

**ENTS 689i**

**Lecture 10:  
Network Security Protocols**

**Part III: Network Security**

# Part II: Outline

---

- November 13 (Today)
  - IP Security (IPSec)
  - Secure Socket Layer (SSL)/Transport Layer Security (TLS)
  - Kerberos
  - DNS Sec
  - Network Anonymity
  - HW Questions

# Network Security Protocols

---

- Cryptographic protocols are the foundation for many security services/mechanisms
  - Network computing environment
- Services provided at different layers
  - Assumptions about the network (environment)
  - Needs of communication peers
- Protocols challenges
  - Design (standards)
  - Implementation
  - Environment/threat assumptions

# IP Security (IPSec) Overview

---

- IP Security Protocol Suite
  - Address security concerns in IP (protocols/mechanisms)
    - Eavesdropping, session hijacking, spoofing, etc
- Implemented at the IP layer
  - Transparent to applications and end users (transport layer)
  - Below transport layer (TCP, UDP) / ICMP
- Collection of protocols/mechanisms
  - Confidentiality
  - Data origin authentication
  - Message integrity
  - Access control
  - Replay detection

# IP Security Architecture

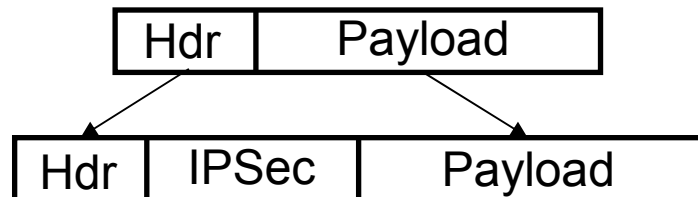
---

- Authentication Header (AH)
  - Provides support for data integrity and authentication
- Encapsulating Security Protocol (ESP)
  - Provides support for confidentiality (authentication optional)
- Security Associations (SA)
  - One way relationship between sender and receiver (sec parameters)
  - Traffic related to SAs (Security Policy Database (SPD))
- Transport mode
  - Protection for upper-layer protocols
- Tunnel mode
  - Protection for entire IP datagram

# Transport Mode

---

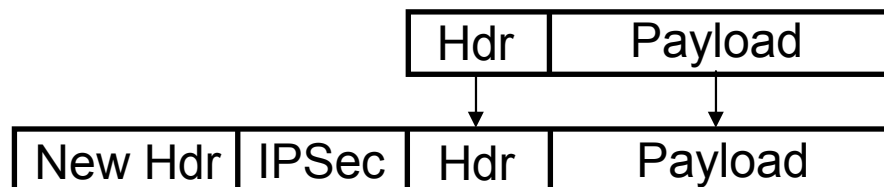
- Protects upper-layer protocols
  - Payload of IP datagram (data)
- IPSec information added between the IP header and the remainder of the packet
- End-to-end communication between hosts
  - Directly between a server and client



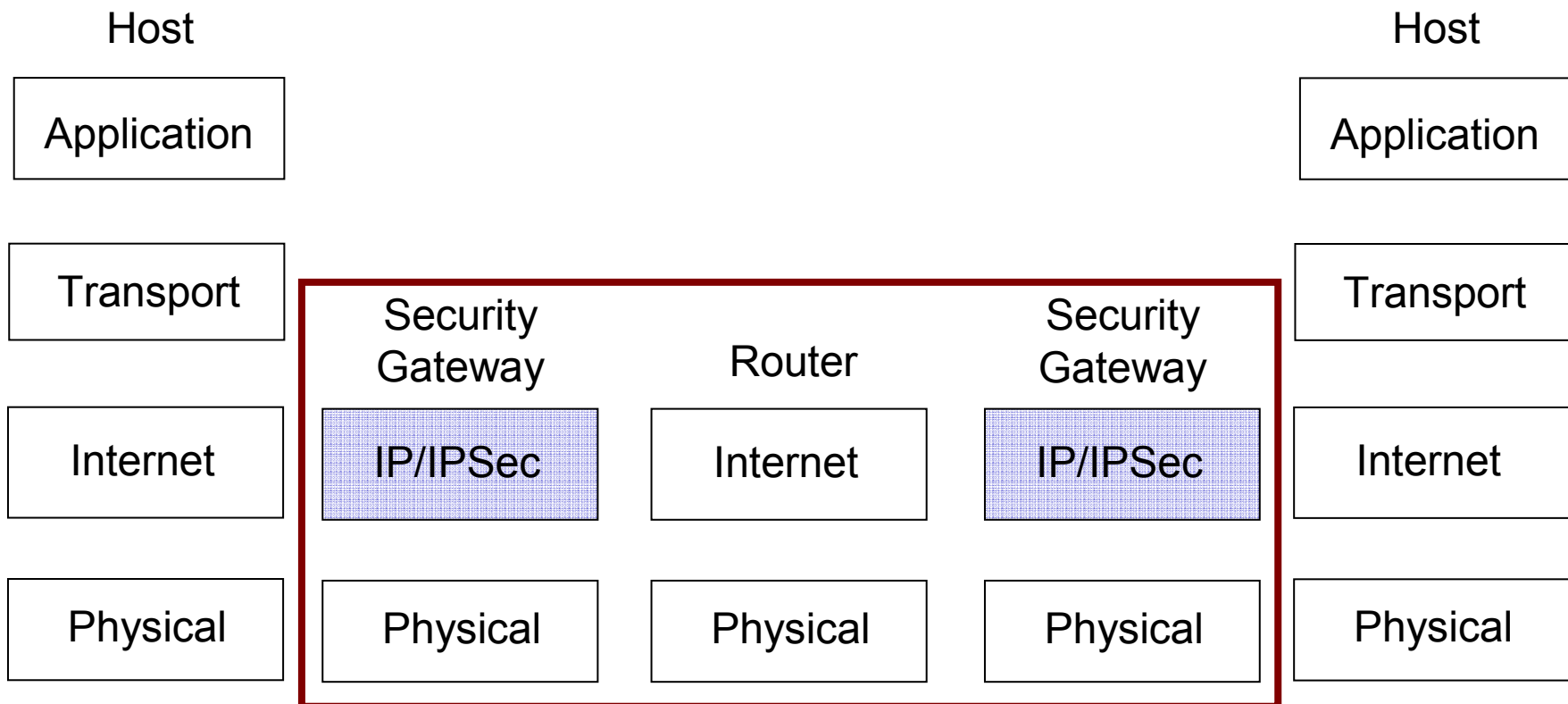
# Tunnel Mode

---

- Protection for entire IP datagram
  - IP datagram (header + data)
- Adds new “outer” IP header
  - Original IP datagram intact
- Between intermediate nodes (security gateway)
  - Firewall to firewall
  - VPN



# Tunnel Mode



# Security Associations

---

- Unidirectional association between peers
  - Describes the security services applied to packets
  - Destination address, security protocol (ESP or AH), Security Parameter Index (SPI) (32-bit)
- Security Association Database (SAD)
- Security association parameters
  - Sequence number counter
  - Sequence counter overflow
  - Anti-replay window
  - AH Information
  - ESP Information
  - Association lifespan (time/bytes)
  - IPSec Protocol mode
  - Address of opposite end

# Security Policy Database

---

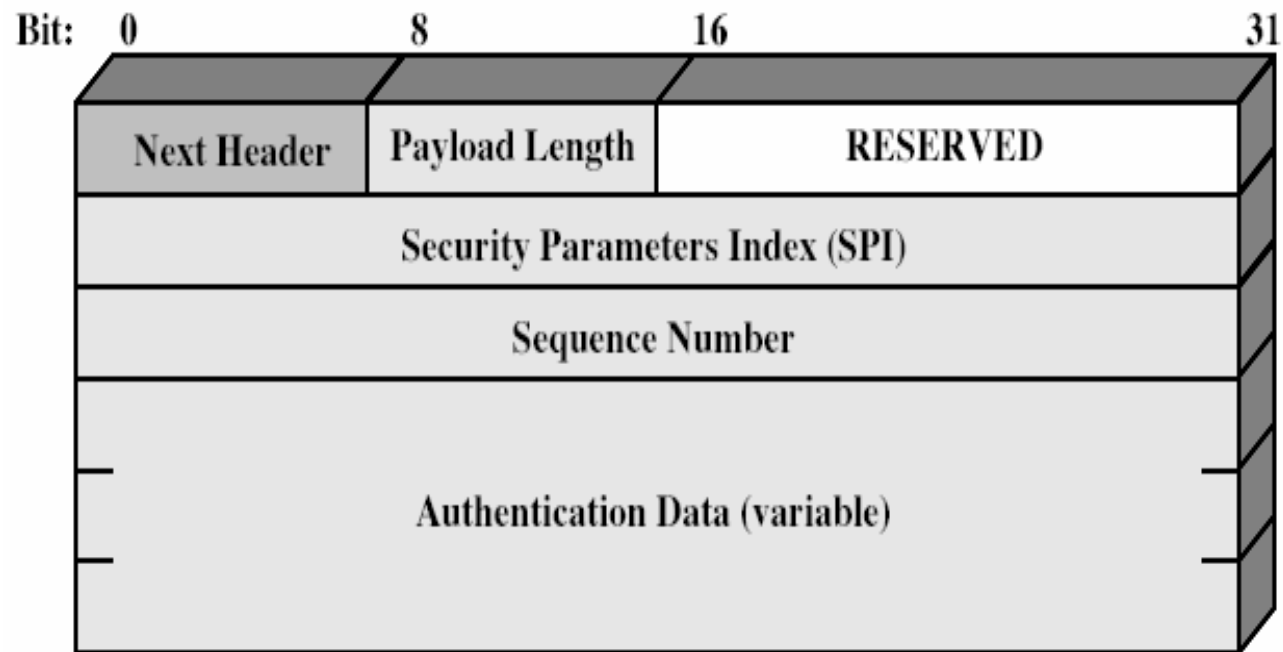
- Security Policy Database (SPD)
  - Maps IP traffic to SAs (think firewall rules)
  - Selectors
    - IP and upper layer protocol field values
    - Destination 10.1.2.3 to 10.1.2.103, port 25, apply IPSec
- Dictates how messages are handled
  - Discard
  - Security services (IPSec)
  - Forwarding
  - Bypass

# Authentication Header (AH)

---

- Security services
  - Message integrity
    - Static fields of IP header + IP Payload (Mutable fields?)
    - Prevents undetected modifications
  - Data origin authentication
    - Authenticate user or application
    - Prevents spoofing attacks
  - Anti-replay (optional)
- Leverages HMAC
  - Integrity Check Value (ICV)
  - MD5, SHA-1

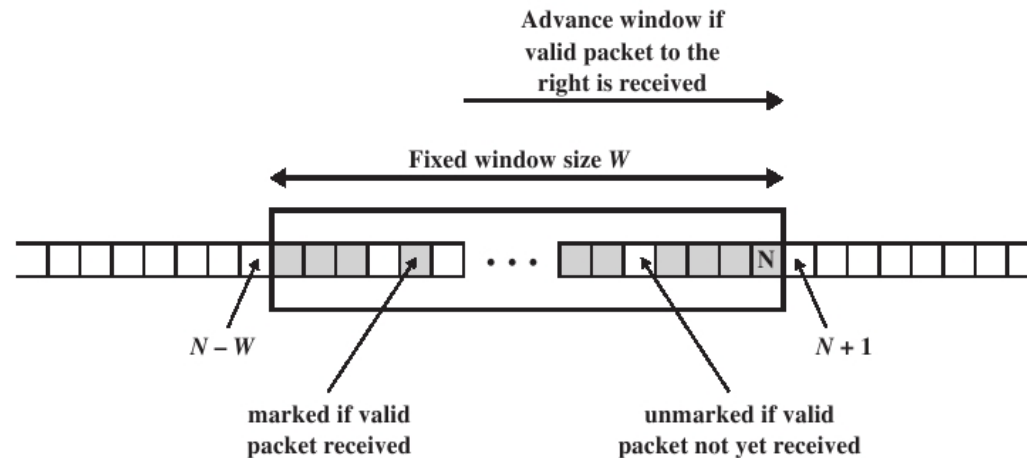
# Authentication Header (AH)



Stallings, William. Cryptography and Network Security. New Jersey: Pearson, 2003

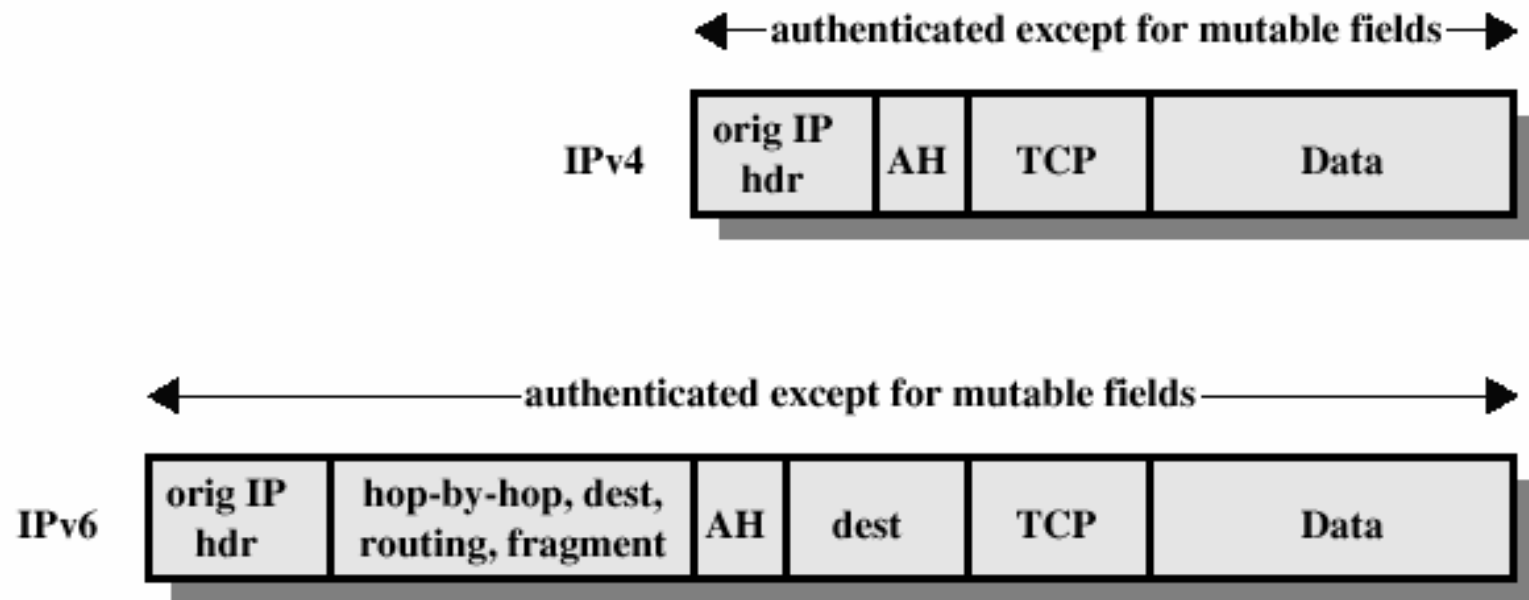
# AH: Anti-replay Service

- **Replay attack**
  - Attacker retransmits a copy of an authenticated packet
  - Disrupt service/undesired consequence
- **Sequence number**
  - Initialized to zero/Incremented for each packet sent on SA ( $2^{32}-1$ )



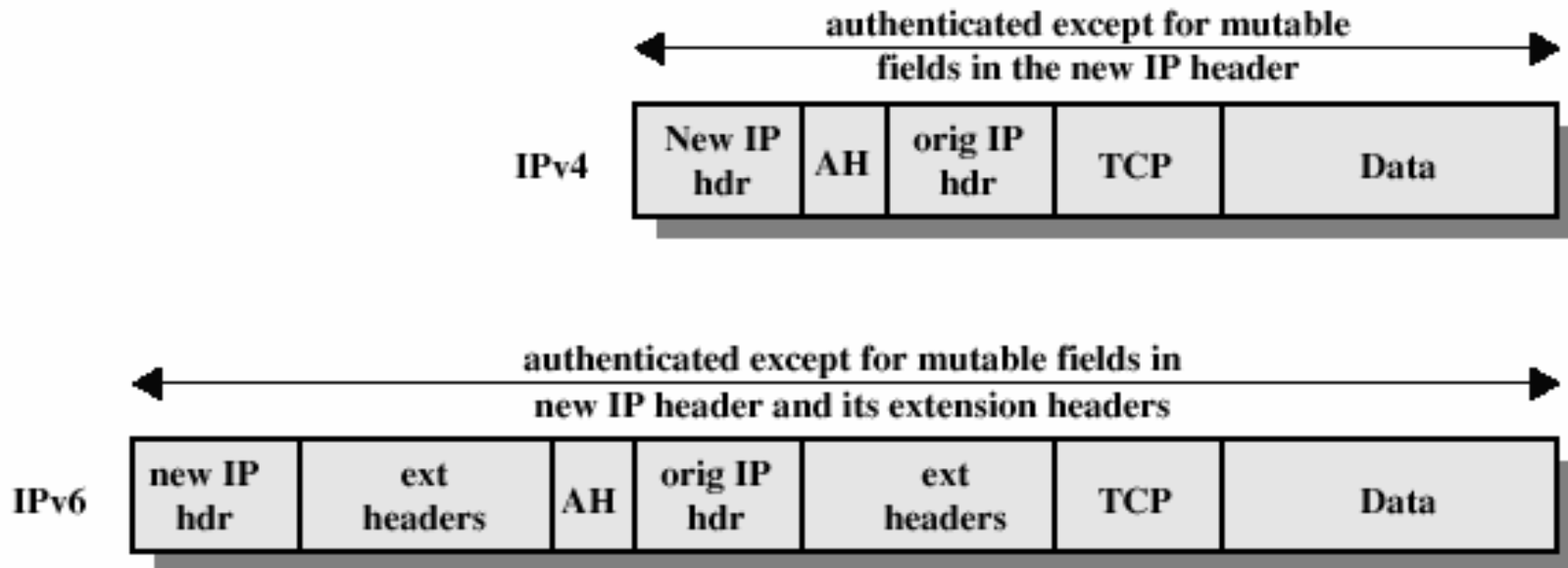
Stallings, William. Cryptography and Network Security. New Jersey: Pearson, 2003

# AH: Transport Mode



Stallings, William. *Cryptography and Network Security*. New Jersey: Pearson, 2003

# AH: Tunnel Mode



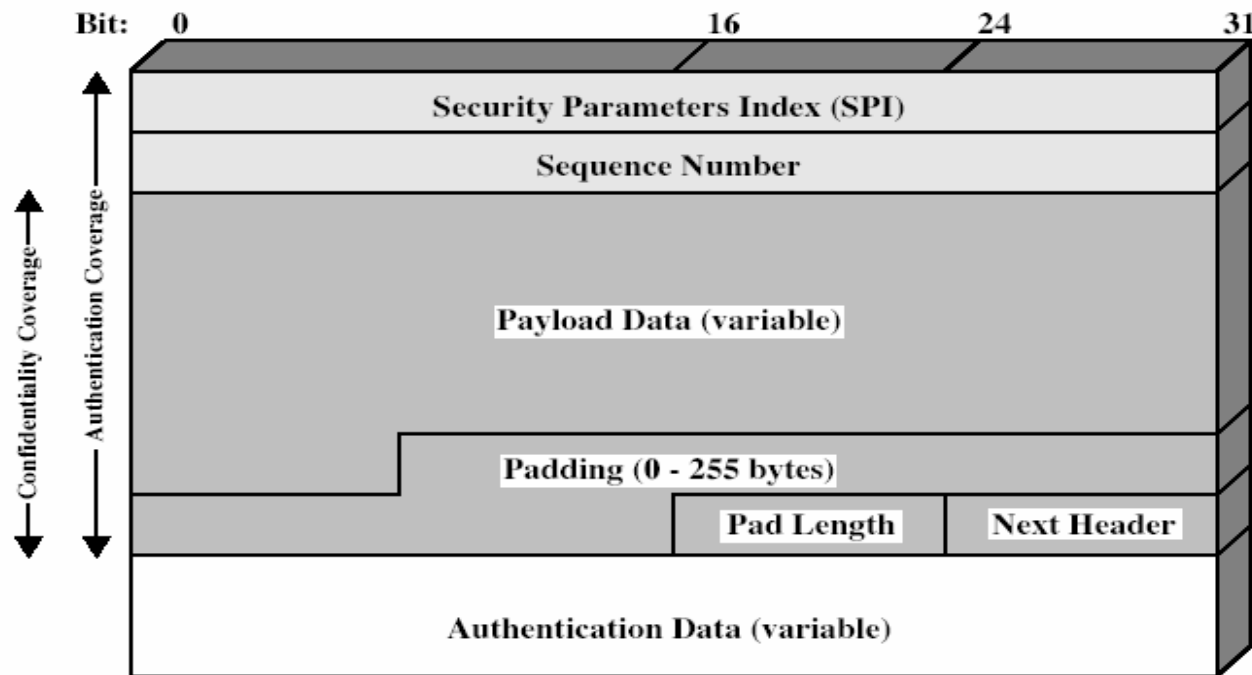
Stallings, William. Cryptography and Network Security. New Jersey: Pearson, 2003

# Encapsulating Security Payload

---

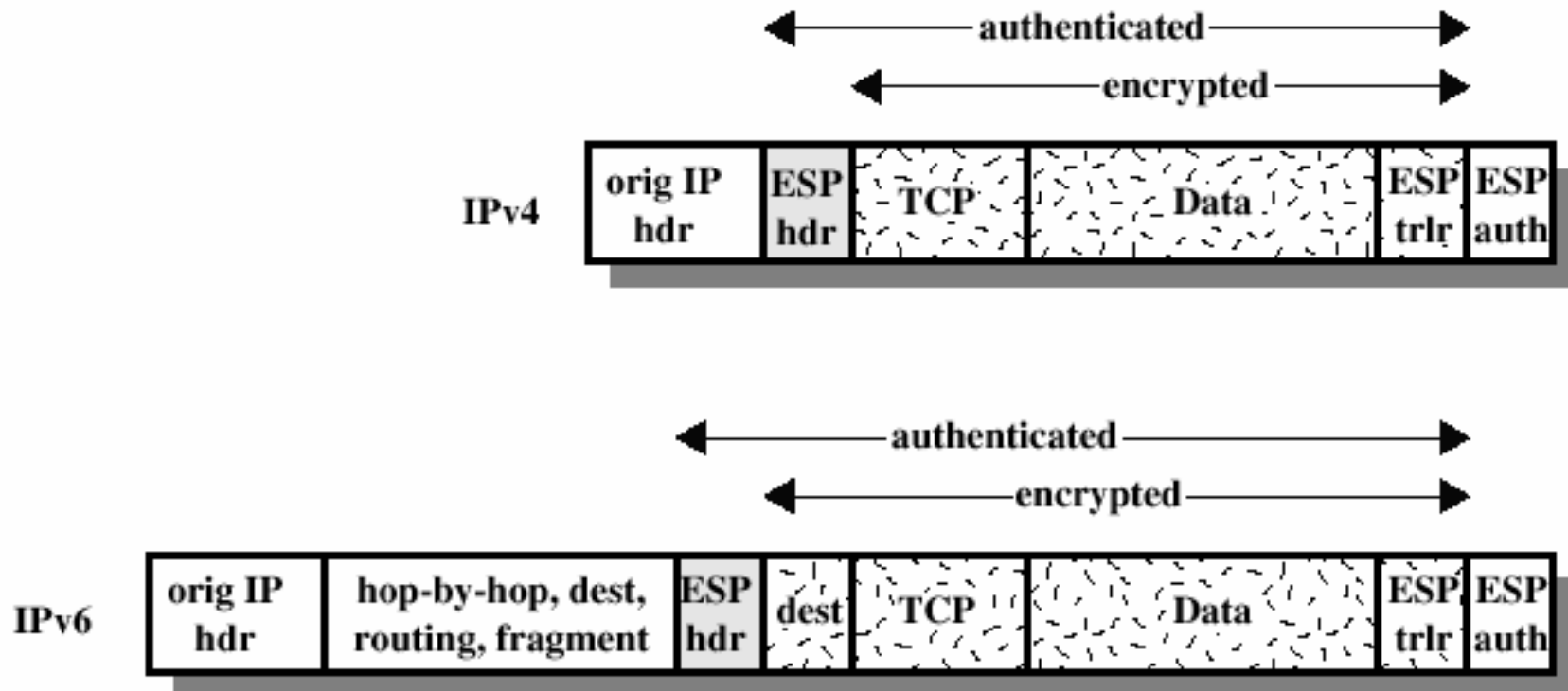
- Security Services:
  - Confidentiality
  - Traffic flow confidentiality
  - AH services
- Adds both a header and trailer
- Authentication does not include IP Header
  - Authentication happens after encryption
- Encryption and authentication algorithms
  - 3DES, RC5, IDEA,3IDEA, CAST, Blowfish
  - MD5, SHA-1

# Encapsulating Security Payload



Stallings, William. *Cryptography and Network Security*. New Jersey: Pearson, 2003

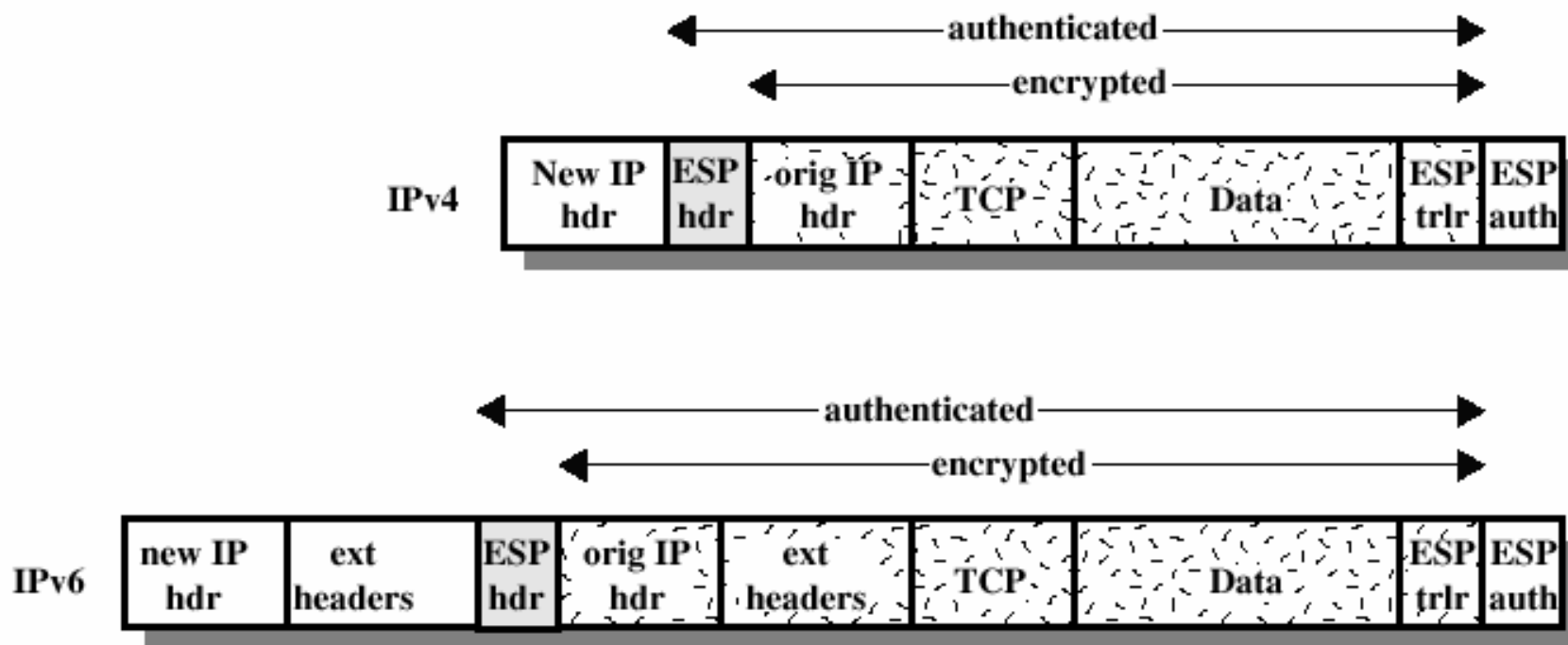
# ESP: Transport Mode



(a) Transport Mode

Stallings, William. *Cryptography and Network Security*. New Jersey: Pearson, 2003

# ESP: Tunnel Mode



(b) Tunnel Mode

Stallings, William. Cryptography and Network Security. New Jersey: Pearson, 2003

# Protocols and Modes

	Transport Mode	Tunnel Mode
AH	Authenticates IP datagram payload and immutable fields of IP header	Authenticates entire inner IP datagram plus immutable fields of outer IP header
ESP	Encrypts IP datagram payload	Encrypts inner IP packet
ESP with authentication	Encrypts IP datagram payload. Authenticates IP datagram payload <b>but not IP header.</b>	Encrypts inner IP packet. Authenticates inner IP packet.

# Key Management

---

- Generation and distribution of secret keys
  - Manual: system administrator
  - Automated: on-demand key generation
- Automated key management protocols
  - Internet Security Association and Key Management Protocol (ISAKMP)
    - Procedures (exchanges) and packet formats
      - Establish, negotiate, modify, delete security associations
  - Oakley Key Determination Protocol
    - Key exchange protocol (similar to Diffie-Hellman)
  - Internet Key Exchange
    - Oakley + SKEME + ISAKMP

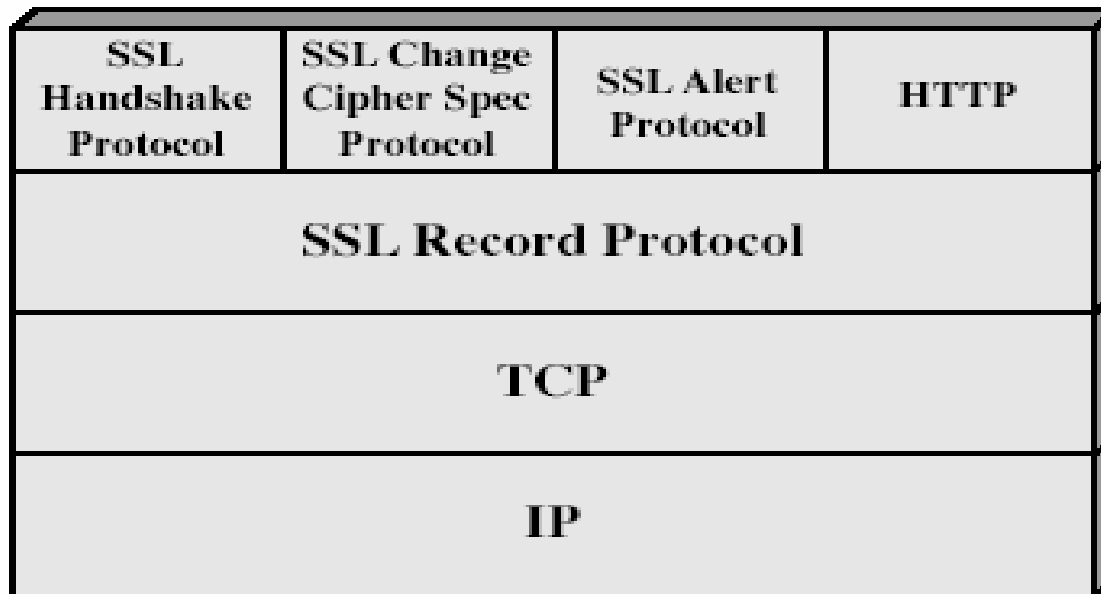
# SSL Overview

---

- Secure Socket Layer (SSL)
  - Developed by Netscape (www browsers/servers)
  - SSLv3
  - Negotiate mutually supported encryption suite
- Secure reliable end-to-end communication
  - Confidentiality, integrity, authentication
- Transport Layer Security (TLS) protocol
  - Internet standardization
  - Minor differences/backwards compatible
- Requires reliable end-to-end communication (TCP)
  - User-level not operating system changes

# SSL Architecture

---



Stallings, William. *Cryptography and Network Security*. New Jersey: Pearson, 2003

# Sessions and Connections

---

- **SSL session**
  - Ongoing association between peers (client/server)
  - Established by SSL Handshake Protocol
  - Specifies cryptographic parameters
    - Shared across multiple connections
    - Expensive public-key crypto
- **SSL connection**
  - Set of mechanisms used to transport data in an SSL session
    - Describes how data is sent/received between peers (transient)
    - Type of service
  - Every connection is associated with one SSL session

# SSL Session State

---

- **Session identifier**
  - Chosen by server to uniquely identify active or resumable session state
- **Peer certificate**
  - X509.v3 certificate of peer
- **Compression method**
  - Algorithm used to compress data before encryption
- **Cipher spec**
  - Specified encryption, hash algorithm, and crypto attributes
- **Master secret**
  - 48 byte secret shared between client and server
- **Is resumable**
  - Used to indicate if session can be used to initiate new connections

# SSL Connection State

---

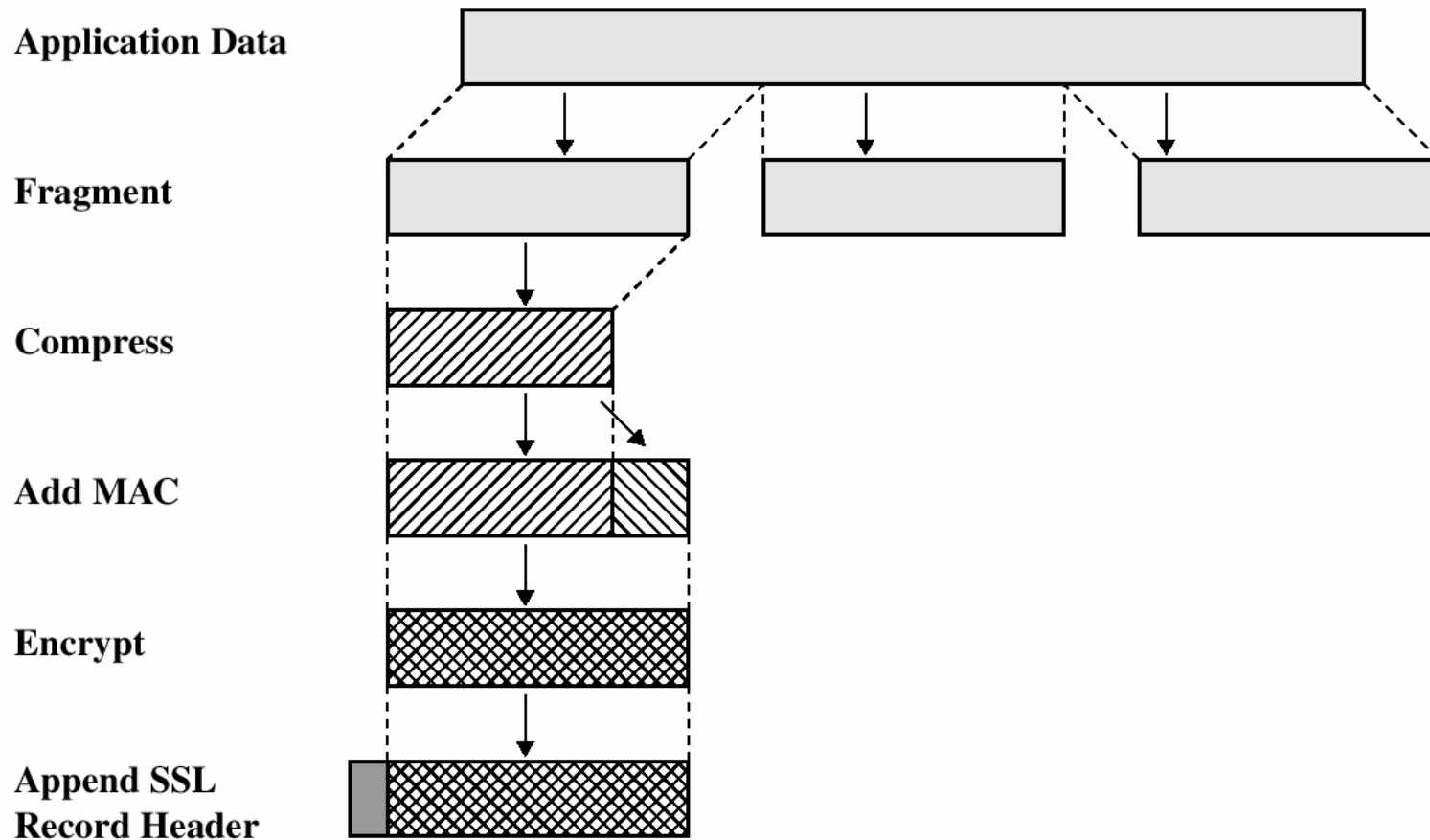
- **Server and client random**
  - Byte sequences chosen for each connection (32-bytes)
- **Server write MAC secret**
  - Shared key used to compute MAC on data sent by the server
- **Client write MAC secret**
  - Shared key used to compute MAC on data sent by the client
- **Server write key**
  - Shared key used to encrypt data sent by the server
- **Client write key**
  - Shared key used to encrypt data sent by the client
- **Initialization vectors**
  - Required when block cipher in CBC mode is used
- **Sequence numbers**
  - Client/server maintain separate sequence numbers for transmitted/received messages

# SSL Record Protocol

---

- Basic security services to higher-layer protocols (SSL connections)
  - Confidentiality
  - Message Integrity
- Application Message
  - Fragments: 16 Kb (or smaller)
  - Compressed (optional)
- Message Authentication Code (MAC)
  - Similar to HMAC
- Encrypted with symmetric encryption
- Protocol Header
  - Content Type, Major Version, Minor Version, Length
- Transmits in a TCP segment
  - Retransmission and reliable delivery

# SSL Record Protocol



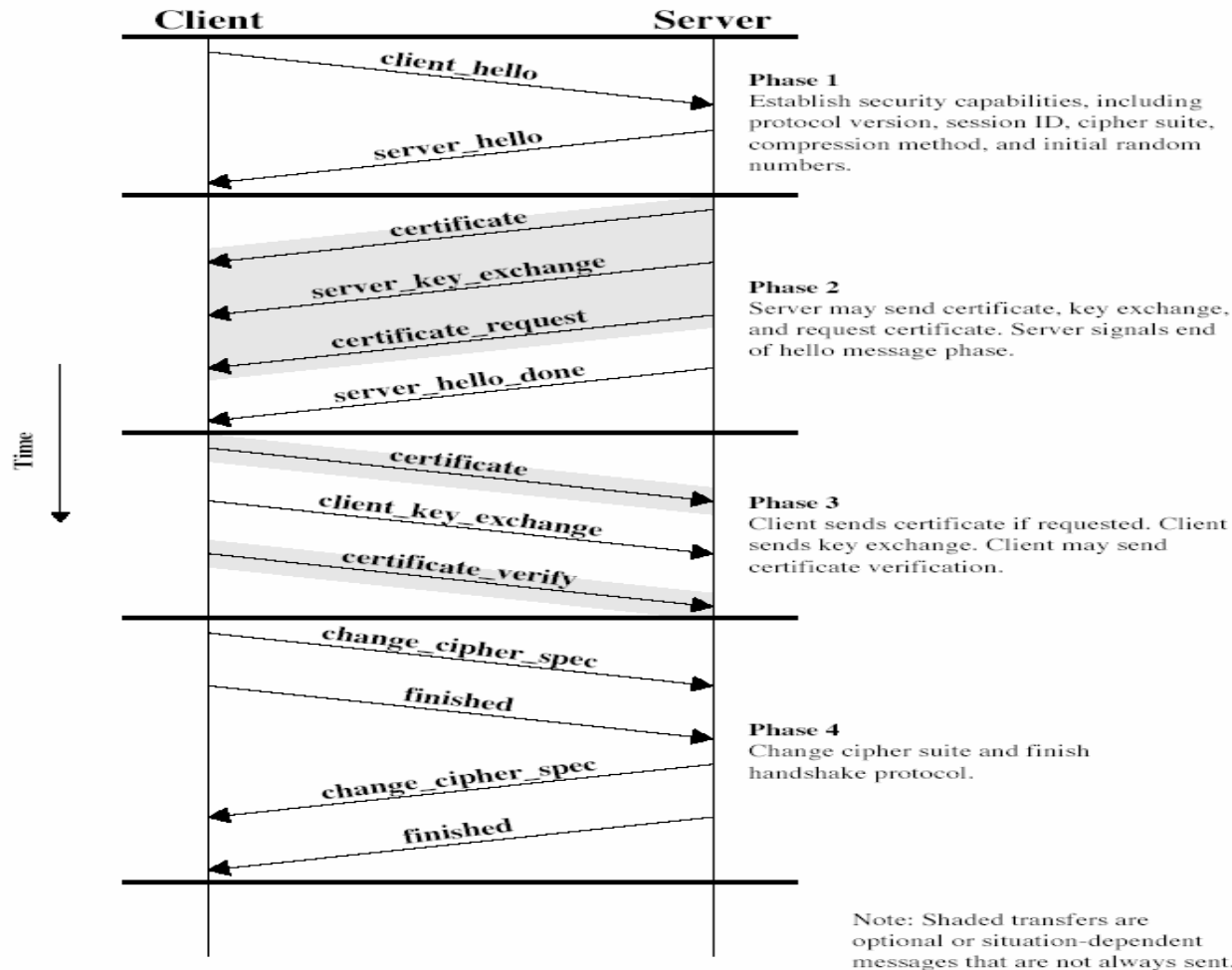
Stallings, William. Cryptography and Network Security. New Jersey: Pearson, 2003

# SSL Handshake Protocol

---

- Sets up parameters for the SSL record protocol
  - Cryptographic keys, symmetric ciphers, MAC algorithms
  - Authenticate server/client (optional)
  - Before any application data is transmitted
- Four phases
  - Establish security capabilities
  - Server authentication and key exchange
  - Client authentication and key exchange
  - Finish
- Messages depend on level of security

# SSL Handshake Protocol



Stallings, William. Cryptography and Network Security. New Jersey: Pearson, 2003

# Phase 1: Establish Capabilities

---

- Initiate a connection and establish security capabilities
- Client initiates (client\_hello)
  - Version: highest supported version of SSL
  - Random: nonce (32-bit ts /28 bytes random)
  - Session ID: session identifier
  - CipherSuite: client supported crypto-algorithms
  - Compression Method: supported compression
- Server responds (server\_hello)
  - Selects parameters

# Phase 2/3: Key Exchange/Authentication

---

- Key exchange methods
  - RSA
  - Fixed Diffie-Hellman
  - Ephemeral Diffie-Hellman
  - Anonymous Diffie-Hellman
- Verifies identity of peers
- Phase 2: Server Authentication/Key Exchange
  - Sends certificate/server\_key\_exchange (if required)
  - Server notifies client (server\_done)
- Phase 3: Client Authentication/Key Exchange
  - Verifies certificate
  - Completes key exchange (client\_key\_exchange)

# Phase 4: Finish

---

- Completes secure connection setup
- Notifies peer to use new cipher (change\_cipher\_spec)
  - Change Cipher Spec Protocol
- Verifies key exchange and authentication were successful (finished)
  - New keys, algorithms, parameters
- Handshake is complete!

# SSL Change Cipher Spec Protocol

---

- Simplest SSL protocol
  - Transferred using the SSL record protocol
  - Handshake Protocol
- Indicates subsequent records will use updated cipher suite (transition)
  - Pending state becomes fixed
- Message is single byte
  - Carries the value 1
- Sent by both client and server
  - Session state is considered agreed

# SSL Alert Protocol

---

- Signals SSL-related alerts between peers
- Messages (2-bytes)
  - Severity Level (1-byte)
    - Fatal (2): Connection immediately terminated
    - Warning (1): Connection/security unstable
  - Alert code (1-byte)
- Closure alert (`close_notify`) (truncation attack)
- Error alerts
  - Fatal :
    - `unexpected_message`, `bad_record_mac`, `decompression_failure`, `handshake_failure`, `illegal_parameter`
  - Warning/Fatal:
    - `no_certificate`, `bad_certificate`, `unsupported_certificate`, `certificate_revoked`, `certificate_expired`, `certificate_unknown`

# Kerberos

---

- Kerberos
  - Developed at MIT
  - Version 4/ Version 5
  - A centralized network protocol that provides distributed authentication
  - Trusted third-party authentication service
- Challenge
  - A network environment where users at workstations need to access restricted distributed services (files, printers, etc)
- Services (symmetric key cryptography)
  - Authentication
  - Confidentiality
  - Data integrity
  - Authorization/Access control

# Threats

---

- Unauthorized user gaining access to services/data
- Impersonation
  - Pretend to be another user
  - Address spoofing
- Eavesdropping
  - Replay attack

# Entities

---

- Client (C)
  - Access services on servers throughout network
  - Logs in to workstation (userid/password)
- Kerberos
  - Authentication Server (AS)
    - Responsible for authenticating users (all passwords)
    - Shares a unique key with each server
    - Issues a ticket-granting ticket (TGT)
  - Ticket-granting Server (TGS)
    - Verifies users authenticated by AS
    - Grants users service-granting tickets for particular services
- Application Server (V)
  - Authenticates user using service-granting ticket
  - Provides a service (files, printers, emails, etc)

# Kerberos Realms

---

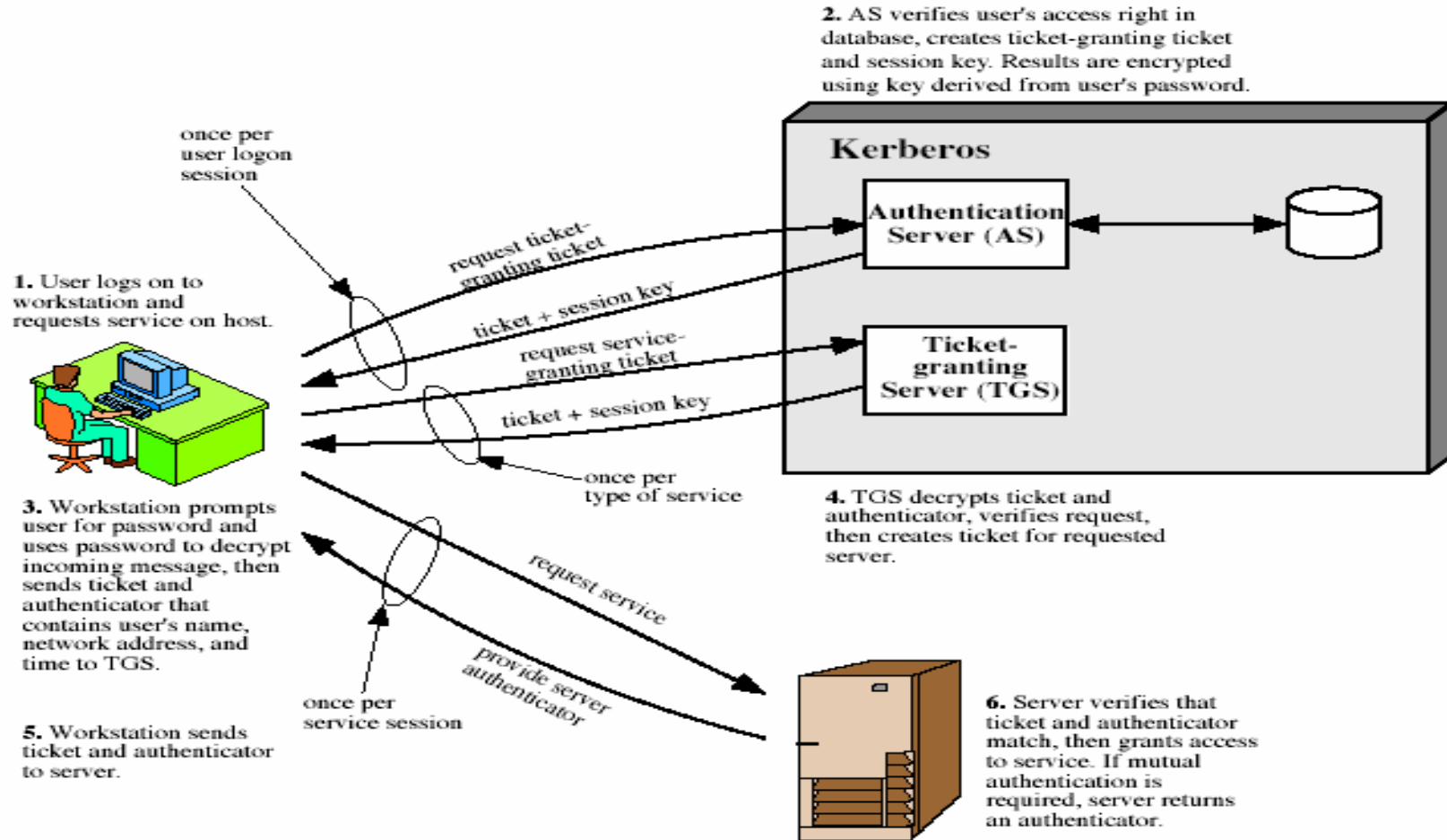
- Networks of servers and clients under control of Authentication Server (administrative domain)
- Kerberos environment
  - Kerberos server
  - Clients
    - UID and hashed passwords registered
  - Application servers
    - Secret key registered
- Inter-realm authentication
  - Secret key shared between realms

# Tickets

---

- Ticket
  - Grants a particular authenticated client (user) the authorization to obtain a service from the specified server
- Types of Tickets
  - Ticket-granting ticket
    - Issued by Authentication Sever (AS)
    - Authenticates client to Ticket Granting Server (TGS)
    - Authorizes ability to ask for a service
    - Lifetime: Once per user logon session
  - Service-granting ticket
    - Issued by Ticket Granting Server (TGS)
    - Authenticates client to server
    - Authorizes ability to use that service
    - Lifetime: Once per type of service

# Overview of Kerberos



Stallings, William. *Cryptography and Network Security*. New Jersey: Pearson, 2003

# Kerberos Version 4 Messages

## Authentication Service Exchange, To obtain Ticket-Granting Ticket

(1) **C → AS:**  $ID_c \parallel ID_{tgs} \parallel TS_1$

(2) **AS → C:**  $E_{K_c} [K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}]$

$Ticket_{tgs} = E_{K_{tgs}} [K_{c,tgs} \parallel ID_c \parallel AD_c \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2]$

## Ticket-Granting Service Exchange: To obtain Service-Granting Ticket

(3) **C → TGS:**  $ID_v \parallel Ticket_{tgs} \parallel Authenticator_c$

(4) **TGS → C:**  $E_{K_{c,tgs}} [K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v]$

$Authenticator_c = E_{K_{c,tgs}} [ID_c \parallel AD_c \parallel TS_3]$

## Client/Server Authentication Exchange: To Obtain Service

(5) **C → V:**  $Ticket_v \parallel Authenticator_c$

(6) **V → C:**  $E_{K_{c,v}} [TS_5 + 1]$

$Ticket_v = E_{K_v} [K_{c,v} \parallel ID_c \parallel AD_c \parallel ID_v \parallel TS_4 \parallel Lifetime_4]$

$Authenticator_c = E_{K_{c,v}} [ID_c \parallel AD_c \parallel TS_5]$

# Kerberos Version 5

---

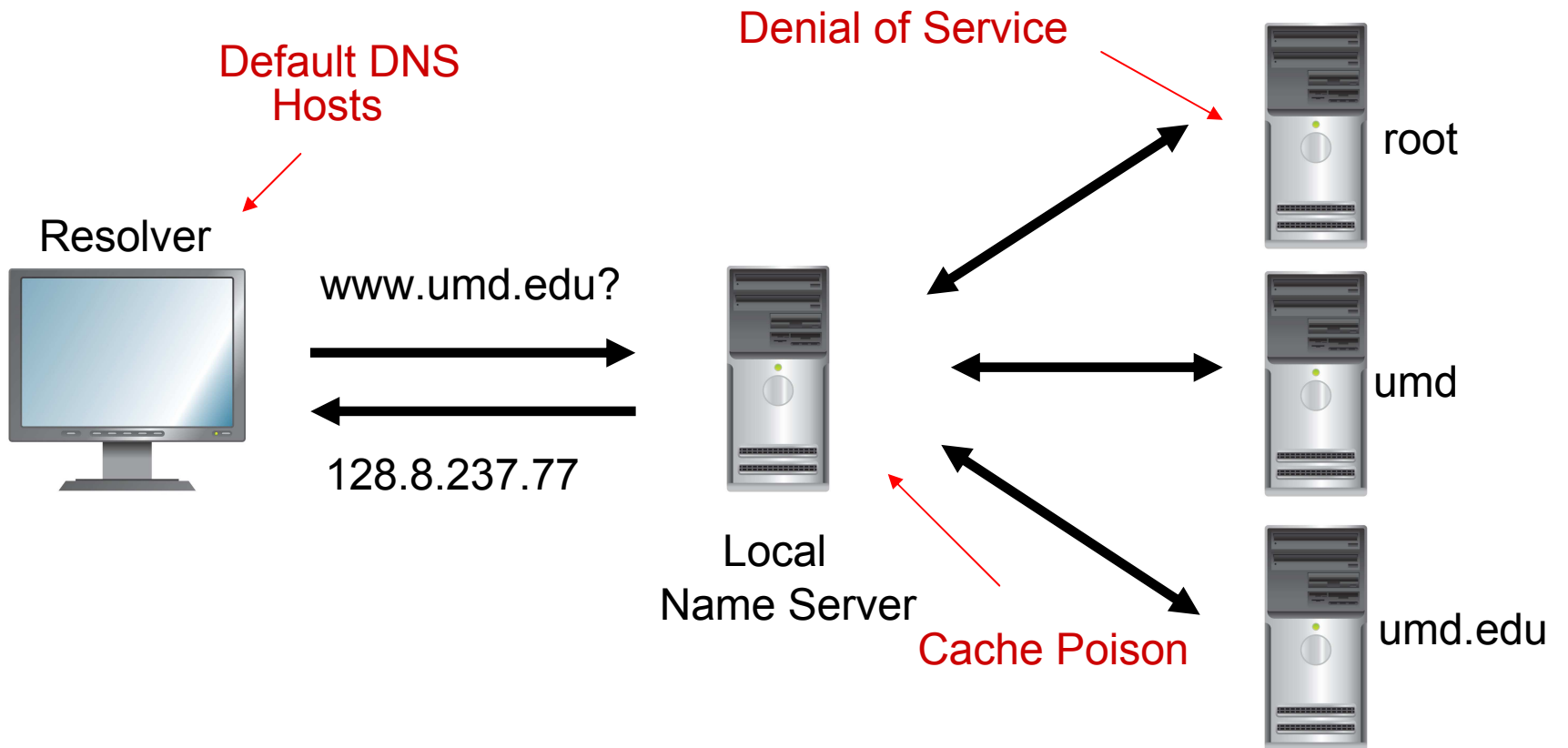
- Limitations of version 4
- Environmental shortcomings
  - Encryption algorithms, IP dependence, byte ordering, ticket lifetimes, authentication forwarding, inter-realm authentication
- Technical deficiencies
  - Unnecessary encryption, PCBC→CBC, sub-session key, password attacks

# Domain Name Service (DNS)

---

- Associates host names and IP addresses
  - Forward record: host name → IP address
  - Reverse record: IP address → host name
  - Data is distributed (hierarchical: root servers)
  - Caching
- Vulnerabilities:
  - Denial of service
  - Information disclosure (zone transfers)
  - DNS cache poisoning
  - DNS Spoofing (authoritative)
- Malware

# DNS Resolving



# DNSSec

---

- Domain Name System Security Extensions (DNSSEC)
  - Leverages public key cryptography (digital signatures)
  - Authenticate servers
  - DNS zone signs its data with private key (offline)
  - Public keys are published in DNS
- Provides to clients (resolvers)
  - Data origin authentication (authoritative source)
  - Data integrity (not been modified)
  - Authenticated denial of existence
- Digitally sign query answers
  - DNS resource record (A, MX, PTR)

# DNSSec Issues

---

- Public key storage?
- Key management
  - Updating keys
- Information leak
  - Zone enumeration (NSEC3)
- Secures traffic to local name server
- Failed validation?
- Does not address
  - Confidentiality
  - Availability

# Network Anonymity

---

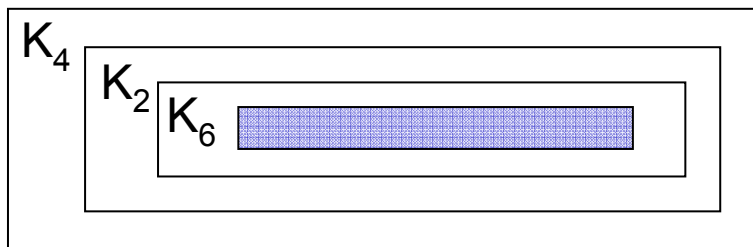
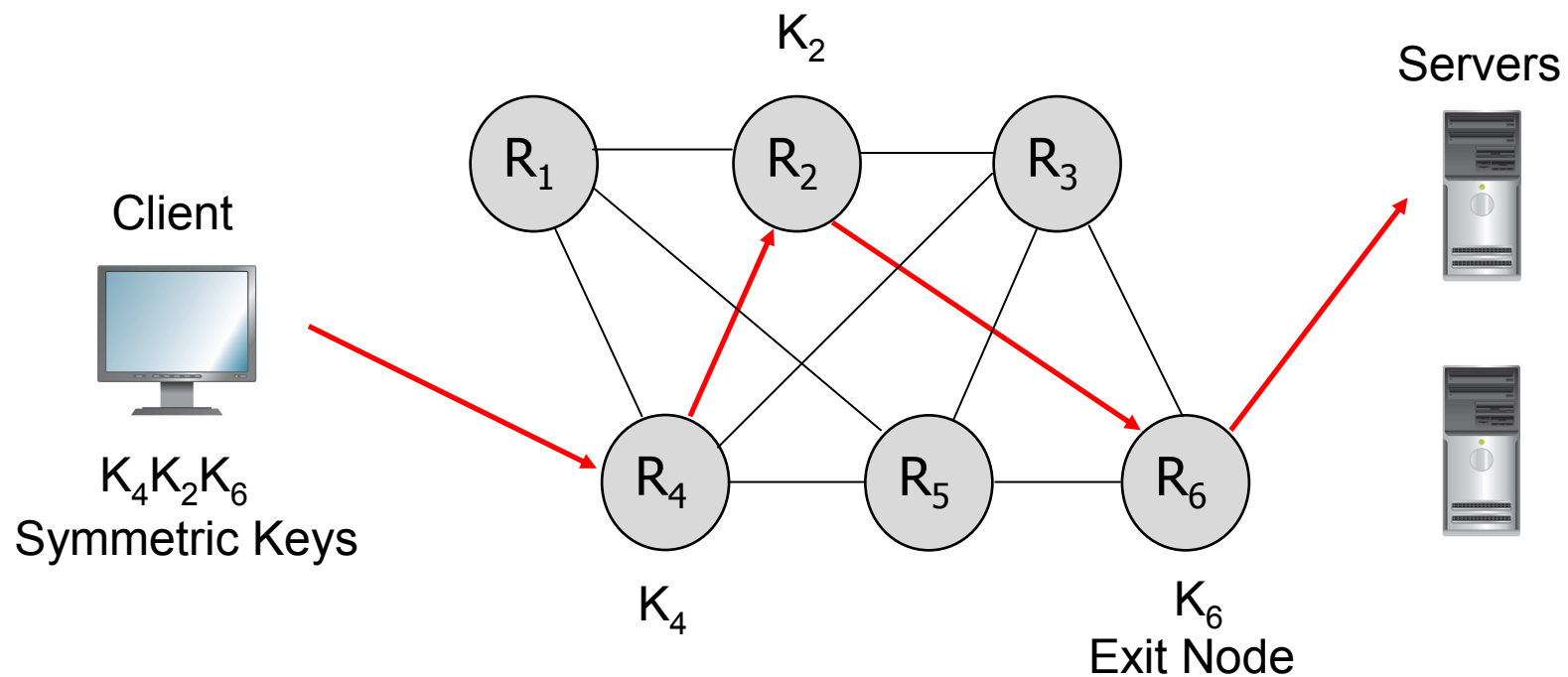
- Protect identity/data association (links)
  - Shield our identity within a group of entities
- Why would someone want anonymity?
  - Internet censorship
  - Freedom of speech (journalists, whistleblowers, political dissidents)
  - Protects privacy (medical, financial, etc)
    - ISPs, marketers, etc
- Negatives
  - Criminal attribution (attacks, contraband)
- Network Anonymity
  - Anonymous Mailers (remailers)
  - Anonymous Routing

# The Onion Router (TOR)

---

- The Onion Router (TOR)
  - Low latency anonymous Internet connections
  - <http://www.torproject.org/>
- Client (Onion Proxy)
  - Transport layer (TCP) (SOCKS)
  - Web browsing, IRC, instant messaging
- Onion routers
  - Distributed overlay network
  - Virtual circuit
- Directory servers
  - Sender chooses random sequence of routers
- Layered cryptography (fixed sized cells)

# The Onion Router (TOR)



# TOR Issues

---

- Out of band leaks
  - DNS
  - JavaScript/Flash errors (application layer)
- Traffic analysis
  - Global passive adversary
    - Enter/exit nodes
  - Delay watermarking (Timing)
  - Congestion (Volume)
- Exit node eavesdropping
  - Anonymity  $\neq$  security
  - Intercept usernames and passwords (2007)
- Routing Optimizations (adaptation)

# References

---

- **Cryptography and Network Security**
  - William Stallings
- **Computer Security: Art and Science**
  - Matt Bishop
- **Information Security**
  - Cristina Nita-Rotaru
- **Security in Computing**
  - Charles P. Pfleeger and Shari Lawrence Pfleeger
- **Network Security**
  - Charlie Kaufman, Radia Perlman, and Mike Speciner