IS 2150 / TEL 2810
Introduction to Security

James Joshi
Assistant Professor, SIS

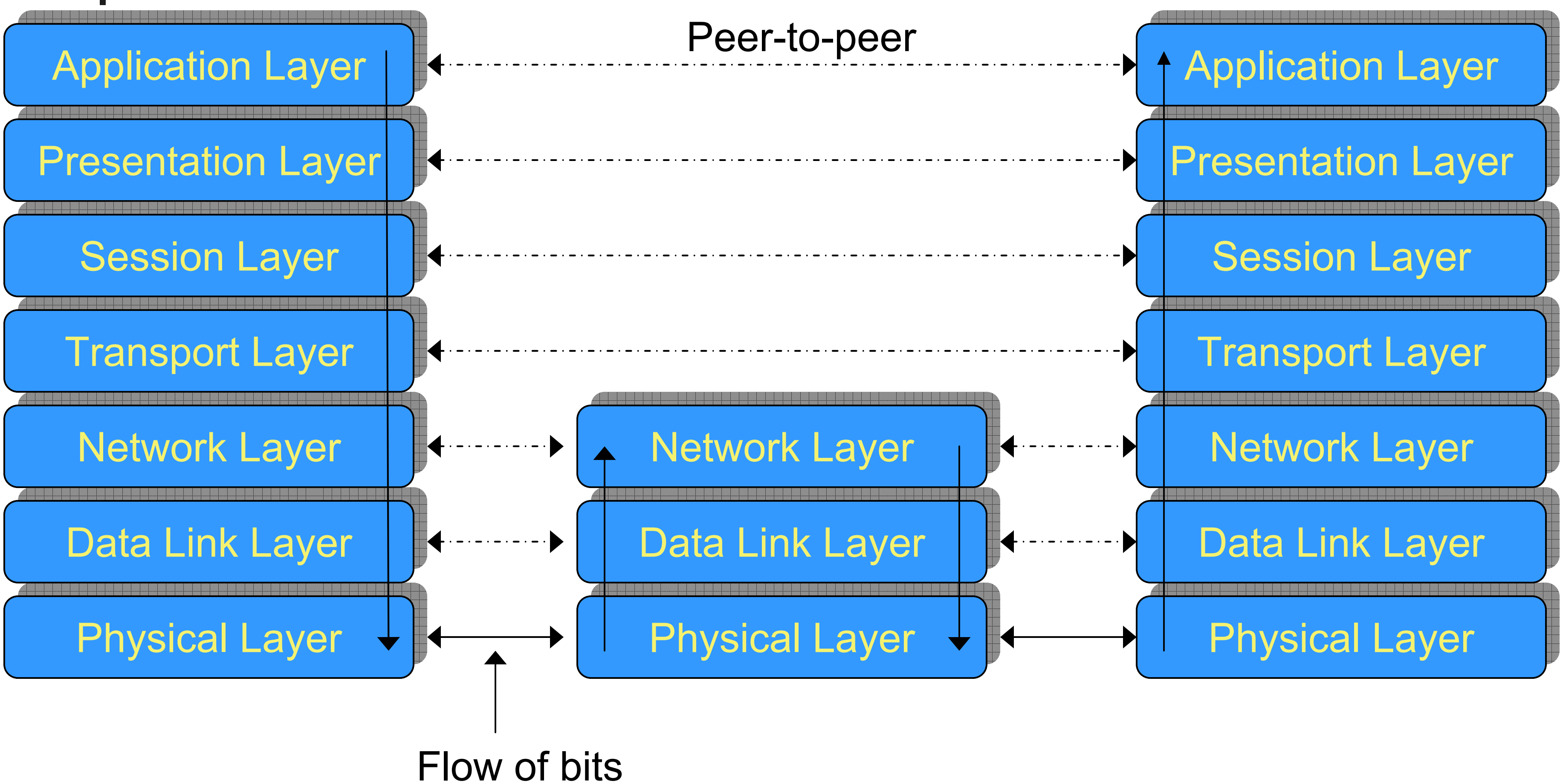
Lecture 10
Nov 15, 2007

Network Security, Authentication, Identity

Objectives

- Understand/explain the issues related to, and utilize the techniques
  - Security at different levels of OSI model
    - Privacy Enhanced email
    - IPSec
    - Misc.
  - Authentication and identification
    - password

ISO/OSI Model

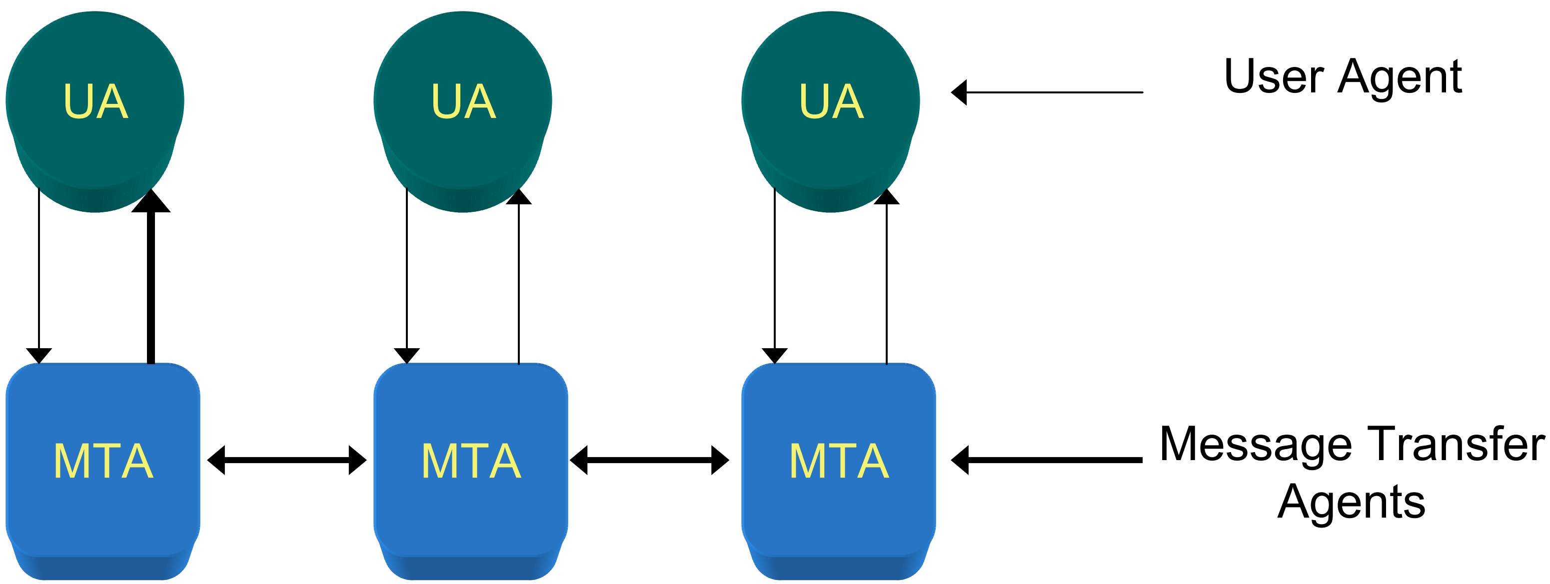


Protocols

- End-to-end protocol
  - Communication protocol that involves end systems with one or more intermediate systems
  - Intermediate host play no part other than forwarding messages
    - Example: telnet
- Link protocol
  - Protocol between every directly connected systems
    - Example: IP – guides messages from a host to one of its immediate host
- Link encryption
  - Encipher messages between intermediate host
  - Each host share a cryptographic key with its neighbor
    - Attackers at the intermediate host will be able to read the message
- End-to-end encryption
  - Example: telnet with messages encrypted/decrypted at the client and server
  - Attackers on the intermediate hosts cannot read the message

Electronic Mail

- UA interacts with the sender
- UA hands it to a MTA
- Attacker can read email on any of the computer with MTA
- Forgery possible



Security at the Application Layer: Privacy-enhanced Electronic Mail

- Study by Internet Research Task Force on Privacy or Privacy Research Group to develop protocols with following services
  - Confidentiality, by making the message unreadable except to the sender and recipients
  - Origin authentication, by identifying the sender precisely
  - Data integrity, by ensuring that any changes In the message are easy to detect
  - Non-repudiation of the origin (if possible)

Design Considerations/goals for PEM

- Not to redesign existing mail system protocols
- To be compatible with a range of MTAs, UAs and other computers
- To make privacy enhancements available separately so they are not required
- To enable parties to use the protocol to communicate without prearrangement

PEM
Basic Design

- Defines two keys
  - Data Encipherment Key (DEK) to encipher the message sent
    - Generated randomly
    - Used only once
    - Sent to the recipient
  - Interchange key: to encipher DEK
    - Must be obtained some other way than through the message

Protocols

- Confidential message (DEK: k_s)

$$\text{Alice} \xrightarrow{\{m\}k_s \,\|\, \{k_s\}k_{\text{Bob}}} \text{Bob}$$

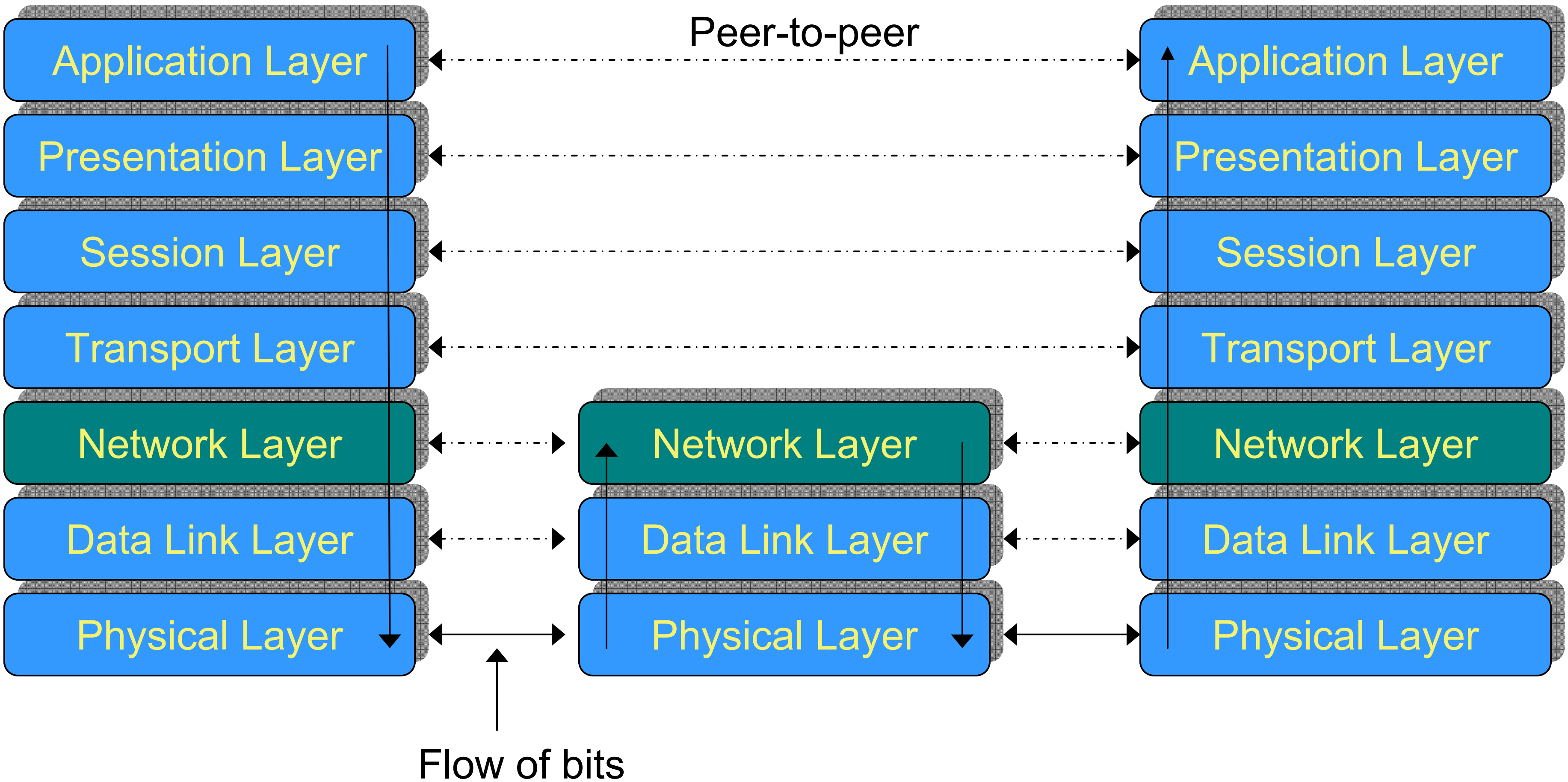
- Authenticated, integrity-checked message

$$\text{Alice} \xrightarrow{m \,\|\, \{h(m)\}k_{\text{Alice}}} \text{Bob}$$

- Enciphered, authenticated, integrity checked message

$$\text{Alice} \xrightarrow{??} \text{Bob}$$

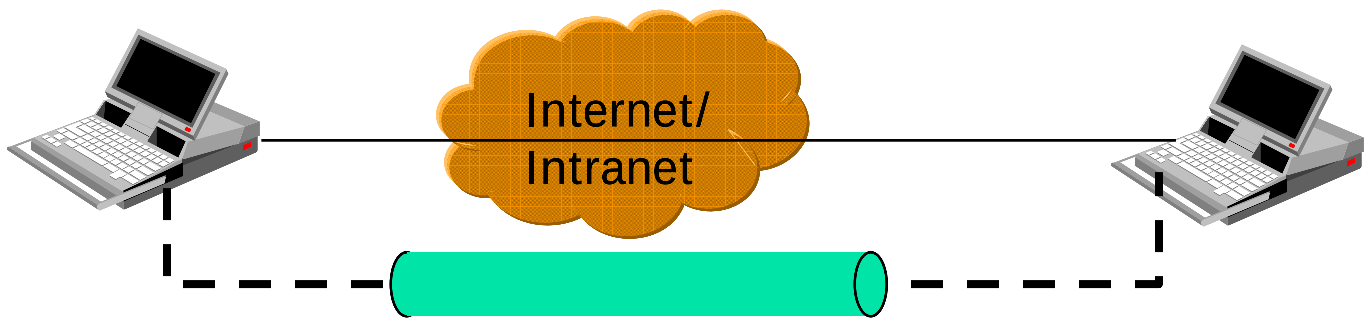
ISO/OSI Model
IPSec: Security at Network Layer



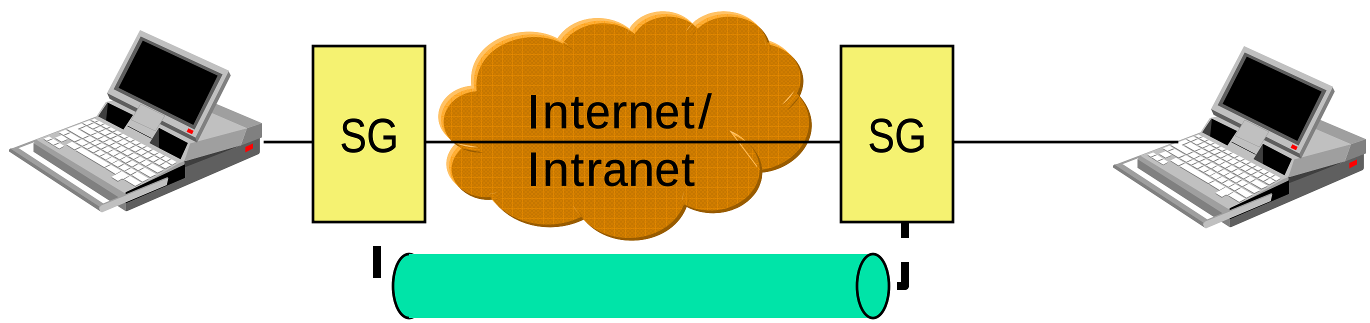
IPSec

- Set of protocols/mechanisms
  - Encrypts and authenticates all traffic at the IP level
    - Protects all messages sent along a path
    - Intermediate host with IPSec mechanism (firewall, gateway) is called a *security gateway*
- Application independent (Transparent to user)
  - Web browsing, telnet, ftp...
- Provides at the IP level
  - Access control
  - Connectionless integrity
  - Data origin authentication
  - Rejection of replayed packets
  - Data confidentiality
  - Limited traffic analysis confidentiality

Cases where IPSec can be used

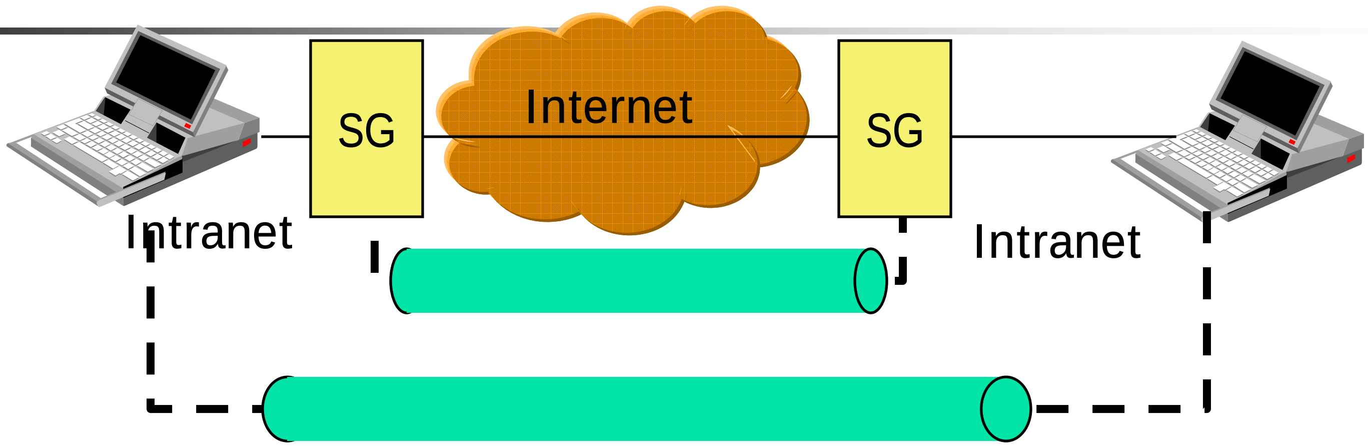


End-to-end security between two hosts

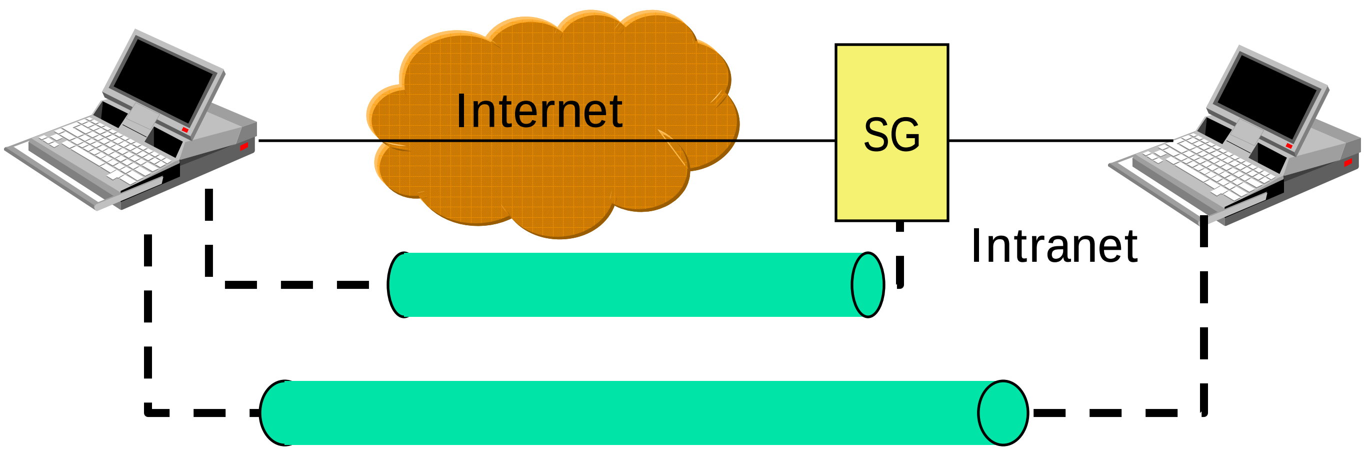


End-to-end security between two security gateways

Cases where IPSec can be used (2)



End-to-end security between two hosts + two gateways



End-to-end security between two hosts during dial-up

IPSec Protocols

- Authentication header (AH) protocol
  - Message integrity
  - Origin authentication
  - Anti-replay services
- Encapsulating security payload (ESP) protocol
  - Confidentiality
  - Message integrity
  - Origin authentication
  - Anti-replay services
- Internet Key Exchange (IKE)
  - Exchanging keys between entities that need to communicate over the Internet
  - What authentication methods to use, how long to use the keys, etc.

Security Association (SA)

- Unidirectional relationship between peers
- Specifies the security services provided to the traffic carried on the SA
  - Security enhancements to a channel along a path
- Identified by three parameters:
  - IP Destination Address
  - Security Protocol Identifier
    - Specifies whether AH or ESP is being used
  - Security Parameters Index (SPI)
    - Specifies the security parameters associated with the SA

Security Association (2)

- Each SA uses AH or ESP (not both)
  - If both required two SAs are created
- Multiple security associations may be used to provide required security services
  - A sequence of security associations is called *SA bundle*
  - Example: We can have an AH protocol followed by ESP or vice versa

Security Association Databases

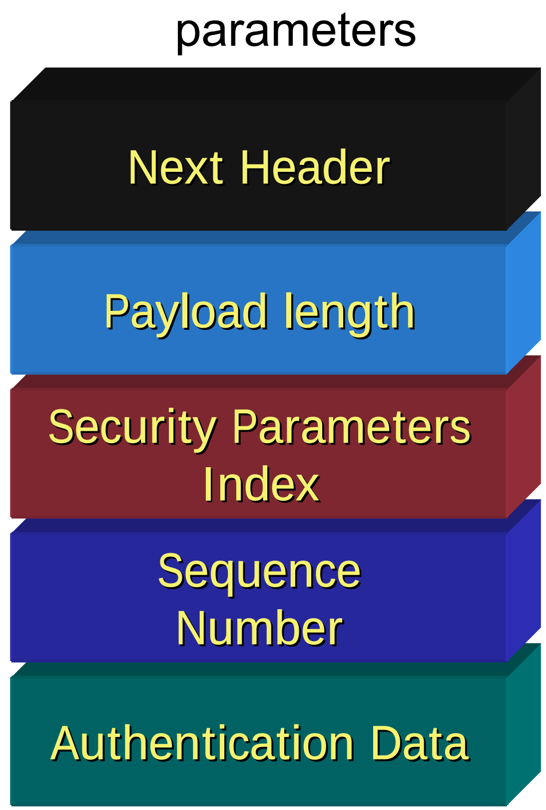
- IP needs to know the SAs that exist in order to provide security services
- Security Policy Database (SPD)
  - IPSec uses SPD to handle messages
    - For each IP packet, it decides whether an IPSec service is provided, bypassed, or if the packet is to be discarded
- Security Association Database (SAD)
    - Keeps track of the sequence number
    - AH information (keys, algorithms, lifetimes)
    - ESP information (keys, algorithms, lifetimes, etc.)
    - Lifetime of the SA
    - Protocol mode
    - MTU et.c.

IPSec Modes

- Two modes
  - Transport mode
    - Encapsulates IP packet data area
    - IP Header is not protected
      - Protection is provided for the upper layers
      - Usually used in host-to-host communications
  - Tunnel mode
    - Encapsulates entire IP packet in an IPSec envelope
      - Helps against traffic analysis
      - The original IP packet is untouched in the Internet

Authentication Header (AH)

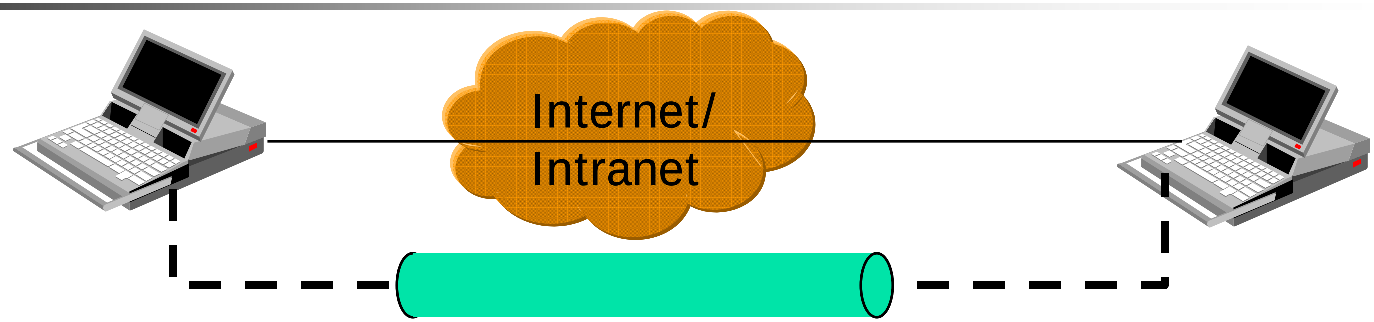
- Next header
  - Identifies what protocol header follows
- Payload length
  - Indicates the number of 32-bit words in the authentication header
- Security Parameters Index
  - Specifies to the receiver the algorithms, type of keys, and lifetime of the keys used
- Sequence number
  - Counter that increases with each IP packet sent from the same host to the same destination and SA
- Authentication Data



Preventing replay

- Using 32 bit sequence numbers helps detect replay of IP packets
- The sender initializes a sequence number for every SA
  - Each succeeding IP packet within a SA increments the sequence number
- Receiver implements a window size of W to keep track of authenticated packets
- Receiver checks the MAC to see if the packet is authentic

Transport Mode AH



Original IP Header	TCP Header	Payload Data

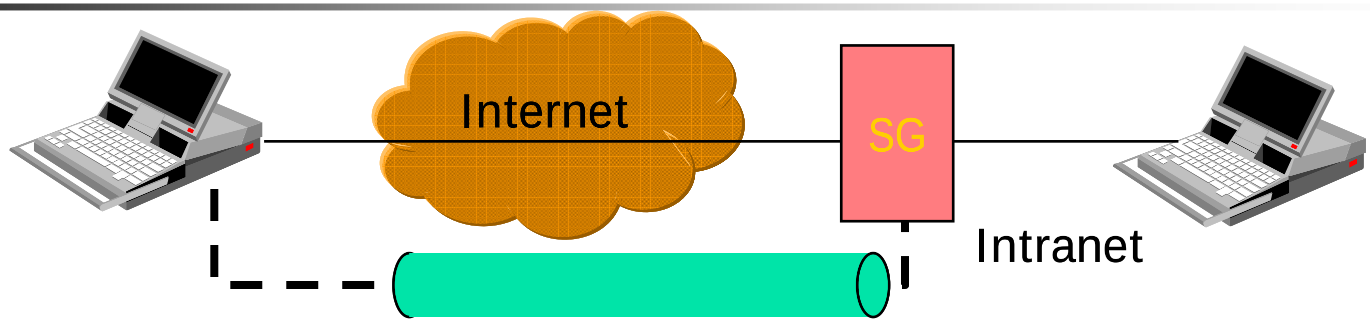
Without IPSec

Original IP Header	Auth Header	TCP Header	Payload Data

Next Header	Payload Length	SPI	Seq. No.	MAC

Authenticate
IP Payload

Tunnel Mode AH



Original IP Header	TCP Header	Payload Data

Without IPSec

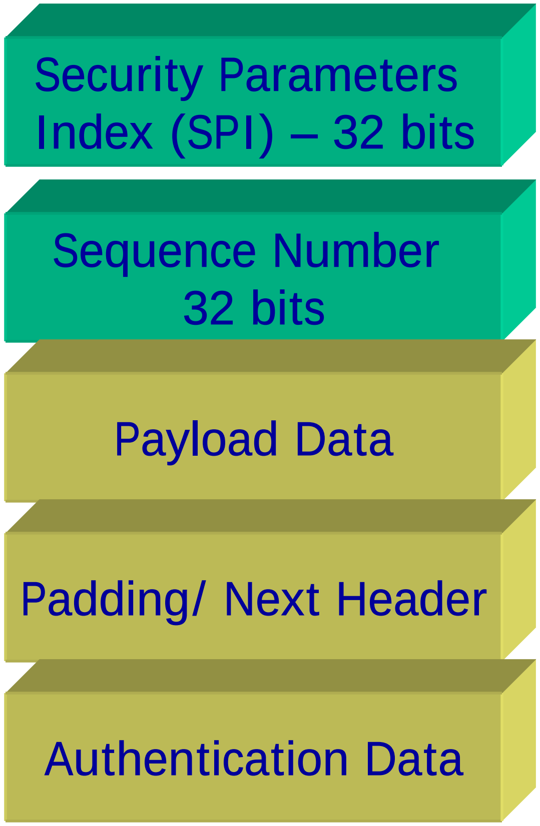
New IP Header	Auth Header	Original IP Header	TCP Header	Payload Data

Next Header	Payload Length	SPI	Seq. No.	MAC

Authenticate
Entire IP Packet

ESP – Encapsulating Security Payload

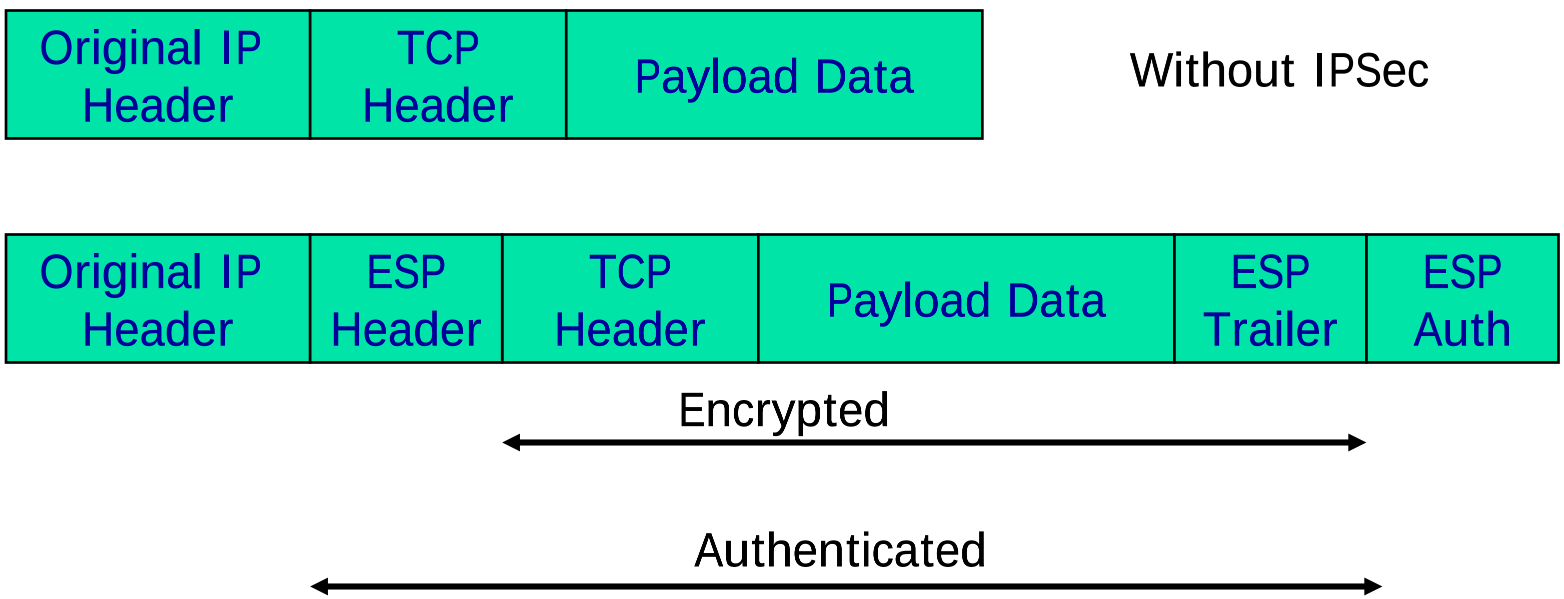
- Creates a new header in addition to the IP header
- Creates a new trailer
- Encrypts the payload data
- Authenticates the security association
- Prevents replay



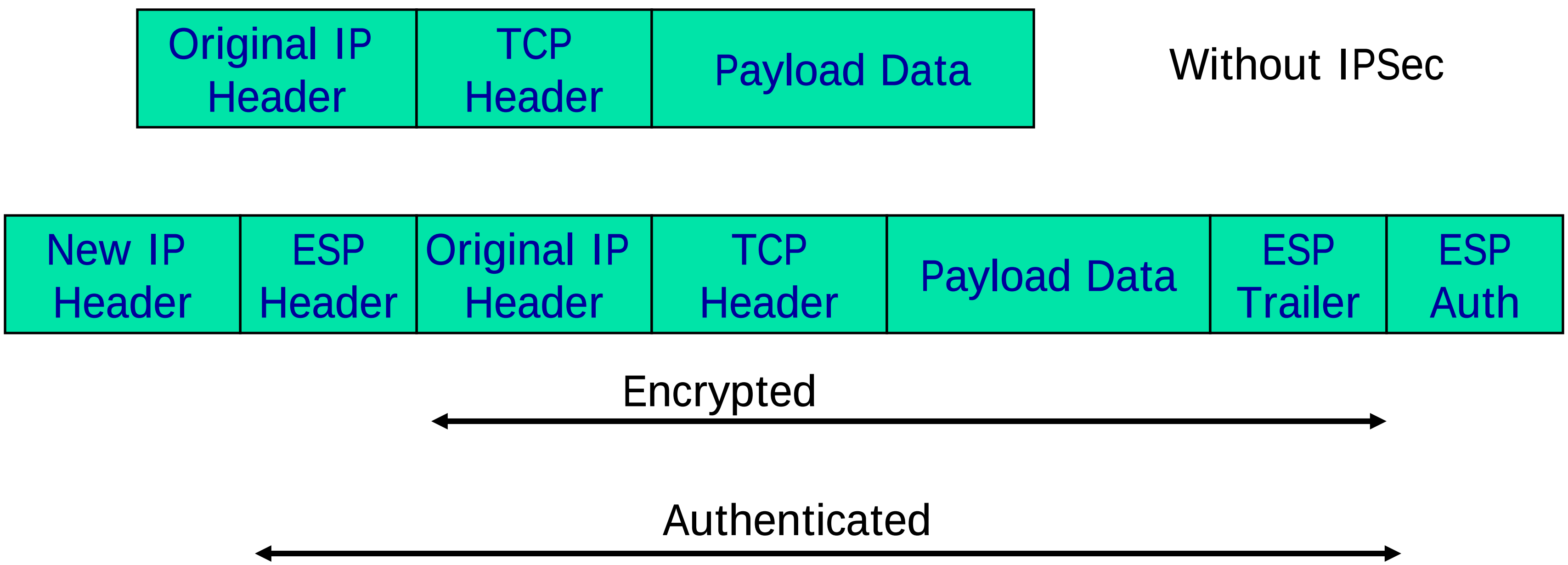
Details of ESP

- Security Parameters Index (SPI)
  - Specifies to the receiver the algorithms, type of keys, and lifetime of the keys used
- Sequence number
  - Counter that increases with each IP packet sent from the same host to the same destination and SA
- Payload
  - Application data carried in the TCP segment
- Padding
  - 0 to 255 bytes of data to enable encryption algorithms to operate properly
  - To mislead sniffers from estimating the amount of data transmitted
- Authentication Data
  - MAC created over the packet

Transport mode ESP



Tunnel mode ESP



Perimeter Defense

- Organization system consists of a network of many host machines –
  - the system is as secure as the weakest link
- Use perimeter defense
  - Define a border and use gatekeeper (firewall)
- If host machines are scattered and need to use public network, use encryption
  - Virtual Private Networks (VPNs)

Perimeter Defense

- Is it adequate?
  - Locating and securing all perimeter points is quite difficult
    - Less effective for large border
  - Inspecting/ensuring that remote connections are adequately protected is difficult
  - Insiders attack is often the most damaging

Firewalls

- Total isolation of networked systems is undesirable
  - Use firewalls to achieve selective border control
- Firewall
  - Is a configuration of machines and software
  - Limits network access
  - Come “for free” inside many devices: routers, modems, wireless base stations etc.
  - Alternate:

    a firewall is a host that mediates access to a network, allowing and disallowing certain type of access based on a configured security policy

What Firewalls can’t do

- They are not a panacea
  - Only adds to defense in depth
- If not managed properly
  - Can provide false sense of security
- Cannot prevent insider attack
- Firewalls act at a particular layer(s)

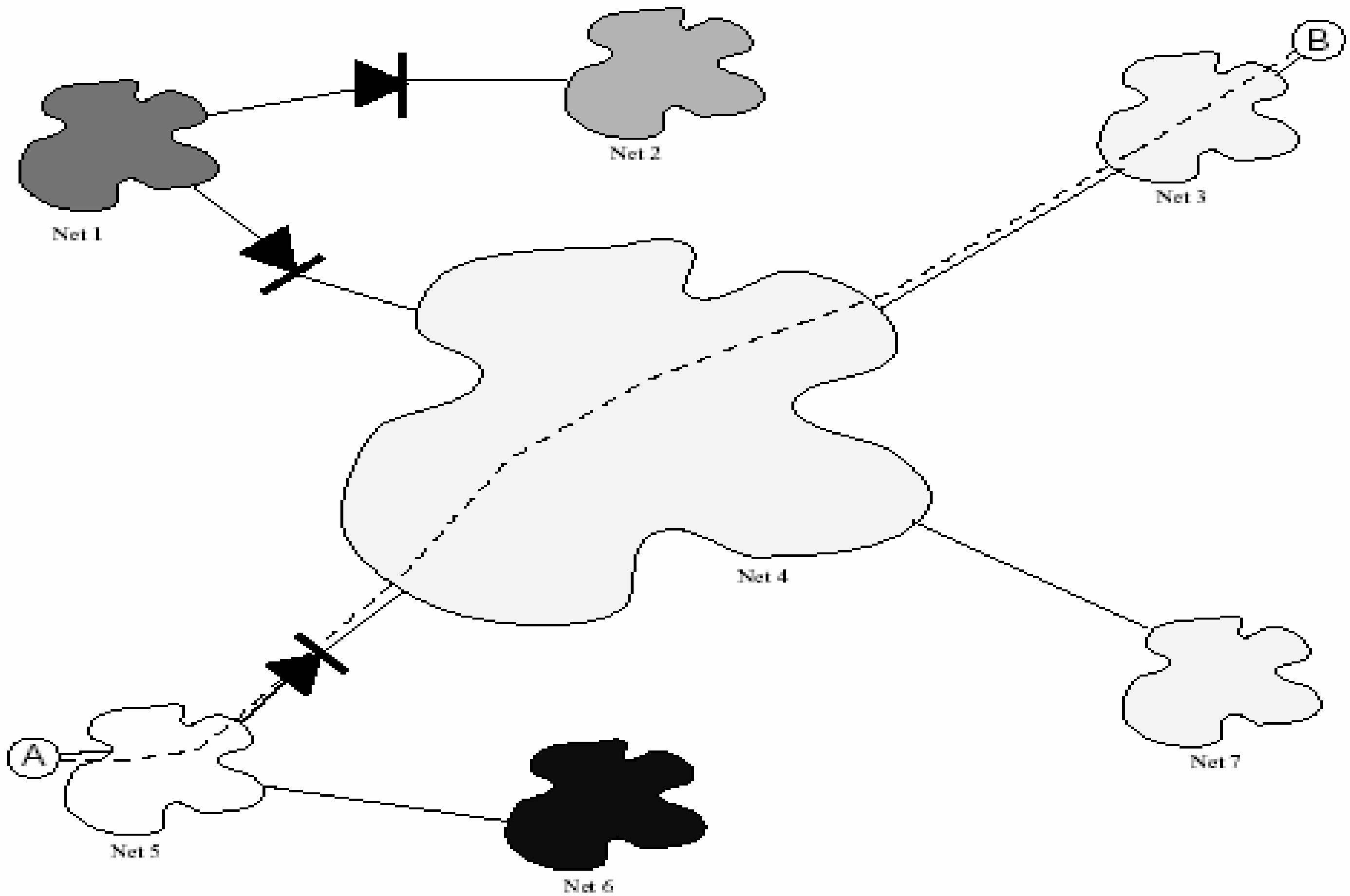
Virtual Private Networks What is it?

- It is a private network that is configured within a public network
- A VPN “appears” to be a private national or international network to a customer
- The customer is actually “sharing” trunks and other physical infrastructure with other customers
- Security?

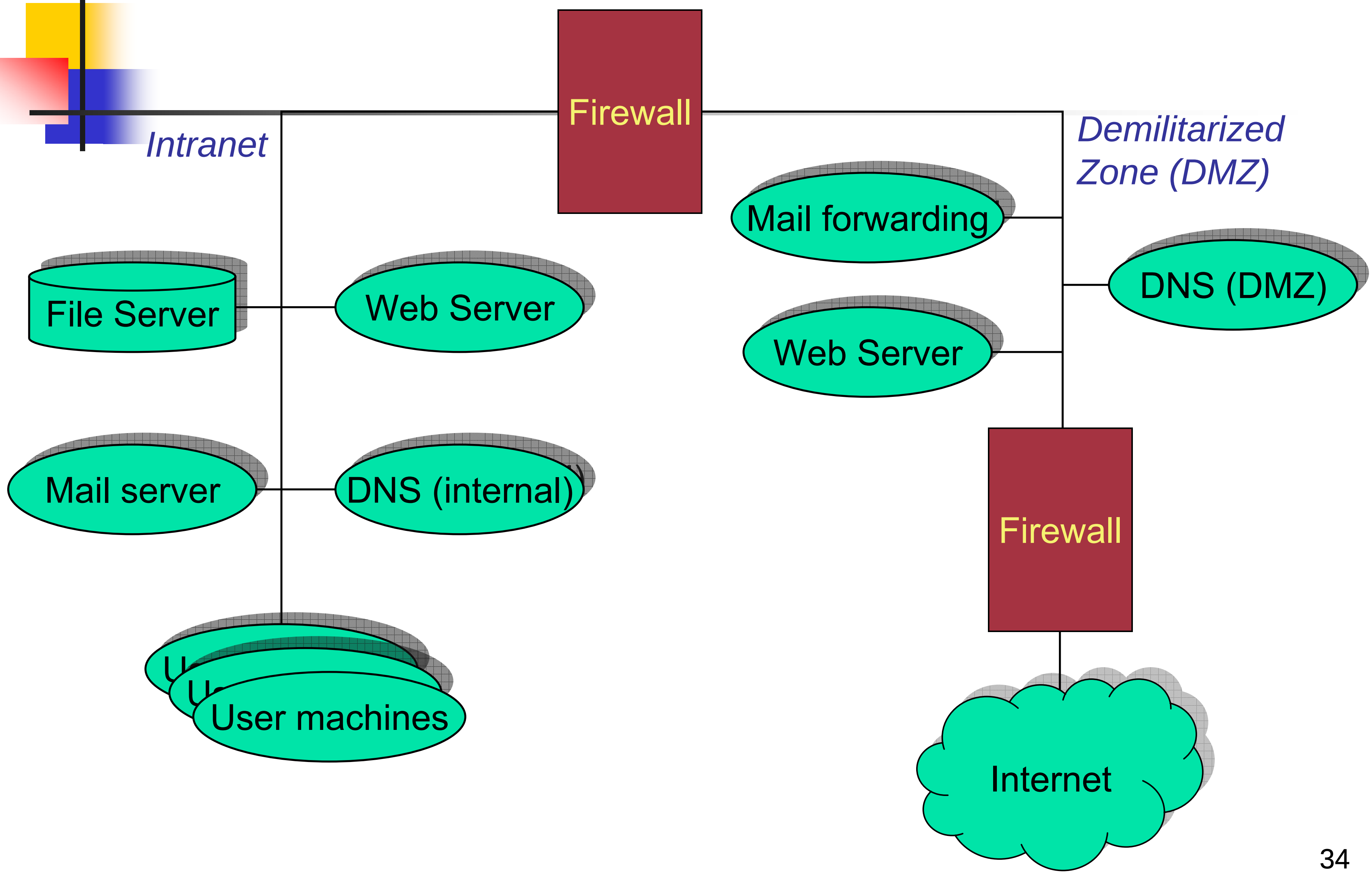
What is a VPN? (2)

- A network that supports a *closed* community of authorized users
- There is traffic isolation
  - Contents are secure
  - Services and resources are secure
- Use the public Internet as part of the virtual private network
- Provide security!
  - Confidentiality and integrity of data
  - User authentication
  - Network access control
- IPSec can be used

Tunneling in VPN



"Typical" corporate network



Authentication and Identity

What is Authentication?

- Authentication:
  - Binding identity and external entity to subject
- How do we do it?
  - Entity *knows* something (secret)
    - Passwords, id numbers
  - Entity *has* something
    - Badge, smart card
  - Entity *is* something
    - Biometrics: fingerprints or retinal characteristics
  - Entity is in *someplace*
    - Source IP, restricted area terminal

Authentication System: Definition

- A: Set of *authentication information*
  - used by entities to prove their identities (e.g., password)
- C: Set of *complementary information*
  - used by system to validate authentication information (e.g., hash of a password or the password itself)
- F: Set of *complementation functions* (to generate C)
  - $f : A \rightarrow C$
  - Generate appropriate $c \in C$ given $a \in A$
- L: set of *authentication functions*
  - l: $A \times C \rightarrow$ { **true, false** }
  - verify identity
- S: set of *selection functions*
  - Generate/alter A and C
  - e.g., commands to change password

Authentication System: Passwords

- Example: plaintext passwords
  - $A = C = \text{alphabet}^*$
  - f returns argument: $f(a)$ returns a
  - l is string equivalence: $l(a, b)$ is true if $a = b$

- Complementation Function
  - Null (return the argument as above)
    - requires that c be protected; i.e. password file needs to be protected
  - One-way hash – function such that
    - *Complementary information* $c = f(a)$ easy to compute
    - $f^{-1}(c)$ difficult to compute

Passwords

- Example: Original Unix
  - A password is up to eight characters each character could be one of 127 possible characters;
  - *A* contains approx. 6.9×10^{16} passwords
  - Password is hashed using one of 4096 functions into a 11 character string
  - 2 characters pre-pended to indicate the hash function used
  - *C* contains passwords of size 13 characters, each character from an alphabet of 64 characters
    - Approximately 3.0×10^{23} strings
  - Stored in file */etc/passwd* (all can read)

Authentication System

- Goal: identify the entities correctly
- Approaches to protecting
  - Hide enough information so that one of *a, c* or *f* cannot be found
    - Make C readable only to root
    - Make F unknown
  - Prevent access to the authentication functions *L*
    - *root* cannot log in over the network

Attacks on Passwords

- Dictionary attack: Trial and error guessing
  - Type 1: attacker knows A, f, c
    - Guess g and compute $f(g)$ for each f in F
  - Type 2: attacker knows A, l
    - l returns **True** for guess g

- Counter: Difficulty based on $|A|$, Time
    - Probability P of breaking in time T
    - G be the number of guesses that can be tested in one time unit
    - $|A| \geq TG/P$
    - Assumptions:
      - time constant; all passwords are equally likely

Password Selection

- Random
  - Depends on the quality of random number generator; size of legal passwords
  - 8 characters: humans can remember only one
    - Will need to write somewhere
- Pronounceable nonsense
  - Based on unit of sound (phoneme)
    - “Helgoret” vs “pxnftr”
  - Easier to remember
- User selection (proactive selection)
  - Controls on allowable
  - Reasonably good:
    - At least 1 digit, 1 letter, 1 punctuation, 1 control character
    - Obscure poem verse

Password Selection

- Reusable Passwords susceptible to dictionary attack (type 1)
  - *Salting* can be used to increase effort needed
    - makes the choice of complementation function a function of randomly selected data
    - Random data is different for different user
    - Authentication function is chosen on the basis of the salt
    - Many Unix systems:
      - A salt is randomly chosen from 0..4095
      - Complementation function depends on the salt

Password Selection

- Password aging
  - Change password after some time: based on expected time to guess a password
  - Disallow change to previous *n* passwords
- Fundamental problem is *reusability*
  - Replay attack is easy
  - Solution:
    - Authenticate in such a way that the transmitted password changes each time

Authentication Systems: Challenge-Response

- Pass algorithm
  - authenticator sends message *m*
  - subject responds with *f*(*m*)
    - *f* is a secret encryption function
    - In practice: key known only to subject
  - Example: ask for second input based on some algorithm

Authentication Systems: Challenge-Response

- One-time password: *invalidated after use*
  - f changes after use
    - Challenge is the number of authentication attempt
    - Response is the one-time password
- S/Key uses a hash function (MD4/MD5)
  - User chooses an initial seed k
  - Key generator calculates
    - $k_1 = h(k)$, $k_2 = h(k_1)$ …, $k_n = h(k_{n-1})$
  - Passwords used in the order
    - $p_1 = k_n$, $p_2 = k_{n-1}$, …, $p_n = k_1$
  - Suppose $p_1 = k_n$ is intercepted;
    - the next password is $p_2 = k_{n-1}$
    - Since $h(k_{n-1}) = k_n$, the attacker needs to invert h to determine the next password

Authentication Systems: Biometrics

- Used for human subject identification based on physical characteristics that are tough to copy
  - Fingerprint (optical scanning)
    - Camera's needed (bulky)
  - Voice
    - Speaker-verification (identity) or speaker-recognition (info content)
  - Iris/retina patterns (unique for each person)
    - Laser beaming is intrusive
  - Face recognition
    - Facial features can make this difficult
  - Keystroke interval/timing/pressure

Attacks on Biometrics

- Fake biometrics
  - fingerprint “mask”
  - copy keystroke pattern
- Fake the interaction between device and system
  - Replay attack
  - Requires careful design of entire authentication system

Authentication Systems: Location

- Based on knowing physical location of subject
- Example: Secured area
  - Assumes separate authentication for subject to enter area
  - In practice: early implementation of challenge/response and biometrics
- What about generalizing this?
  - Assume subject allowed access from limited geographic area
    - I can work from (near) home
  - Issue GPS Smart-Card
  - Authentication tests if smart-card generated signature within spatio/temporal constraints
  - Key: authorized locations known/approved in advance