Network Security

Asim Rasheed

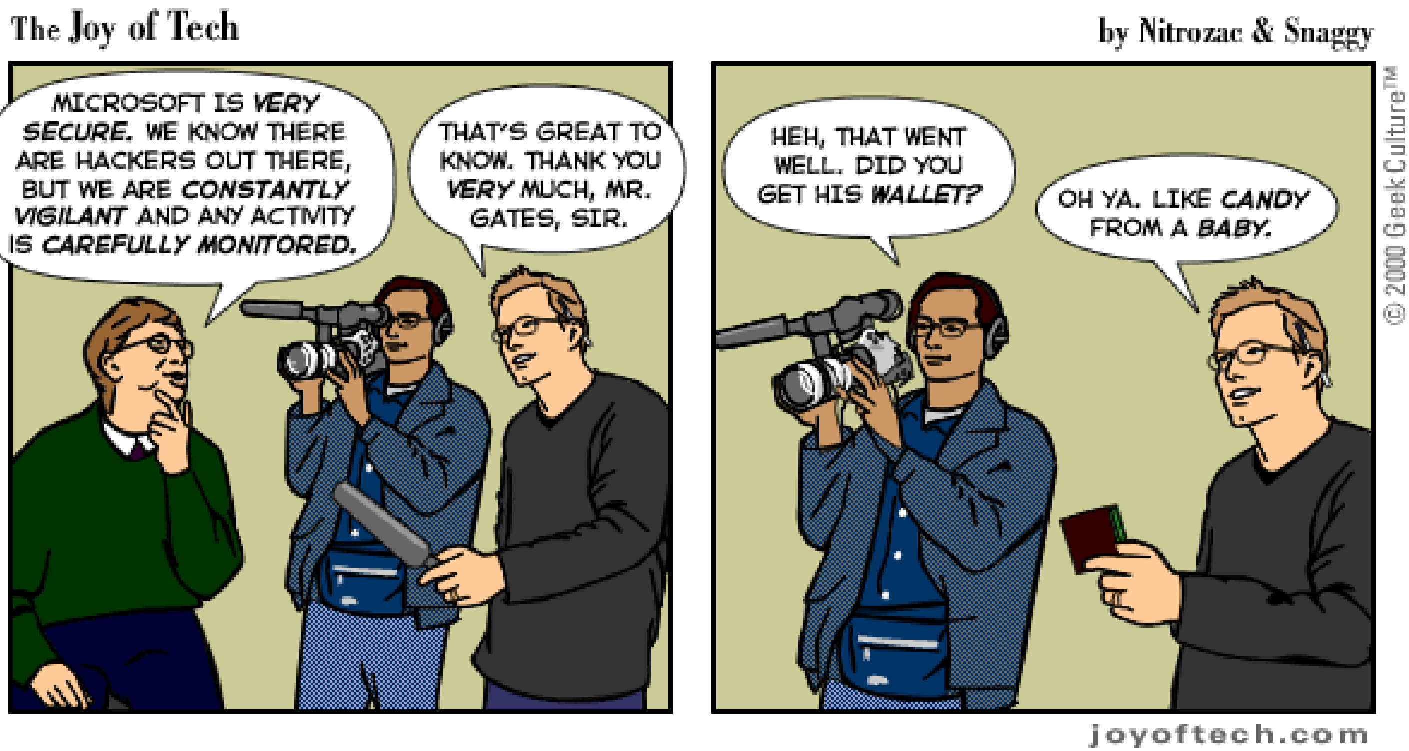
Where we are …

- Introduction to network security
- Vulnerabilities in IP
- I. CRYPTOGRAPHY
  - Symmetric Encryption and Message Confidentiality
  - Public-Key Cryptography and Message Authentication
- II. NETWORK SECURITY APPLICATIONS
  - Authentication Applications (Kerberos, X.509)
  - Electronic Mail Security (PGP, S/MIME)
  - IP Security (IPSec, AH, ESP, IKE)
  - Web Security (SSL, TLS, SET)
- III. SYSTEM SECURITY
  - Intruders and intrusion detection
  - Malicious Software (viruses)
  - Firewalls and trusted systems

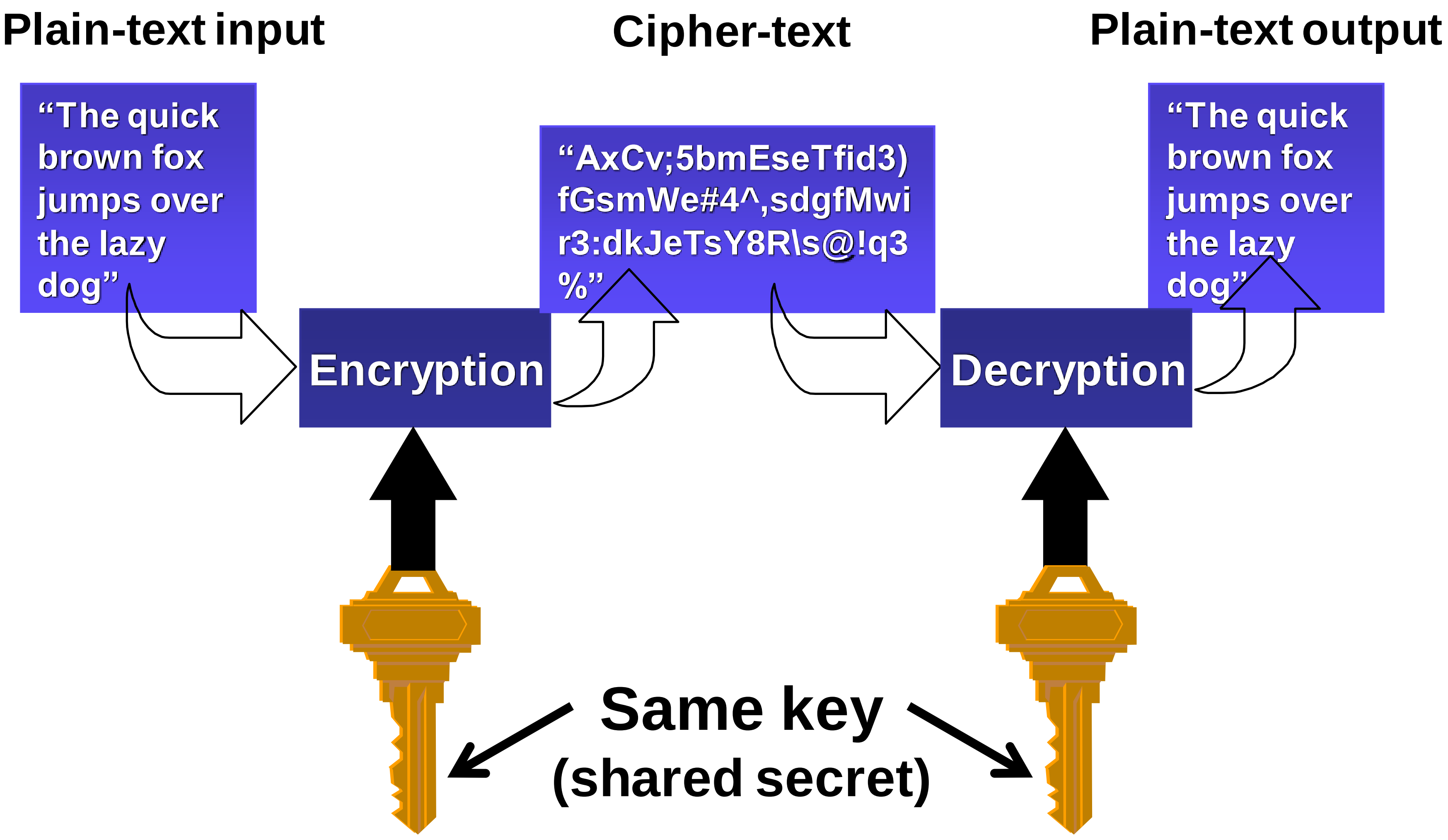
Confidentiality Using Symmetric Encryption

Encryption

only the basics



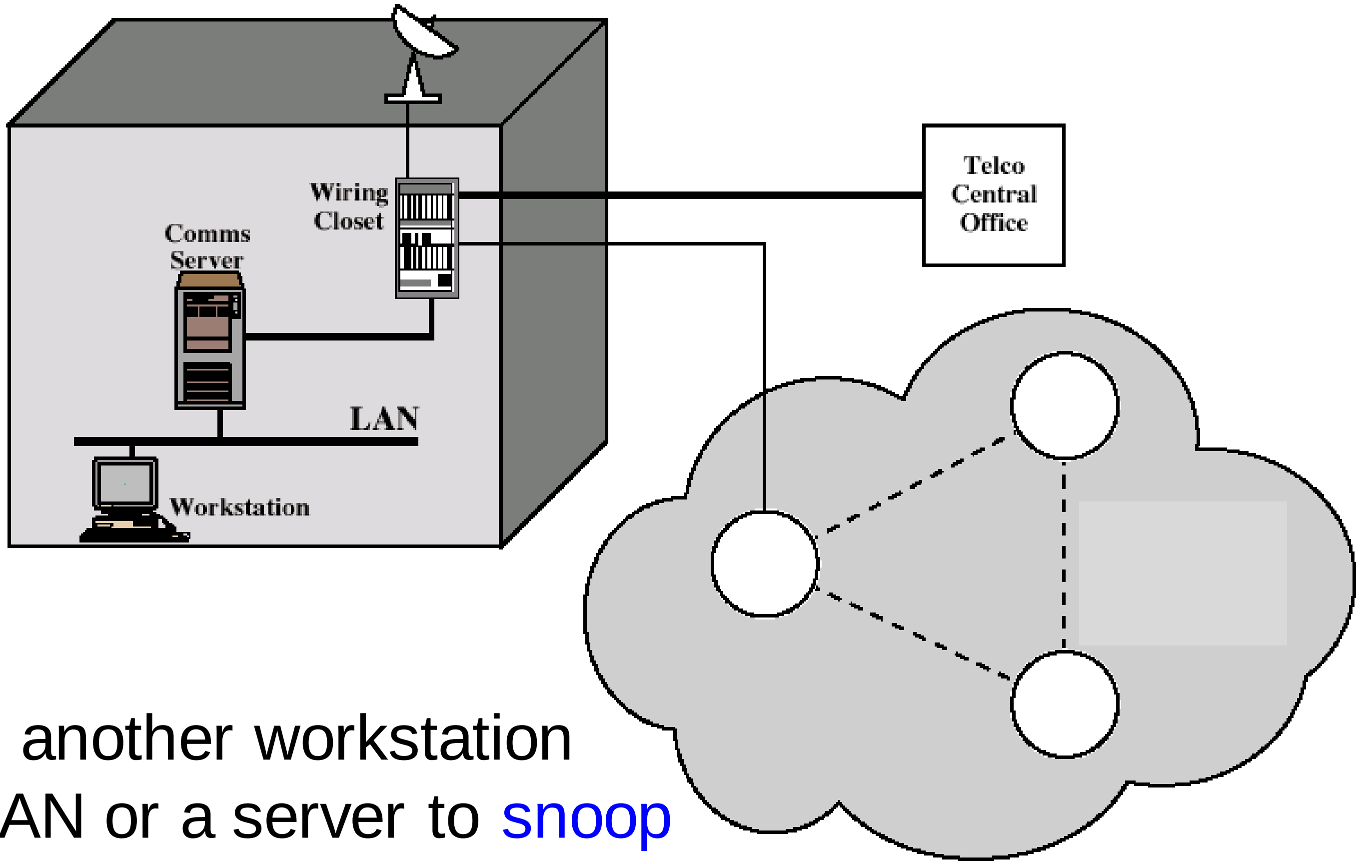
Symmetric Key Cryptography



Confidentiality using Symmetric Encryption

- Traditionally symmetric encryption is used to provide message confidentiality
- Consider a typical scenario
  - Workstations on LANs access other workstations & servers on LAN
  - LANs are interconnected using switches/routers
  - With external lines or radio/satellite links

Points of Vulnerability

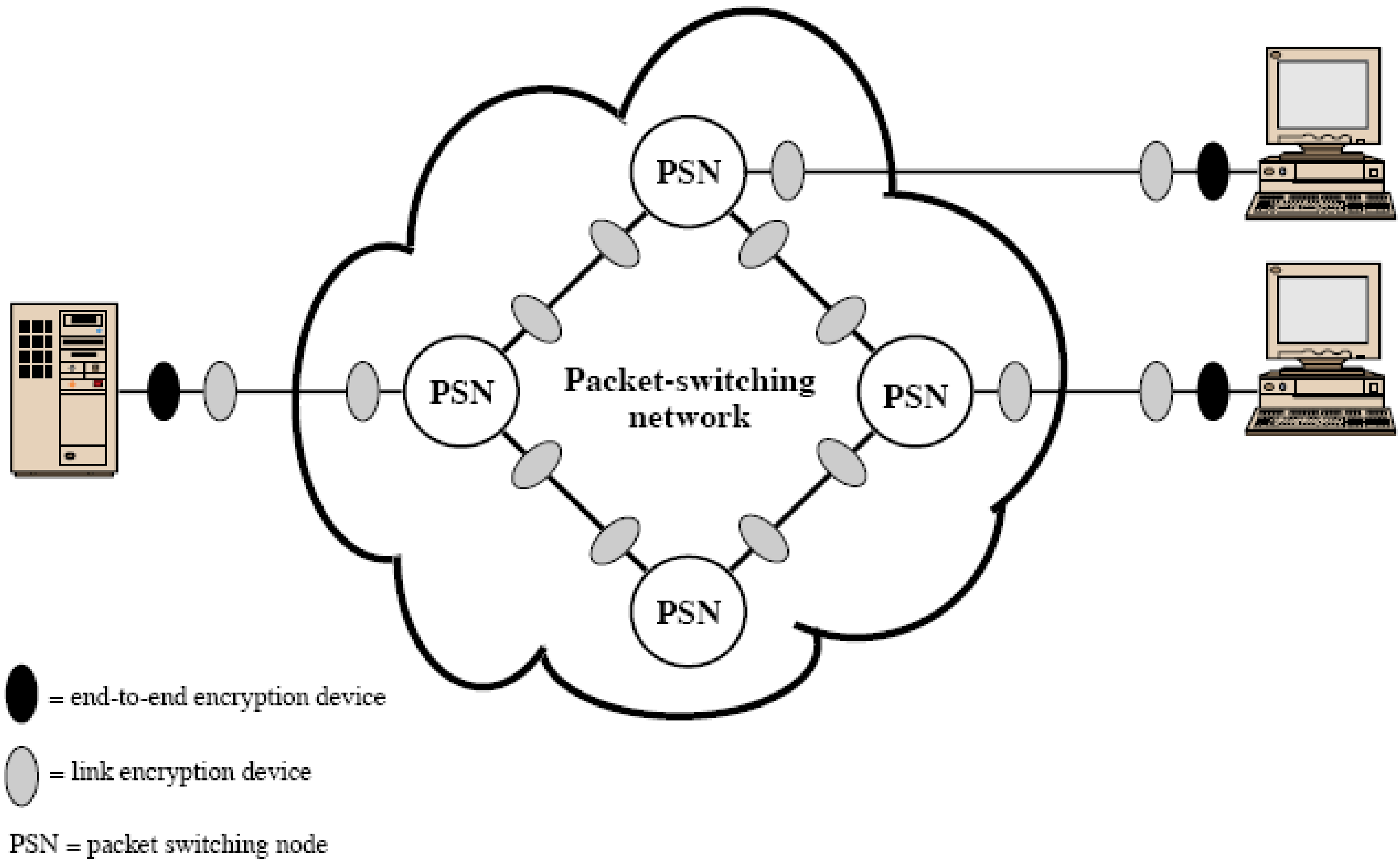


- snooping from another workstation
- connect to a LAN or a server to snoop
- use external router link to enter & snoop
- monitor and/or modify traffic on external links

Confidentiality using Symmetric Encryption

- Have two major placement alternatives
  - Link Encryption
  - End-to-End Encryption

Encryption Across a PSN



End-to-End Encryption

- Source encrypts and the Receiver decrypts
- Payload encrypted
- Header in the clear
- Only destination and reciever share the key
- Destination needs to be concerned about the degree of security in the network and links
- **High Security:** Both link and end-to-end encryptions are needed

Location of Encryption Device
Link Encryption

- Encryption devices are placed at each end of the link
- Encryption occurs independently on every link
- All the communication is made secure
- A lot of encryption devices are required
- Decrypt each packet at every switch
- High level of security

Link Encryption Implications

- All paths must use link encryption
- Each pair of node must share a unique key
  - Large number of keys should be provided

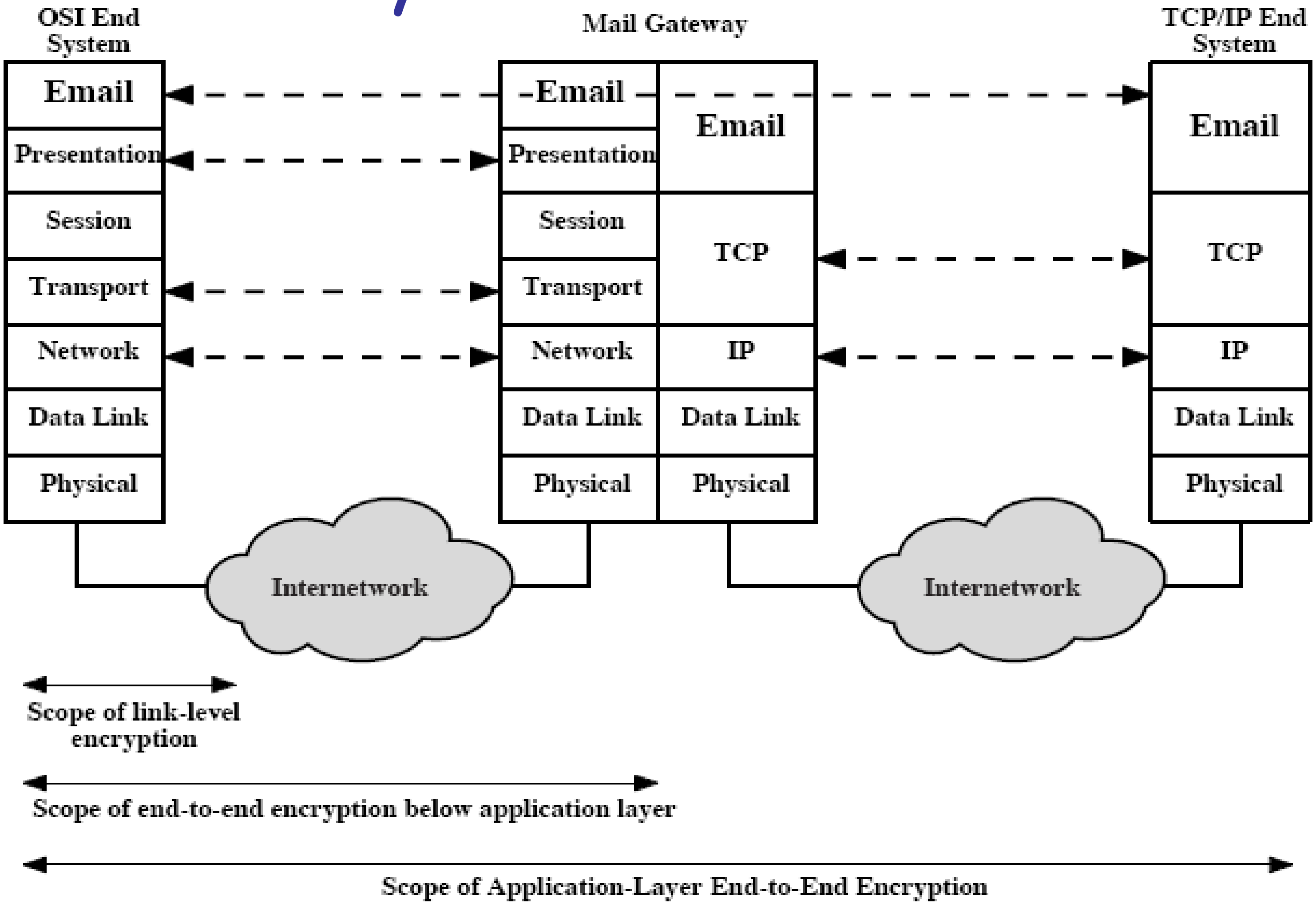
Traffic Analysis

- End-to-end encryption must leave headers in clear
  - So network can correctly route information
- Content may be protected, traffic flow patterns are not
- Ideally want both at once
  - End-to-End protects data contents over entire path and provides authentication
  - Link protects traffic flows from monitoring

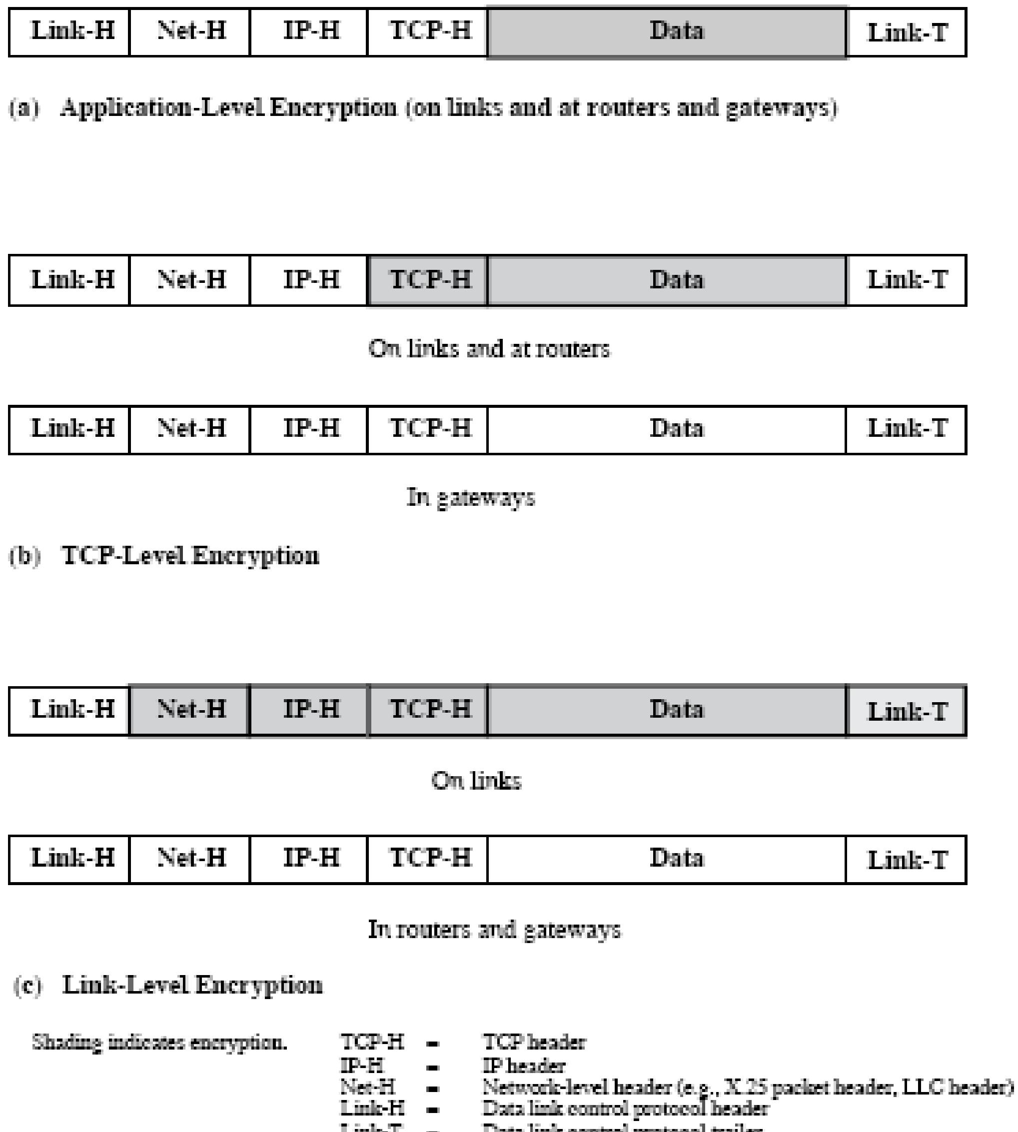
Placement of Encryption

- Can place encryption function at various layers in OSI Reference Model
  - Link encryption occurs at **layers 1 or 2**
  - End-to-End can occur at **layers 3, 4, 6, 7**
  - As move higher, less information is encrypted but it is more secure and more complex with more entities and keys

Encryption coverage implications at different layers



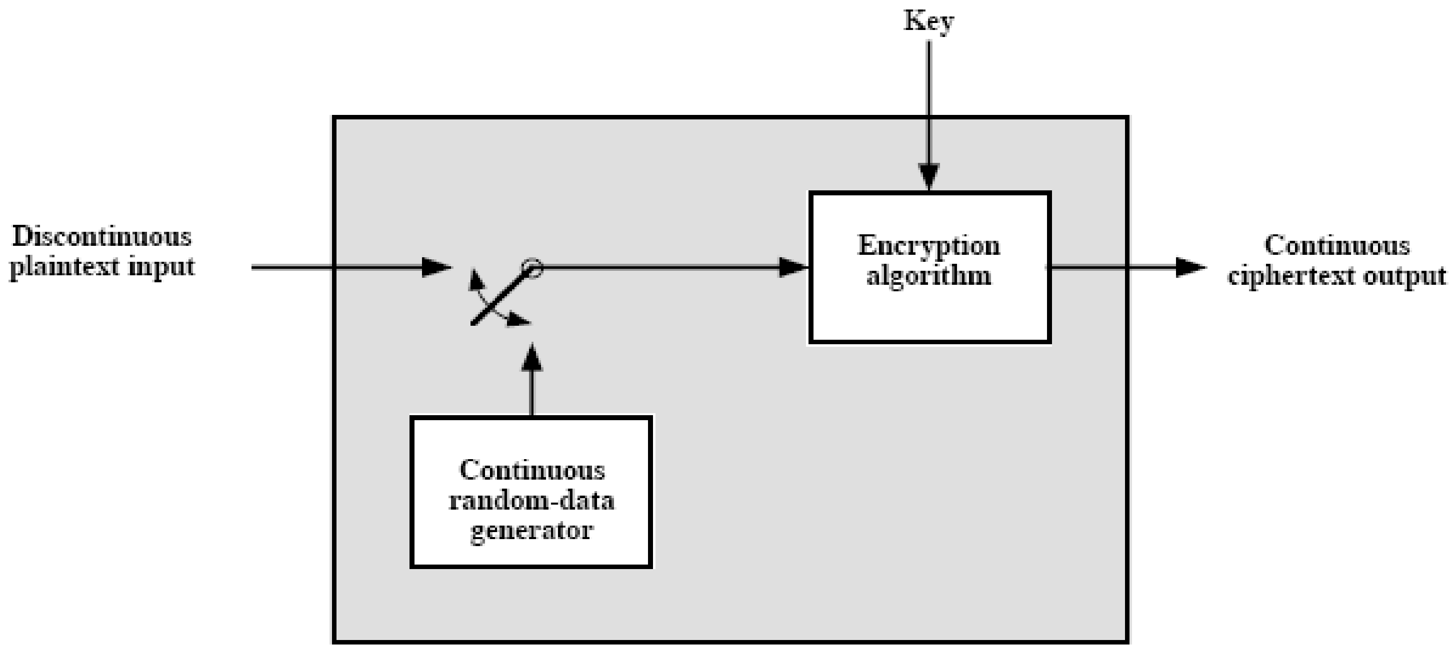
Encryption and Protocol Levels



Traffic Analysis

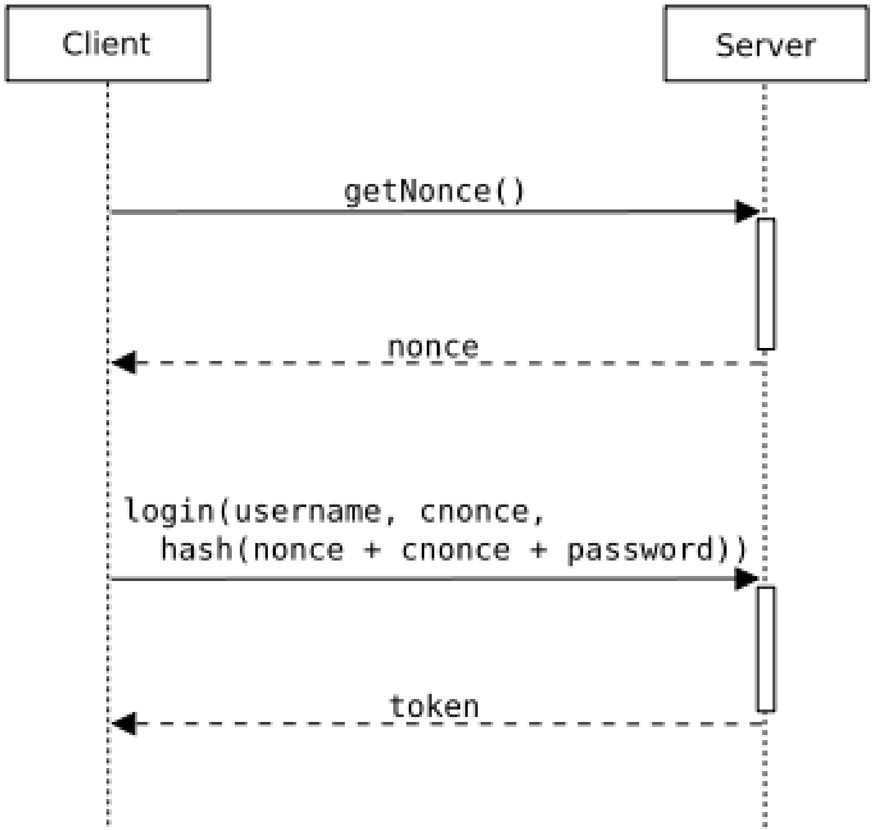
- Monitoring of communications flows between parties
  - Useful both in military & commercial spheres
  - Can also be used to create a covert channel
- **Link encryption** obscures header details
  - But overall traffic volumes in networks and at end-points is still visible
- **Traffic padding** can further obscure flows
  - But at cost of continuous traffic

Traffic Padding Encryption Device



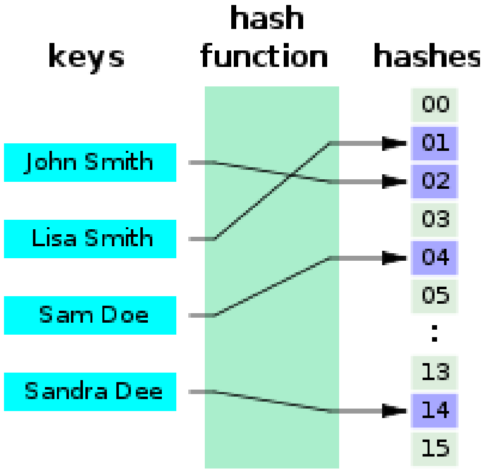
Nonce

- A random or pseudo-random number issued in an authentication protocol to avoid ***replay attacks***
- Must be time-variant (timestamp), or
- Generated with enough random bits

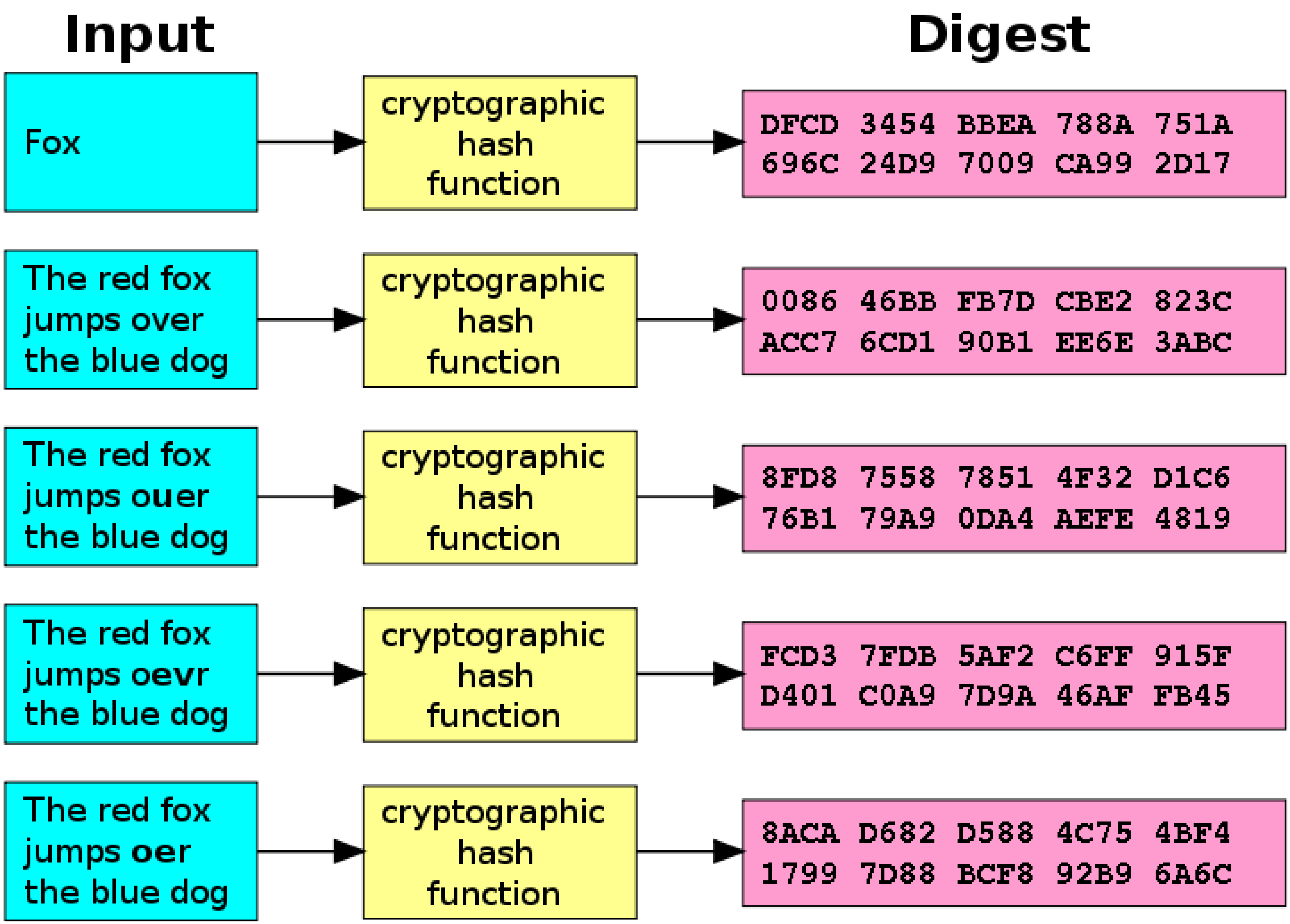


Cryptographic HASH Function

- A deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string
  - The encoded data => "**message**“
  - The hash value => “**message digest** or **digest**”
  - **SHA-1, MD-5, MAC etc**
- Easy to compute for any message
- **Reverse Engineering** not possible
- Always result in unique value,
- Unique message to digest pair



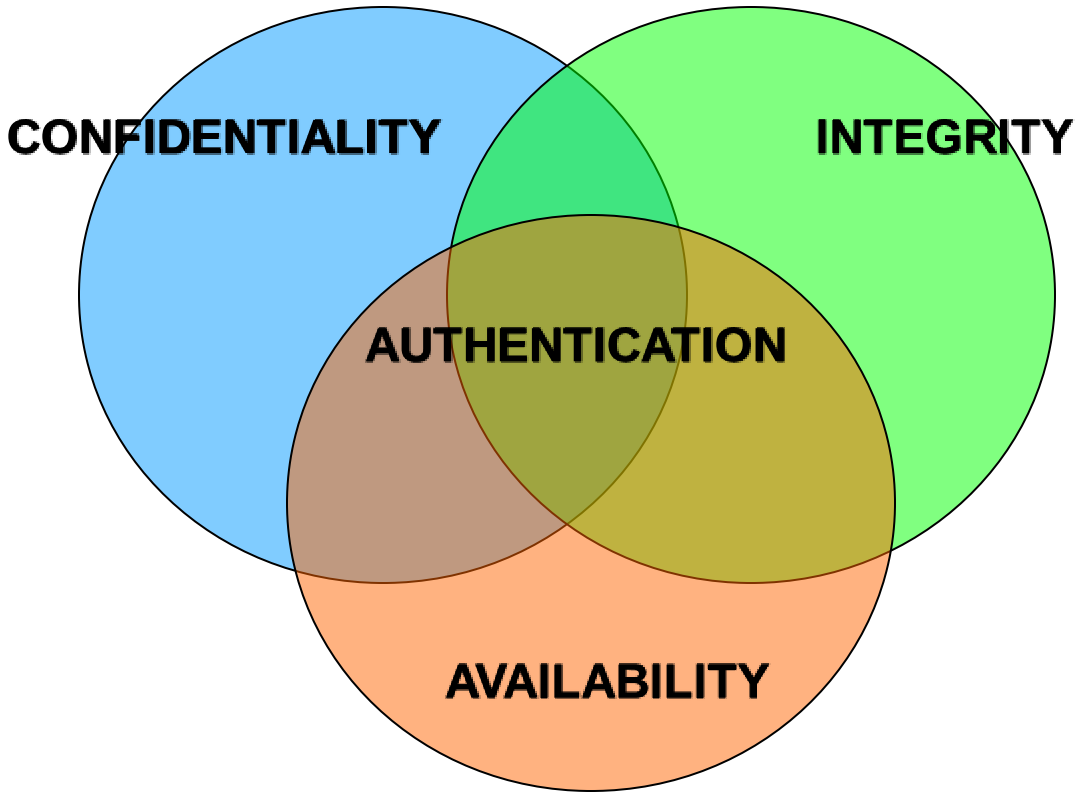
SHA-1, example



Non Irreversible Cryptography?

- Is not the term **SENSELESS**?

Required Key Protection



asimrasheed@mcs.edu.pk

Key Storage

- **In Files**

–Encryption + MAC based on a password

–Using access control of operating system

–Encryption + MAC (or signature) with other keys

- **In Crypto Tokens**

–Smart card, USB crypto token, ...

–Supports complete key life-cycle on token

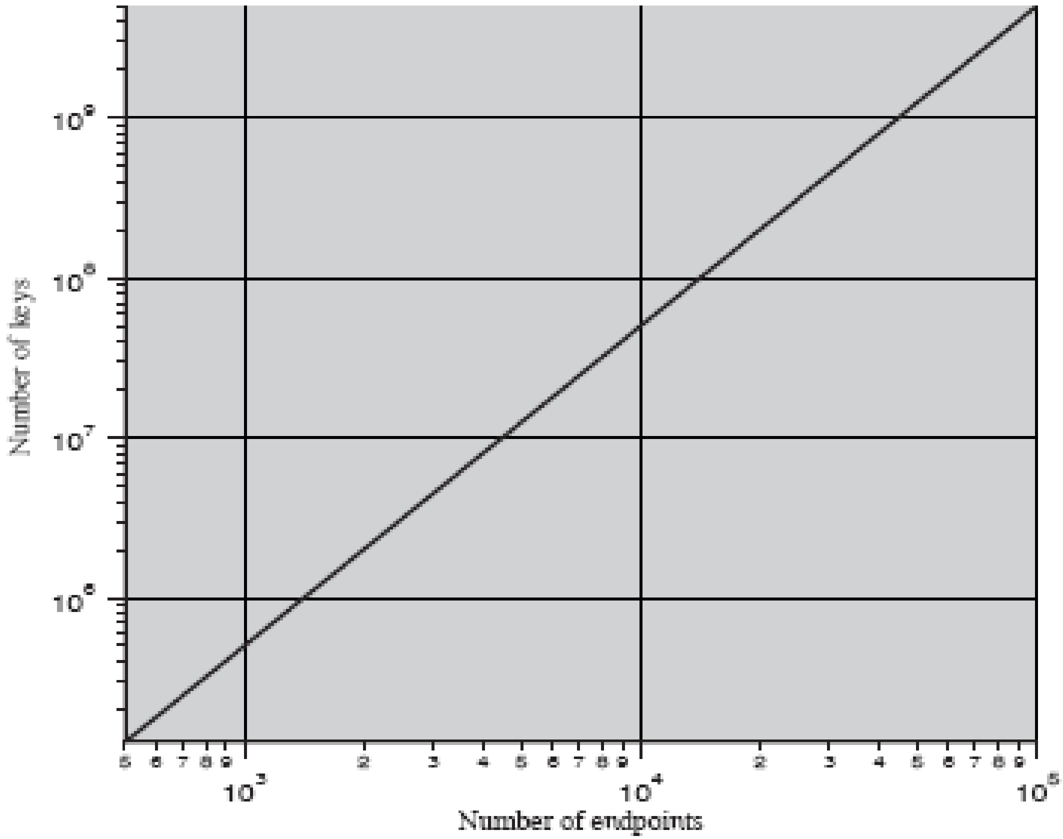
•Generation – storage – use – destruction

–provide means to ensure that there is no way to get a key out

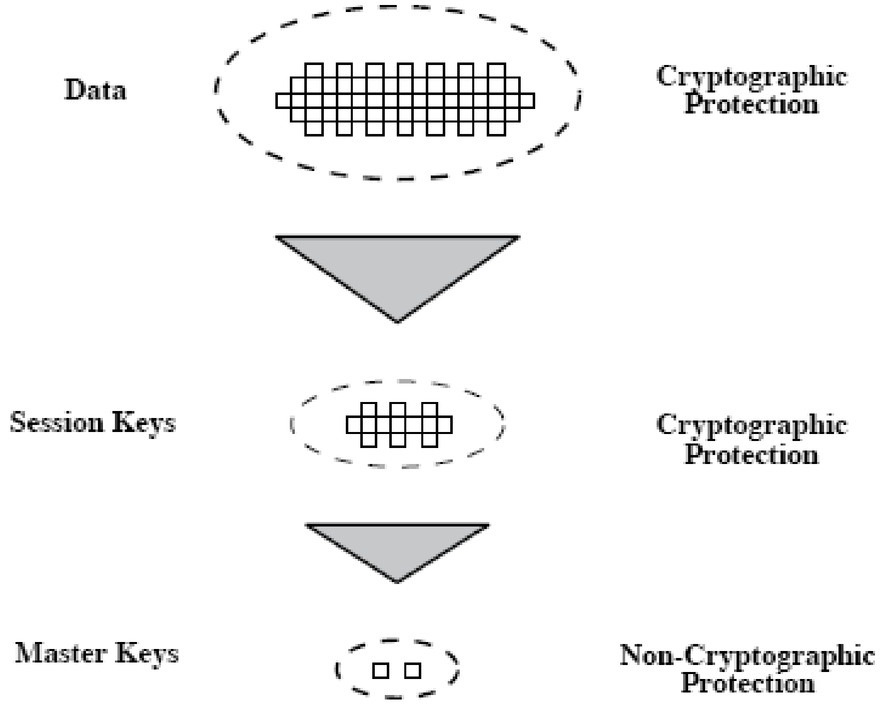
- **Key Backup** (also known as key escrow)

–Usually required for decryption keys

Number of keys required to support Arbitrary connections



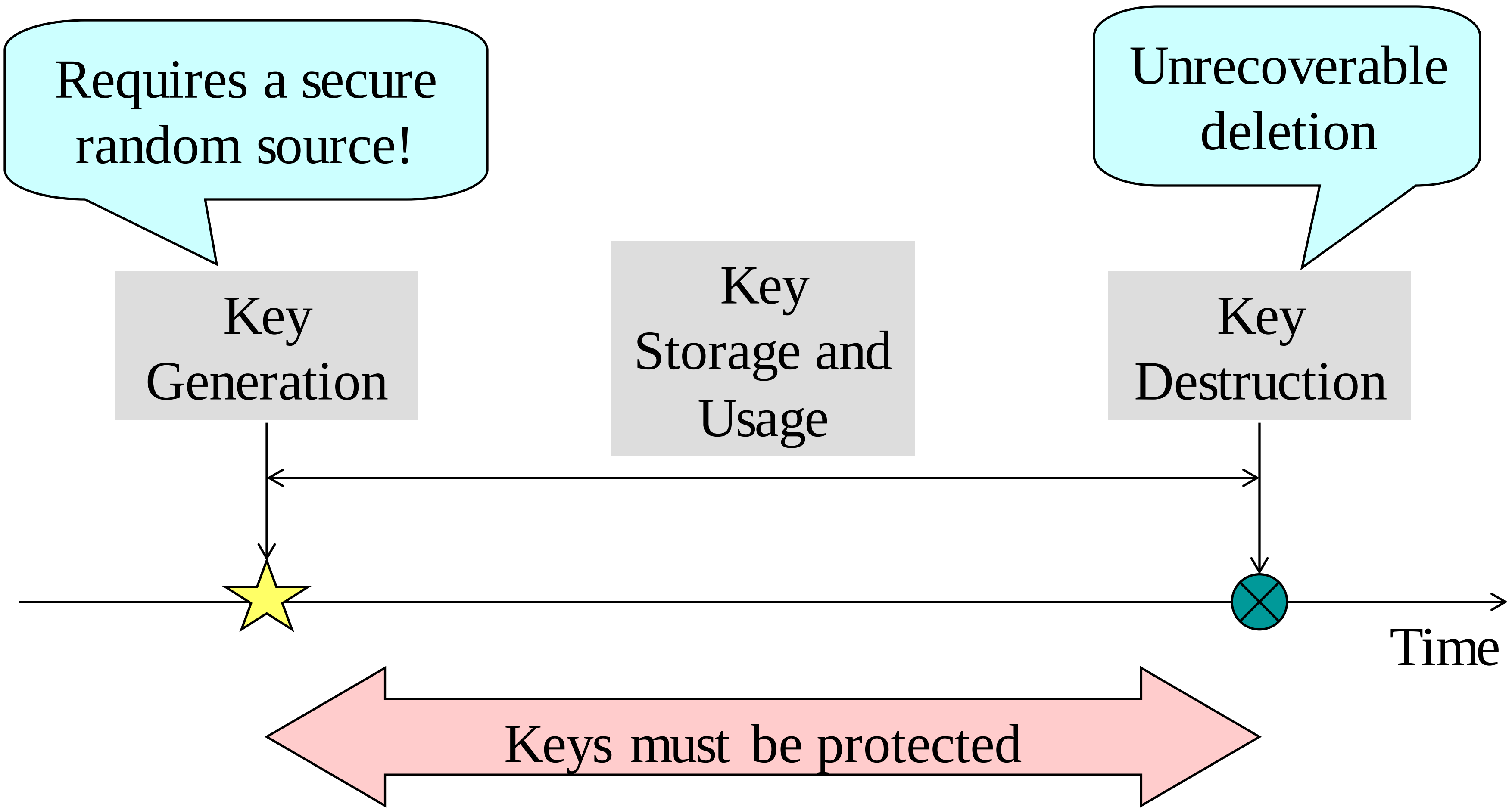
Use of a Key Hierarchy



Key Renewal

- Keys should be renewed
- More available cipher texts may facilitate certain attacks
- How often depends on the crypto algorithm
  - Can depend on the amount of encrypted data
  - May depend on time (exhaustive key search requires time)
- Regular key renewal can reduce damage in case of (unnoticed) key compromise
- Protocols like SSL/TLS include features for (secret) key renewal

Key Life-Cycle



Key Distribution

- Means of Exchanging Keys between two parties
- Keys are used for conventional encryption
- Frequent key exchanges are desirable
  – Limiting the amount of data compromised
- Strength of cryptographic system rests with Key Distribution Mechanism

Key Distribution

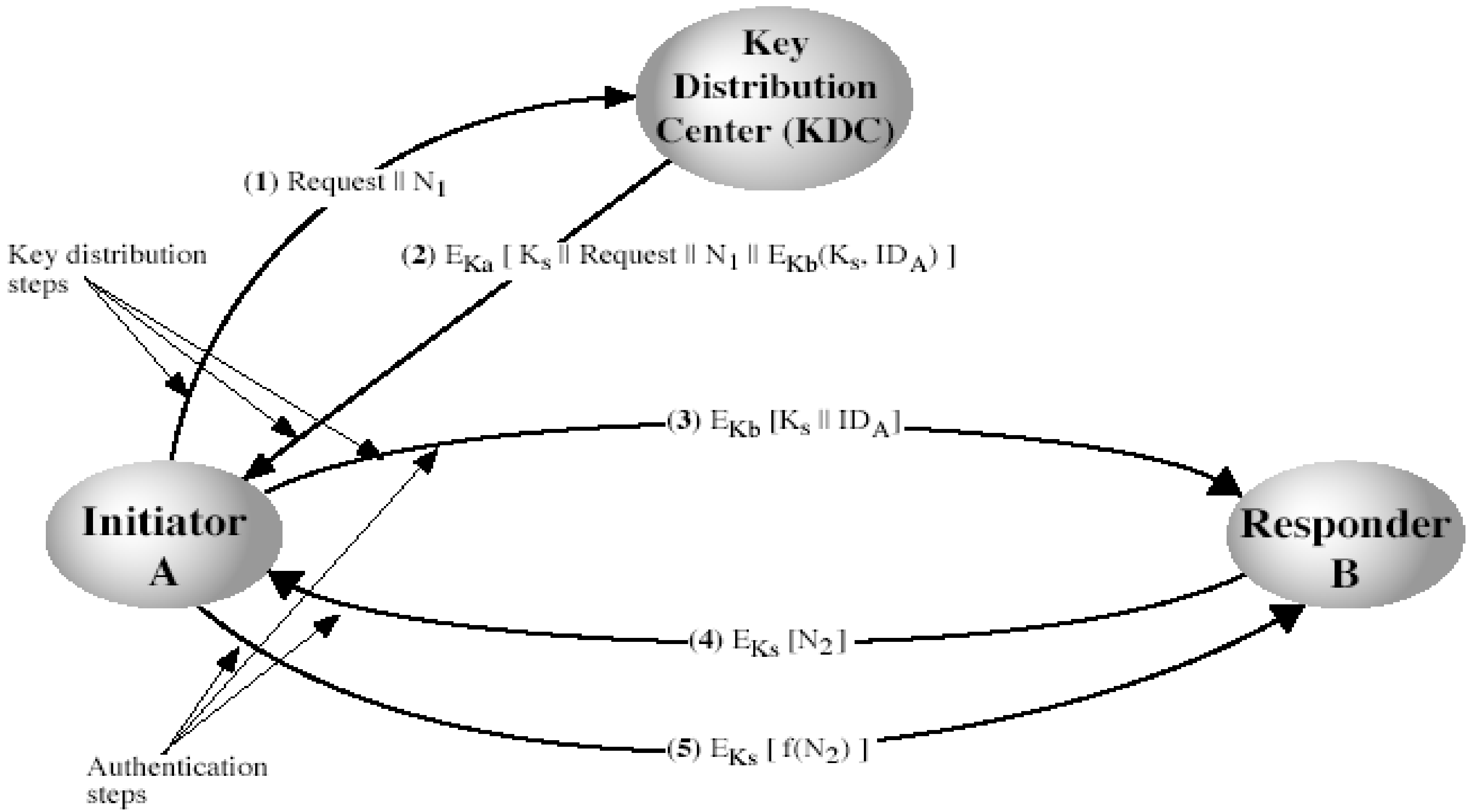
- Symmetric schemes require both parties to share a common secret key
- Issue is how to securely distribute this key
- Often a secure system failure due to a break in the key distribution scheme

Key Distribution

- Two parties A and B can have various **key distribution** alternatives:

1. A can select key and physically deliver to B
2. third party can select & deliver key to A & B
3. if A & B have communicated previously can use previous key to encrypt a new key
4. if A & B have secure communications with a third party C, C can relay key between A & B

Key Distribution Scenario



asimrasheed@mcs.edu.pk

Key Distribution Scenario

1. A issues a request to the KDC for a session key
   - Nonce is also sent
   - Nonce includes identities of communicating parties and a unique value
2. KDC sends a response encrypted with A's secret key K_A
   - It includes one time session key K_S
   - Original request message, including the nonce
   - Message also includes K_S and ID of A encrypted with KB intended for B

Key Distribution Scenario

1. A stores K_S and forwards information for B i.e., $E_{K_B}[K_S||ID_A]$
2. B sends a nonce to A encrypted with K_S
3. A responds by performing some function on nonce like incrementing

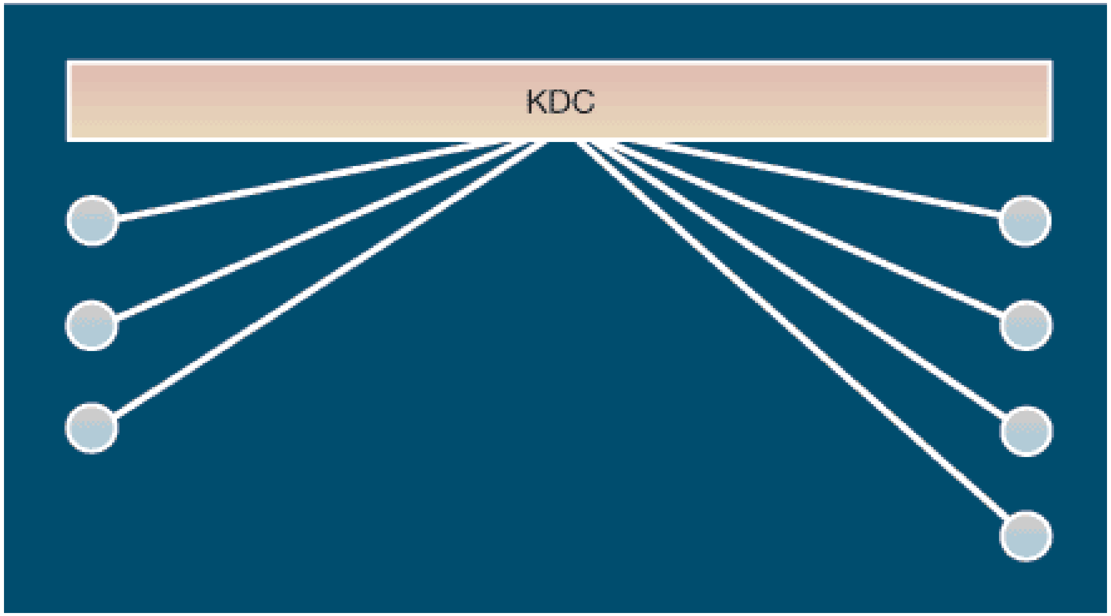
The last two steps assure B that the message it received was not a replay

Key Distribution Entities

- **Key Distribution Center**
  - Provides one time session key to valid users for encryption
- **Front end Processor**
  - Carries out the end to end encryption
  - Obtains session key from the KDC on behalf of its host

Key distribution for symmetric keys

- Key distribution for symmetric keys by a central server (KDC):
  - fixed number of distributions (for given n)
  - However, need security protocol



Key Distribution Issues with Hierarchical Key Control

- Not suitable that a single KDC is used for all the users
- Hierarchies of KDC's required for large networks
- A single KDC may be responsible for a small number of users since it shares the master keys of all the entities attached to it
- If two entities in different domains want to communicate, local KDCs communicate through a global KDC
- Must trust each other

Session Key Lifetimes

- Session key lifetimes should be limited for greater security
- More frequently the session keys are exchanged, more secure they become
- For connection oriented protocols, it should be valid for the duration of connection
- For connectionless protocols key should be valid for a certain duration

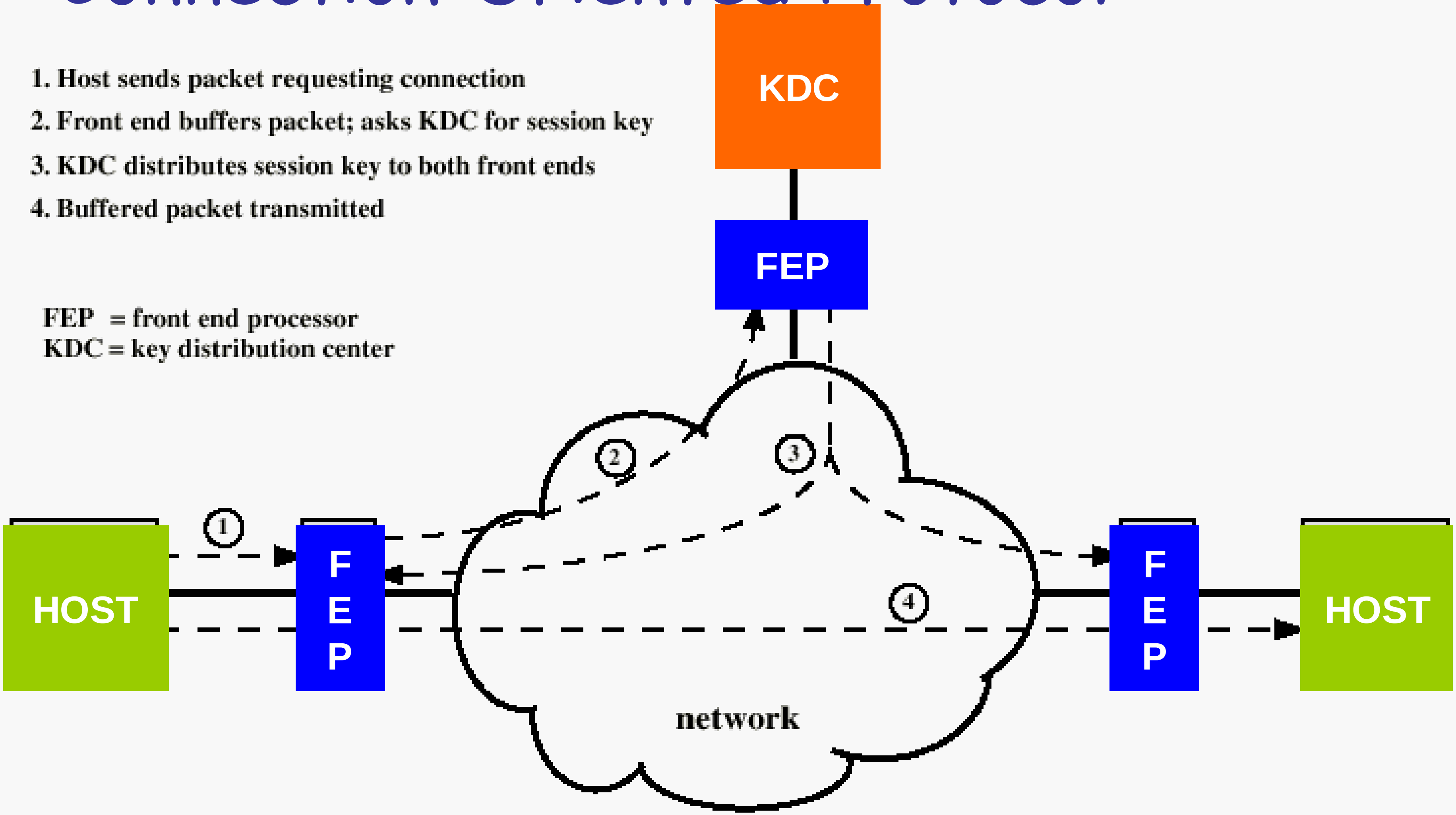
Transparent Key Control

- Use of automatic key distribution on behalf of users, but must trust system
  - Host sends packet requesting connection
  - Front End buffers packet; asks KDC for session key
  - KDC distributes session key to both front ends
  - Buffered packet transmitted

Automatic Key Distribution for Connection-Oriented Protocol

1. Host sends packet requesting connection
2. Front end buffers packet; asks KDC for session key
3. KDC distributes session key to both front ends
4. Buffered packet transmitted

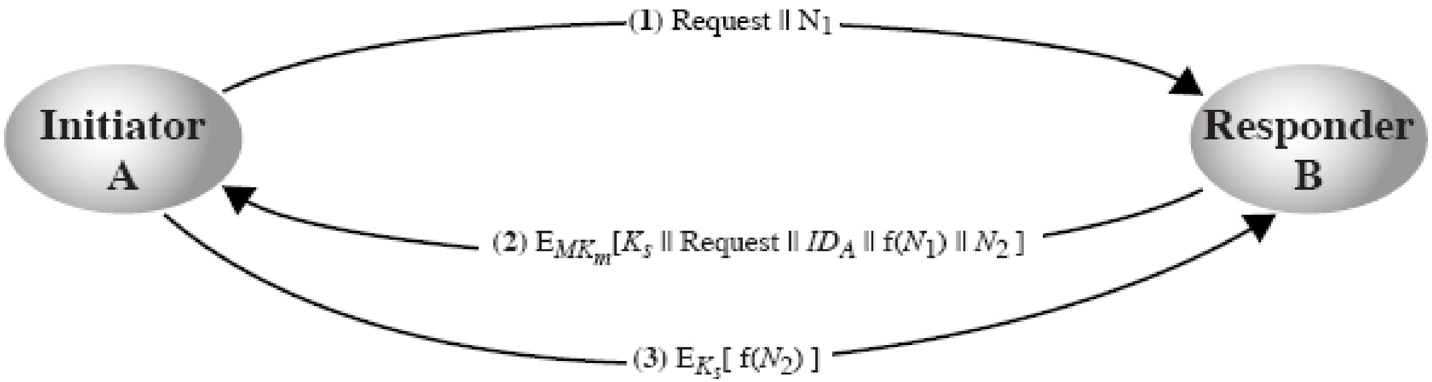
FEP = front end processor
KDC = key distribution center



Decentralized Key Control

- KDCs need to be trusted and protected
- Can be avoided by using decentralized distribution
- Decentralized approach requires that each node be able to communicate in a secure manner
- Session key may be established in following way
  - A issues a request to B for a session key and includes a nonce, N1
  - B responds with a message that is encrypted using the shared secret key
  - Response includes session key, ID of B, the value f(N1) and nonce N2
  - Using the new session key, A returns f(N2) to B

Decentralized Key Distribution



Controlling Key Usage

- Different types of session keys e.g.,
  - **Data encrypting key**: for general communication across network
  - **PIN-encrypting key**: for PIN used in electronic funds
  - **File encrypting key**: for encrypting files stored on a publicly accessible location
- Avoid using master key instead of session key as any unauthorized application may obtain the master key and exploit
- Controlling purposes keys are used
- Associate a tag or a control vector to specify where and how the key should be used

Key Distribution

- **Session key**
  - Data encrypted with a one-time session key. At the conclusion of the session, the key is destroyed
- **Permanent key**
  - Used between entities for the purpose of distributing session keys

Summary

- Have considered:
  - use of symmetric encryption to protect confidentiality
  - need for good key distribution
  - use of trusted third party KDC's

Any question ?