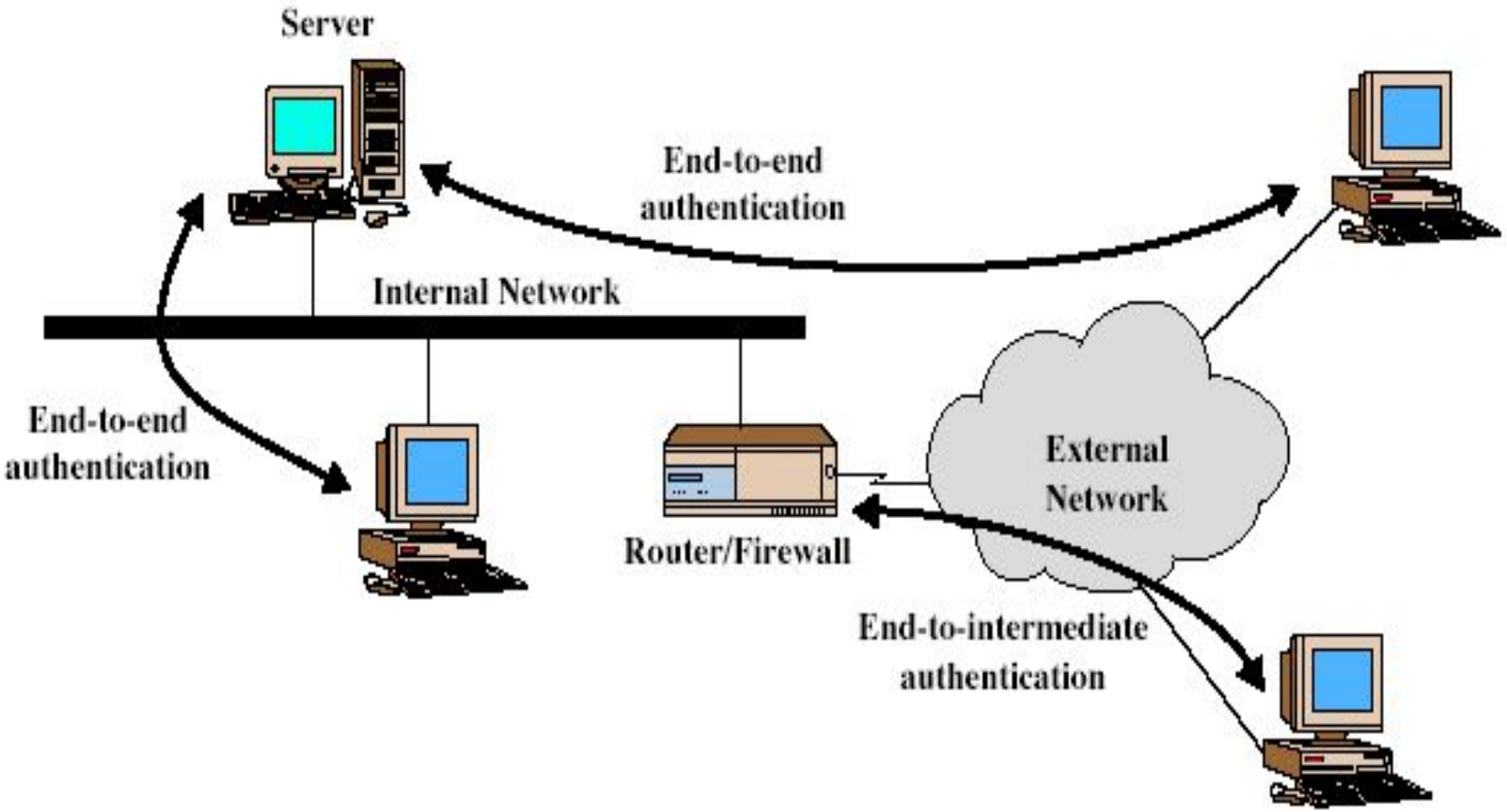
Authentication Header



Authentication Header (AH)

- provides support for data integrity & authentication of IP packets
  - end system/router can authenticate user/app
  - prevents replay attack by tracking sequence numbers
- based on use of a MAC
  - HMAC-MD5-96 or HMAC-SHA-1-96
- parties must share a secret key

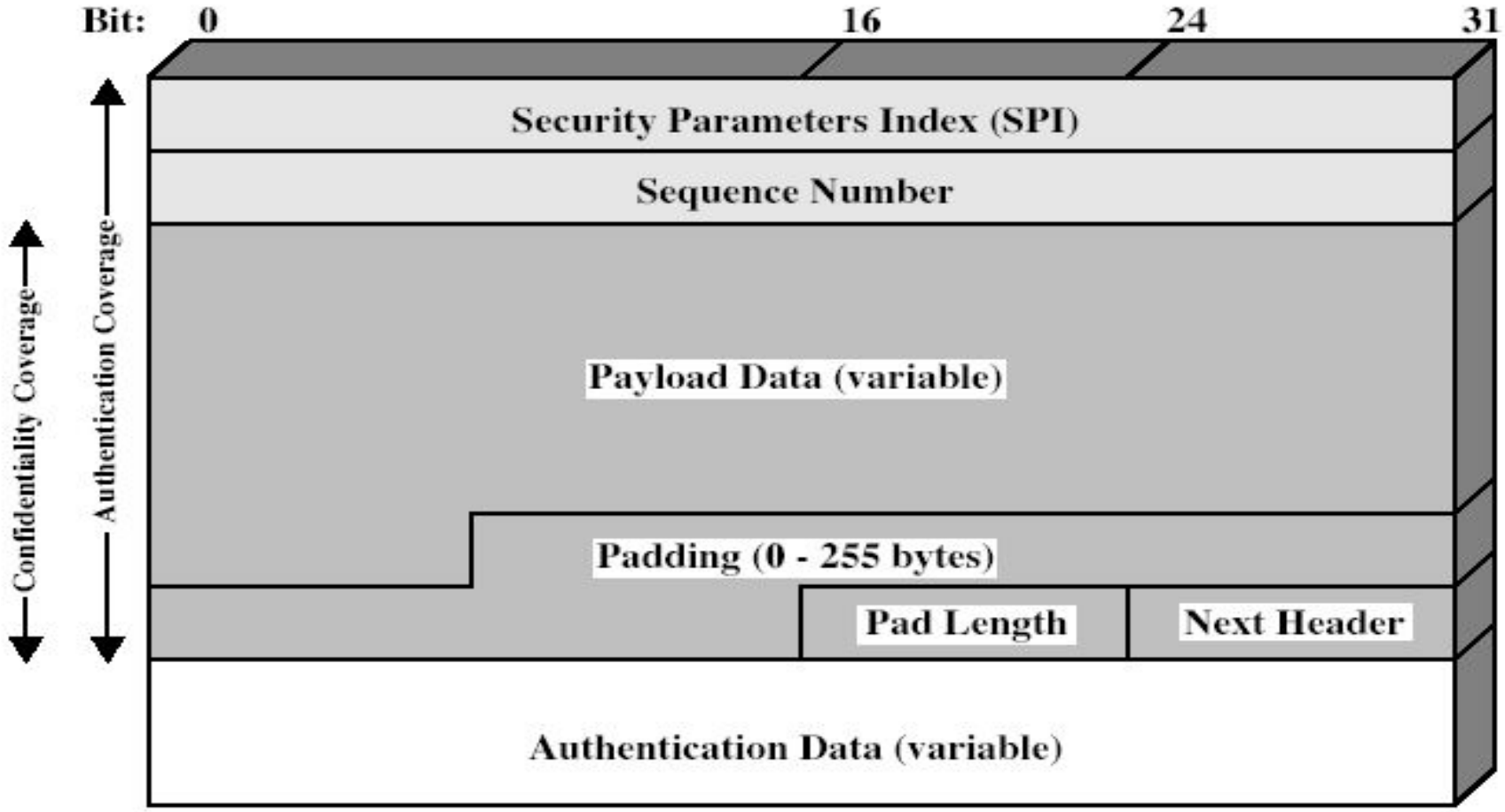
Transport & Tunnel Modes



Encapsulating Security Payload (ESP)

- provides message content confidentiality
- can optionally provide the same authentication services as AH
- supports range of ciphers, modes, padding
  - incl. DES, Triple-DES, RC5, IDEA, CAST etc
  - CBC most common

Encapsulating Security Payload



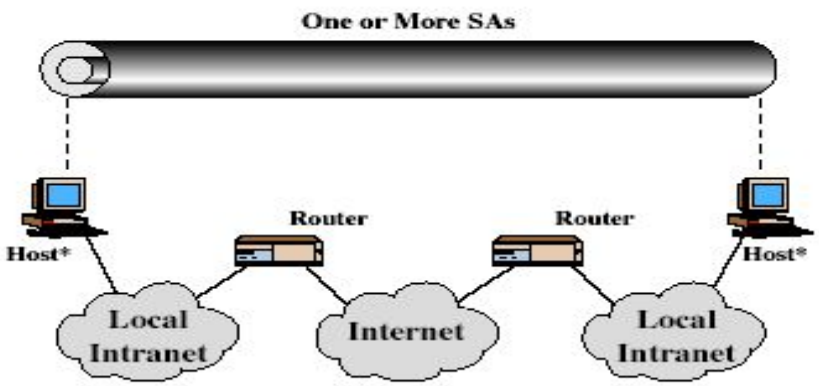
Transport vs Tunnel Mode ESP

- transport mode is used to encrypt & optionally authenticate IP data
  - data protected but header left in clear
  - can do traffic analysis but is efficient
  - good for ESP host to host traffic
- tunnel mode encrypts entire IP packet
  - add new header for next hop
  - good for VPNs, gateway to gateway security

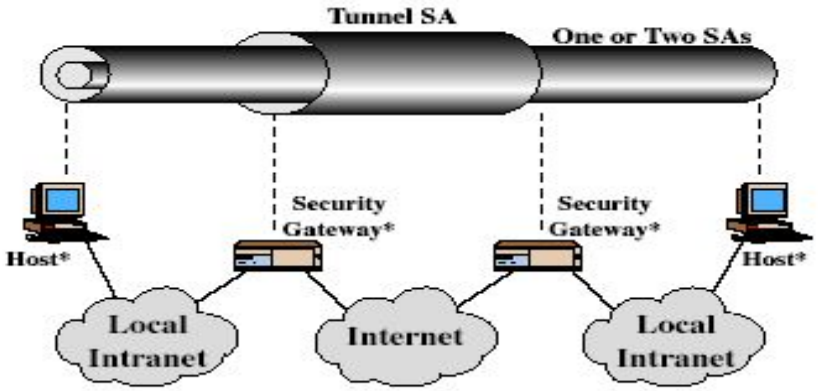
Combining Security Associations

- SA's can implement either AH or ESP
- to implement both need to combine SA's
  - form a security bundle

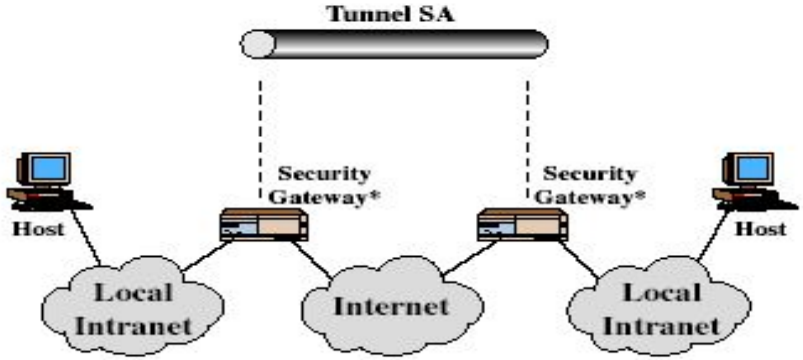
Combining Security Associations



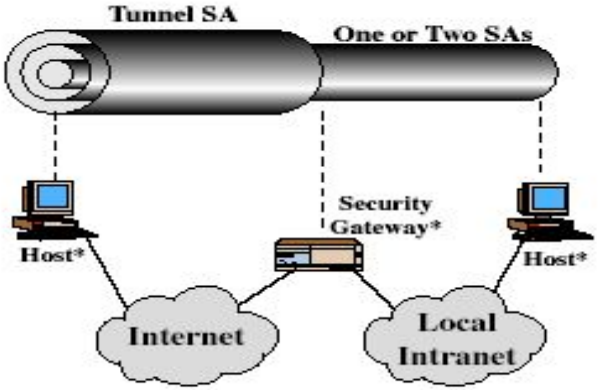
(a) Case 1



(c) Case 3



(b) Case 2



(d) Case 4

UNIT - IV

Intruders

- significant issue for networked systems is hostile or unwanted access
- either via network or local
- can identify classes of intruders:
  - masquerader
  - misfeasor
  - clandestine user
- varying levels of competence

Intruders

- clearly a growing publicized problem
  - from “Wily Hacker” in 1986/87
  - to clearly escalating CERT stats
- may seem benign, but still cost resources
- may use compromised system to launch other attacks

Intrusion Techniques

- aim to increase privileges on system
- basic attack methodology
  - target acquisition and information gathering
  - initial access
  - privilege escalation
  - covering tracks
- key goal often is to acquire passwords
- so then exercise access rights of owner

Password Guessing

- one of the most common attacks
- attacker knows a login (from email/web page etc)
- then attempts to guess password for it
  - try default passwords shipped with systems
  - try all short passwords
  - then try by searching dictionaries of common words
  - intelligent searches try passwords associated with the user (variations on names, birthday, phone, common words/interests)
  - before exhaustively searching all possible passwords
- check by login attempt or against stolen password file
- success depends on password chosen by user
- surveys show many users choose poorly

Password Capture

- another attack involves **password capture**
  - watching over shoulder as password is entered
  - using a trojan horse program to collect
  - monitoring an insecure network login (eg. telnet, FTP, web, email)
  - extracting recorded info after successful login (web history/cache, last number dialed etc)
- using valid login/password can impersonate user
- users need to be educated to use suitable precautions/countermeasures

Intrusion Detection

- inevitably will have security failures
- so need also to detect intrusions so can
  - block if detected quickly
  - act as deterrent
  - collect info to improve security
- assume intruder will behave differently to a legitimate user
  - but will have imperfect distinction between

Approaches to Intrusion Detection

- statistical anomaly detection
  - threshold
  - profile based
- rule-based detection
  - anomaly
  - penetration identification

Audit Records

- fundamental tool for intrusion detection
- native audit records
  - part of all common multi-user O/S
  - already present for use
  - may not have info wanted in desired form
- detection-specific audit records
  - created specifically to collect wanted info
  - at cost of additional overhead on system

Statistical Anomaly Detection

- threshold detection
  - count occurrences of specific event over time
  - if exceed reasonable value assume intrusion
  - alone is a crude & ineffective detector
- profile based
  - characterize past behavior of users
  - detect significant deviations from this
  - profile usually multi-parameter

Audit Record Analysis

- foundation of statistical approaches
- analyze records to get metrics over time
  - counter, gauge, interval timer, resource use
- use various tests on these to determine if current behavior is acceptable
  - mean & standard deviation, multivariate, markov process, time series, operational
- key advantage is no prior knowledge used

Rule-Based Intrusion Detection

- observe events on system & apply rules to decide if activity is suspicious or not
- rule-based anomaly detection
  - analyze historical audit records to identify usage patterns & auto-generate rules for them
  - then observe current behavior & match against rules to see if conforms
  - like statistical anomaly detection does not require prior knowledge of security flaws

Rule-Based Intrusion Detection

- rule-based penetration identification
  - uses expert systems technology
  - with rules identifying known penetration, weakness patterns, or suspicious behavior
  - rules usually machine & O/S specific
  - rules are generated by experts who interview & codify knowledge of security admins
  - quality depends on how well this is done
  - compare audit records or states against rules

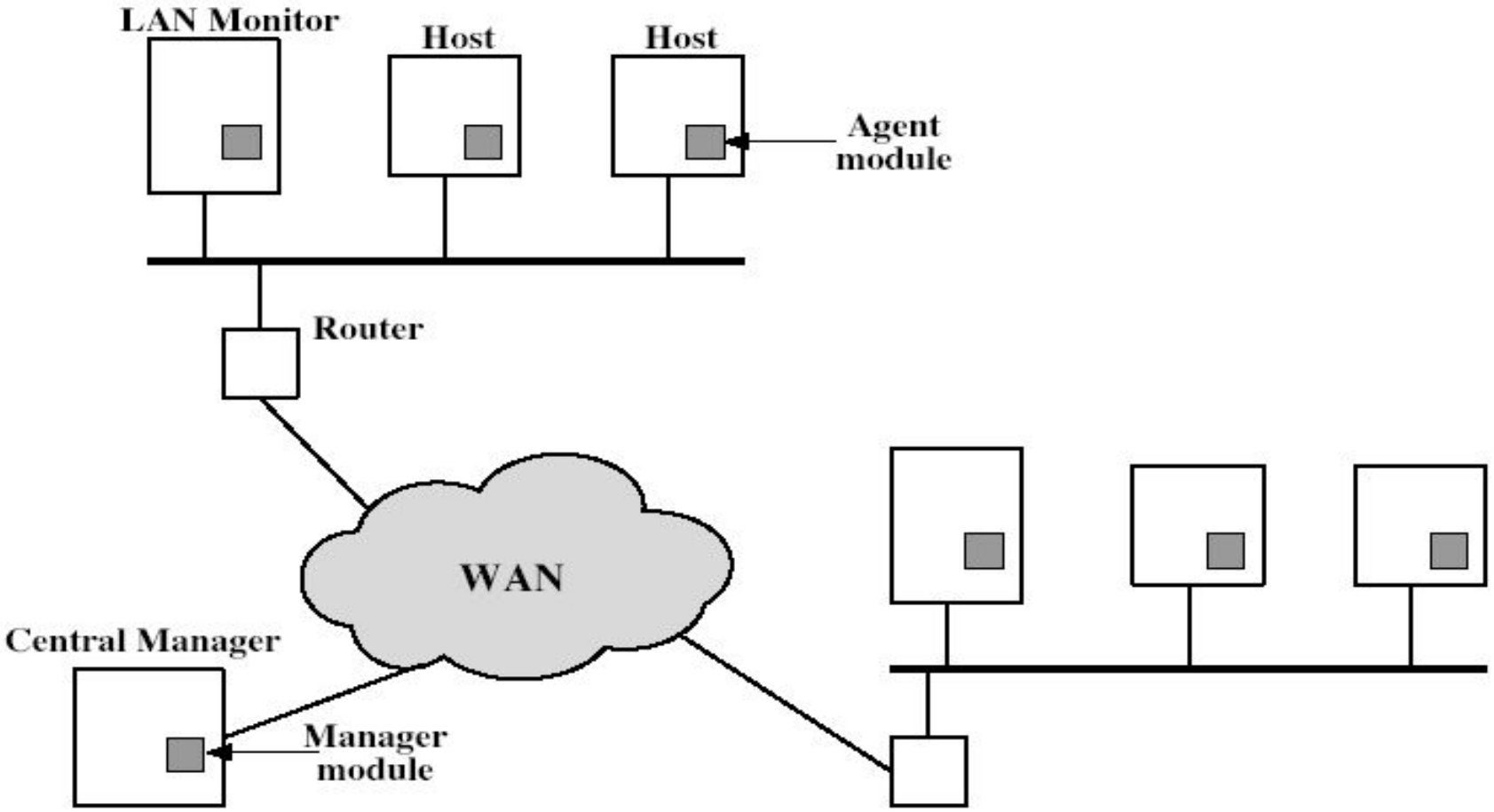
Base-Rate Fallacy

- practically an intrusion detection system needs to detect a substantial percentage of intrusions with few false alarms
  - if too few intrusions detected -> false security
  - if too many false alarms -> ignore / waste time
- this is very hard to do
- existing systems seem not to have a good record

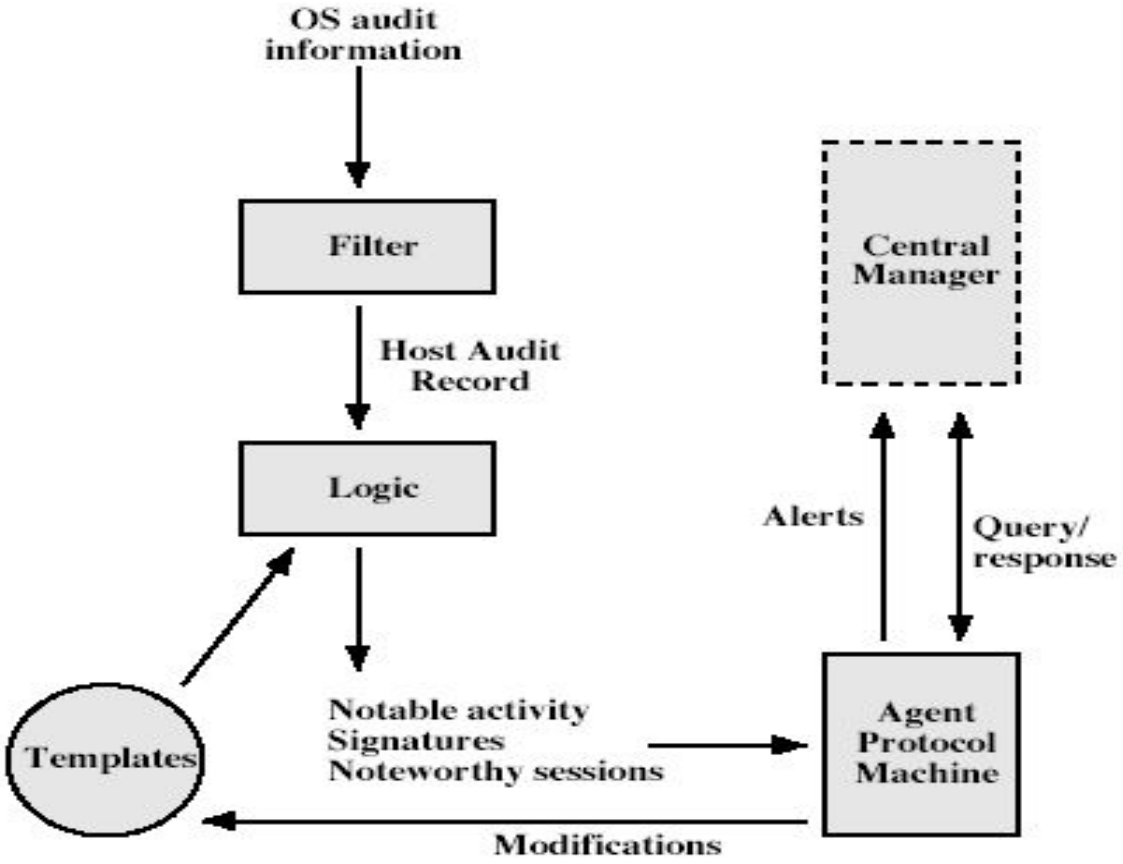
Distributed Intrusion Detection

- traditional focus is on single systems
- but typically have networked systems
- more effective defense has these working together to detect intrusions
- issues
  - dealing with varying audit record formats
  - integrity & confidentiality of networked data
  - centralized or decentralized architecture

Distributed Intrusion Detection - Architecture



Distributed Intrusion Detection – Agent Implementation



Honeypots

- decoy systems to lure attackers
  - away from accessing critical systems
  - to collect information of their activities
  - to encourage attacker to stay on system so administrator can respond
- are filled with fabricated information
- instrumented to collect detailed information on attackers activities
- may be single or multiple networked systems

Password Management

- front-line defense against intruders
- users supply both:
  - login – determines privileges of that user
  - password – to identify them
- passwords often stored encrypted
  - Unix uses multiple DES (variant with salt)
  - more recent systems use crypto hash function

Managing Passwords

- need policies and good user education
- ensure **every** account has a default password
- ensure users change the default passwords to something they can remember
- protect password file from general access
- set technical policies to enforce good passwords
  - minimum length (>6)
  - require a mix of upper & lower case letters, numbers, punctuation
  - block know dictionary words

Managing Passwords

- may reactively run password guessing tools
  - note that good dictionaries exist for almost any language/interest group
- may enforce periodic changing of passwords
- have system monitor failed login attempts, & lockout account if see too many in a short period
- do need to educate users and get support
- balance requirements with user acceptance
- be aware of **social engineering** attacks

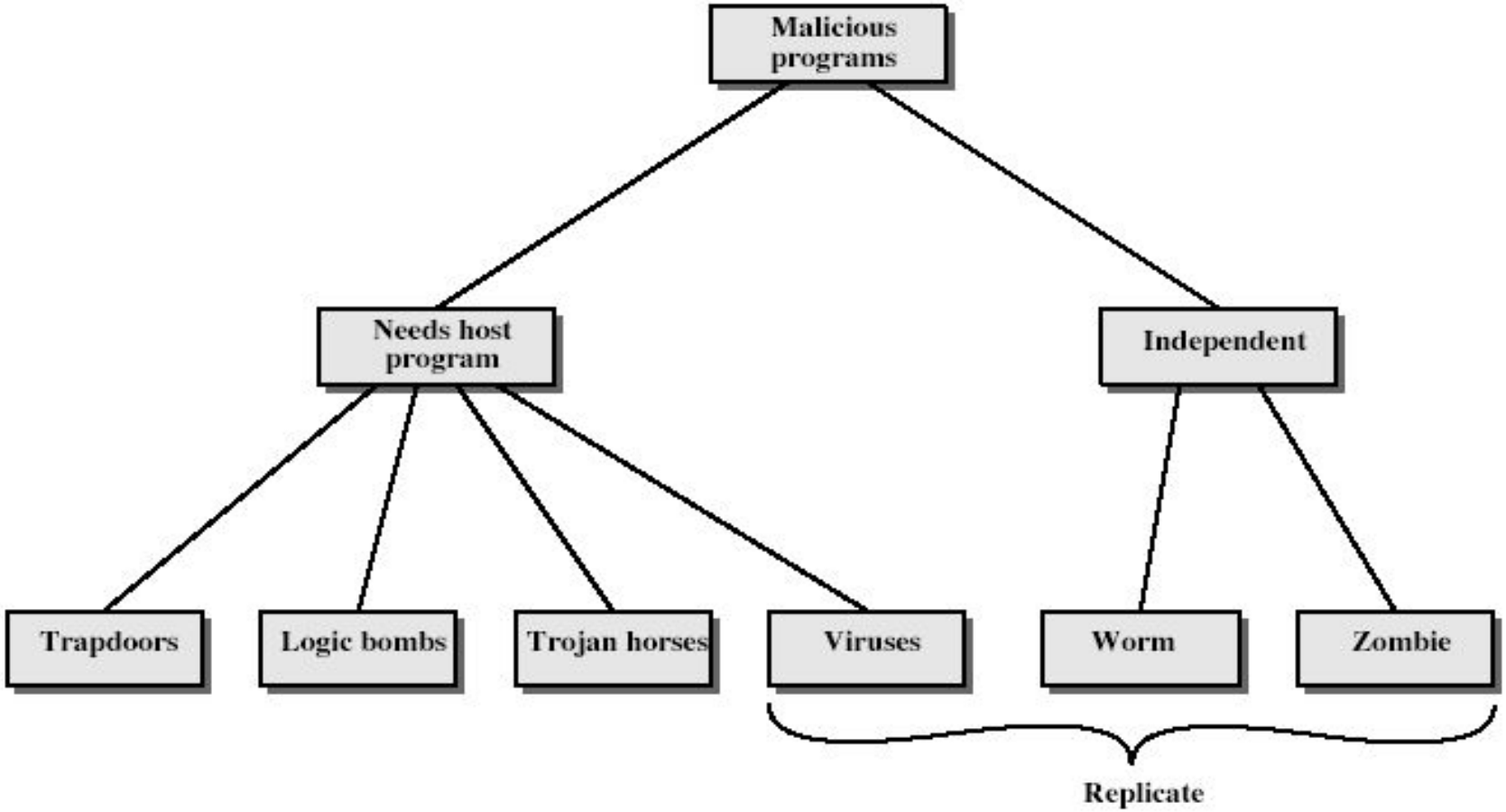
Proactive Password Checking

- most promising approach to improving password security
- allow users to select own password
- but have system verify it is acceptable
  - simple rule enforcement (see previous slide)
  - compare against dictionary of bad passwords
  - use algorithmic (markov model or bloom filter) to detect poor choices

Viruses and Other Malicious Content

- computer viruses have got a lot of publicity
- one of a family of **malicious software**
- effects usually obvious
- have figured in news reports, fiction, movies (often exaggerated)
- getting more attention than deserve
- are a concern though

Malicious Software



Trapdoors

- secret entry point into a program
- allows those who know access bypassing usual security procedures
- have been commonly used by developers
- a threat when left in production programs allowing exploited by attackers
- very hard to block in O/S
- requires good s/w development & update

Logic Bomb

- one of oldest types of malicious software
- code embedded in legitimate program
- activated when specified conditions met
  - eg presence/absence of some file
  - particular date/time
  - particular user
- when triggered typically damage system
  - modify/delete files/disks

Trojan Horse

- program with hidden side-effects
- which is usually superficially attractive
  - eg game, s/w upgrade etc
- when run performs some additional tasks
  - allows attacker to indirectly gain access they do not have directly
- often used to propagate a virus/worm or install a backdoor
- or simply to destroy data

Zombie

- program which secretly takes over another networked computer
- then uses it to indirectly launch attacks
- often used to launch distributed denial of service (DDoS) attacks
- exploits known flaws in network systems

Viruses

- a piece of self-replicating code attached to some other code
  - cf biological virus
- both propagates itself & carries a payload
  - carries code to make copies of itself
  - as well as code to perform some covert task

Virus Operation

- virus phases:
  - dormant – waiting on trigger event
  - propagation – replicating to programs/disks
  - triggering – by event to execute payload
  - execution – of payload
- details usually machine/OS specific
  - exploiting features/weaknesses

Virus Structure

```
program V :=
    {goto main;
    1234567;
    subroutine infect-executable :=  {loop:
                file := get-random-executable-file;
                if (first-line-of-file = 1234567) then goto loop
                else prepend V to file; }
    subroutine do-damage :=     {whatever damage is to be done}
    subroutine trigger-pulled :=  {return true if some condition holds}
    main: main-program :=  {infect-executable;
                            if trigger-pulled then do-damage;
                            goto next;}
    next:
}
```

Types of Viruses

- can classify on basis of how they attack
- parasitic virus
- memory-resident virus
- boot sector virus
- stealth
- polymorphic virus
- macro virus

Macro Virus

- **macro code** attached to some **data file**
- interpreted by program using file
  - eg Word/Excel macros
  - esp. using auto command & command macros
- code is now platform independent
- is a major source of new viral infections
- blurs distinction between data and program files making task of detection much harder
- classic trade-off: "ease of use" vs "security"

Email Virus

- spread using email with attachment containing a macro virus
  - cf Melissa
- triggered when user opens attachment
- or worse even when mail viewed by using scripting features in mail agent
- usually targeted at Microsoft Outlook mail agent & Word/Excel documents

Worms

- replicating but not infecting program
- typically spreads over a network
  - cf Morris Internet Worm in 1988
  - led to creation of CERTs
- using users distributed privileges or by exploiting system vulnerabilities
- widely used by hackers to create **zombie PC's**, subsequently used for further attacks, esp DoS
- major issue is lack of security of permanently connected systems, esp PC's

Worm Operation

- worm phases like those of viruses:
  - dormant
  - propagation
    - search for other systems to infect
    - establish connection to target remote system
    - replicate self onto remote system
  - triggering
  - execution

Morris Worm

- best known classic worm
- released by Robert Morris in 1988
- targeted Unix systems
- using several propagation techniques
  - simple password cracking of local pw file
  - exploit bug in finger daemon
  - exploit debug trapdoor in sendmail daemon
- if any attack succeeds then replicated self

Recent Worm Attacks

- new spate of attacks from mid-2001
- **Code Red**
  - exploited bug in MS IIS to penetrate & spread
  - probes random IPs for systems running IIS
  - had trigger time for denial-of-service attack
  - 2nd wave infected 360000 servers in 14 hours
- **Code Red 2**
  - had backdoor installed to allow remote control
- **Nimda**
  - used multiple infection mechanisms
    - email, shares, web client, IIS, Code Red 2 backdoor

Virus Countermeasures

- viral attacks exploit lack of integrity control on systems
- to defend need to add such controls
- typically by one or more of:
  - **prevention** - block virus infection mechanism
  - **detection** - of viruses in infected system
  - **reaction** - restoring system to clean state

Anti-Virus Software

- **first-generation**
  - scanner uses virus signature to identify virus
  - or change in length of programs
- **second-generation**
  - uses heuristic rules to spot viral infection
  - or uses program checksums to spot changes
- **third-generation**
  - memory-resident programs identify virus by actions
- **fourth-generation**
  - packages with a variety of antivirus techniques
  - eg scanning & activity traps, access-controls

Advanced Anti-Virus Techniques

- generic decryption
  - use CPU simulator to check program signature & behavior before actually running it
- digital immune system (IBM)
  - general purpose emulation & virus detection
  - any virus entering org is captured, analyzed, detection/shielding created for it, removed

Behavior-Blocking Software

- integrated with host O/S
- monitors program behavior in real-time
  - eg file access, disk format, executable mods, system settings changes, network access
- for possibly malicious actions
  - if detected can block, terminate, or seek ok
- has advantage over scanners
- but malicious code runs before detection

UNIT – V
CYBER SECURITY

Cyber Security

What is cybercrime?

Cybercrime is criminal activity that either targets or uses a computer, a computer network or a networked device.

- Most, but not all, cybercrime is committed by cybercriminals or hackers who want to make money. Cybercrime is carried out by individuals or organizations.

- Some cybercriminals are organized, use advanced techniques and are highly technically skilled. Others are novice hackers.

- Rarely, cybercrime aims to damage computers for reasons other than profit. These could be political or personal.

Various cybercrime

Here are some specific examples of the different types of cybercrime:

- Email and internet fraud.
- Identity fraud (where personal information is stolen and used).
- Theft of financial or card payment data.
- Theft and sale of corporate data.
- Cyber extortion (demanding money to prevent a threatened attack).
- Ransom ware attacks (a type of cyberextortion).
- Crypto jacking (where hackers mine cryptocurrency using resources they do not own).
- Cyber espionage (where hackers access government or company data).

Most cybercrime falls under two main categories:

- Criminal activity that targets
- Criminal activity that uses computers to commit other crimes.

Cybercrime that targets computers often involves viruses and other types of malware.

Cybercriminals may infect computers with viruses and malware to damage devices or stop them working. They may also use malware to delete or steal data. Cybercrime that stops users using a machine or network, or prevents a business providing a software service to its customers, is called a Denial-of-Service (DoS) attack.

Cybercrime that uses computers to commit other crimes may involve using computers or networks to spread malware, illegal information or illegal images.

Sometimes cybercriminals conduct both categories of cybercrime at once. They may target computers with viruses first. Then, use them to spread malware to other machines or throughout a network.

The US has signed the ***European Convention of Cybercrime***. The convention casts a wide net and there are numerous malicious computer-related crimes which it considers cybercrime. For example:

- Illegally intercepting or stealing data.
- Interfering with systems in a way that compromises a network.
- Infringing copyright.
- Illegal gambling.
- Selling illegal items online
- Soliciting, producing or possessing child pornography

National Cyber Security Policy is a **policy** framework by Department of Electronics and Information Technology (DeitY) It aims at protecting the public and private infrastructure from **cyber** attacks.

I. Vision

To build a secure and resilient cyberspace for citizens, businesses and Government

II. Mission

- ✓ To protect information and information infrastructure in cyberspace, build capabilities to
- ✓ prevent and respond to cyber threats, reduce vulnerabilities and minimize damage from
- ✓ cyber incidents through a combination of institutional structures, people, processes,

technology and cooperation.

III. Objectives

1) To create a secure cyber ecosystem in the country, generate adequate trust & confidence in IT systems and transactions in cyberspace and thereby enhance adoption of IT in all sectors of the economy.

2) To create an assurance framework for design of security policies and for promotion andenabling actions for compliance to global security standards and best practices by way of conformity assessment (product, process, technology &

3) To strengthen the Regulatory framework for ensuring a Secure Cyberspace ecosystem.

4) To enhance and create National and Sectoral level 24 x 7 mechanisms for obtaining strategic information regarding threats to ICT infrastructure, creating scenarios for response, resolution and crisis management through effective predictive, preventive, protective, response and recovery actions.

5) To enhance the protection and resilience of Nation's critical information infrastructure by operating a 24x7 National Critical Information Infrastructure Protection Centre (NCIIPC) and mandating security practices related to the design, acquisition, development, use and operation of information resources.

6) To develop suitable indigenous security technologies through frontier technology research, solution oriented research, proof of concept, pilot development, transition, diffusion and commercialisation leading to widespread deployment of secure ICT products / processes in general and specifically for addressing National Security requirements.

7) To improve visibility of the integrity of ICT products and services by establishing infrastructure for testing & validation of security of such products.

8) To create a workforce of 500,000 professionals skilled in cyber security in the next 5 years through capacity building, skill development and training.

9) To provide fiscal benefits to businesses for adoption of standard security practices and processes.

10) To enable protection of information while in process, handling, storage & transit so as to safeguard privacy of citizen's data and for reducing economic losses due to cyber crime or data theft.

11) To enable effective prevention, investigation and prosecution of cyber crime and enhancement of law enforcement capabilities through appropriate legislative intervention.

12) To create a culture of cyber security and privacy enabling responsible user behaviour & actions through an effective communication and promotion strategy.

13) To develop effective public private partnerships and collaborative engagements through technical and operational cooperation and contribution for enhancing the security of cyberspace. 1

14) To enhance global cooperation by promoting shared understanding and leveraging relationships for furthering the cause of security of cyberspace.

Indian Cyberspace

Cyberspace is a concept describing a widespread, interconnected digital technology. "The expression dates back from the first decade of the diffusion of the internet. It refers to the online world as a world "apart," as distinct from everyday reality. In cyberspace people can hide behind fake identities, as in the famous The New Yorker cartoon."

Recent definitions of Cyberspace

Although several definitions of cyberspace can be found both in scientific literature and in official governmental sources, there is no fully agreed official definition yet. According to F. D. Kramer there are 28 different definitions of the term cyberspace

Cyberspace is a global and dynamic domain (subject to constant change) characterized by the combined use of electrons and the electromagnetic spectrum, whose purpose is to create, store, modify, exchange, share, and extract, use, eliminate information and disrupt physical resources.

Cyberspace includes:

a) physical infrastructures and telecommunications devices that allow for the connection of technological and communication system networks, understood in the broadest sense (SCADA devices, smartphones/tablets, computers, servers, etc.);
b) computer systems (see point a) and the related (sometimes embedded) software that guarantee the domain's basic operational functioning and connectivity;
c) networks between computer systems;
d) networks of networks that connect computer systems (the distinction between networks and networks of networks is mainly organizational);
e) the access nodes of users and intermediaries routing nodes;
f) constituent data (or resident data). Often, in common parlance (and sometimes in commercial language), networks of networks are called the Internet (with a lowercase i), while networks between computers are called intranet. Internet (with a capital I, in journalistic language sometimes called the Net) can be considered a part of the system a). A distinctive and constitutive feature of cyberspace is that no central entity exercises control over all the networks that make up this new domain.

IT ACT 2008 -History

1999 - Information Technology Bill was prepaaraed

May 2000 – This bill was passed by both the houses of parliament.

August 2000 – This was passed by President ofIndia and was came to be known as ”***Information Technology Act – 2000***”.

2006 – The act was amended and presented to parliament.

December 2008 – The act was passed bynthe parliament & renamed to “***Information Technology(Amendment) Act-2008***’.

ITAA 2008 COMPOSITION

- 4 SCHEDULES
- 13 CHAPTERS
- 90 SECTIONS
- SUB-SECTIONS

SECTION 1. **Short Title, Extent, Commencement and Application**

(1) This Act may be called the Information Technology Act, 2000. [As Amended by Information technology (Amendment) Act 2008] P.S: Information Technology (Amendment) Bill 2006 was amended by Information Technology Act Amendment Bill 2008 and in the process, the underlying Act was renamed as Information Technology (Amendment) Act 2008 herein after referred to as ITAA 2008.

(2) It shall extend to the whole of India and, save as otherwise provided in this Act, it applies also to any offence or contravention hereunder committed outside India by any person.

(3) It shall come into force on such date as the Central Government may, by notification, appoint and different dates may be appointed for different provisions of this Act and any reference in any such provision to the commencement of this Act shall be construed as a reference to the commencement of that provision.[Act notified with effect from October 17, 2000. Amendments vide ITAA-2008 notified with effect from....]

(4) (Substituted Vide ITAA-2008) Nothing in this Act shall apply to documents or transactions specified in the First Schedule by way of addition or deletion of entries thereto

(5) (Inserted vide ITAA-2008) Every notification issued under sub-section (4) shall be laid before each House of Parliament

2 Definitions

(1) In this Act, unless the context otherwise requires,
(2) (a) "Access" with its grammatical variations and cognate expressions means gaining entry into, instructing or communicating with the logical, arithmetical, or memory function resources of a computer, computer system or computer network;
(3) (b) "Addressee" means a person who is intended by the originator to receive the electronic record but does not include any intermediary;
(4) (c) "Adjudicating Officer" means adjudicating officer appointed under subsection (
(5) 1) of section 46;
(6) (d) "Affixing Electronic Signature" with its grammatical variations and cognate expressions means adoption of any methodology or procedure by a person for the purpose of authenticating an electronic record by means of Electronic Signature;

(e) "Appropriate Government" means as respects any matter.

(i) enumerated in List II of the Seventh Schedule to the Constitution;

(ii) relating to any State law enacted under List III of the Seventh Schedule to the Constitution, the State Government and in any other case, the Central Government;

(f) "Asymmetric Crypto System" means a system of a secure key pair consisting of a private key for creating a digital signature and a public key to verify the digital signature;

(g) "Certifying Authority" means a person who has been granted a license to issue a Electronic Signature Certificate under section 24;

(h) "Certification Practice Statement" means a statement issued by a Certifying Authority to specify the practices that the Certifying Authority employs in issuing Electronic Signature Certificates; (ha) "Communication Device" means Cell Phones, Personal Digital Assistance (Sic), or combination of both or any other device used to communicate, send or transmit any text, video, audio, or image. (Inserted Vide ITAA 2008)

(i) "Computer" means any electronic, magnetic, optical or other high-speed data processing device or system which performs logical, arithmetic, and memory functions by manipulations of electronic, magnetic or optical impulses, and includes all input, output, processing, storage, computer software, or communication facilities which are connected or related to the computer in a computer system or computer network;

(ii) (j) (Substituted vide ITAA-2008) "Computer Network" means the interconnection of one or more Computers or Computer systems or Communication device through-

(i) (i) the use of satellite, microwave, terrestrial line,wire,wireless or other communication media; and (ii) terminals or a complex consisting of two or more interconnected computers or communication device whether or not the interconnection is continuously maintained; (k) "Computer Resource" means computer, communication device, computer system, computer network, data, computer database or software;

THANK YOU