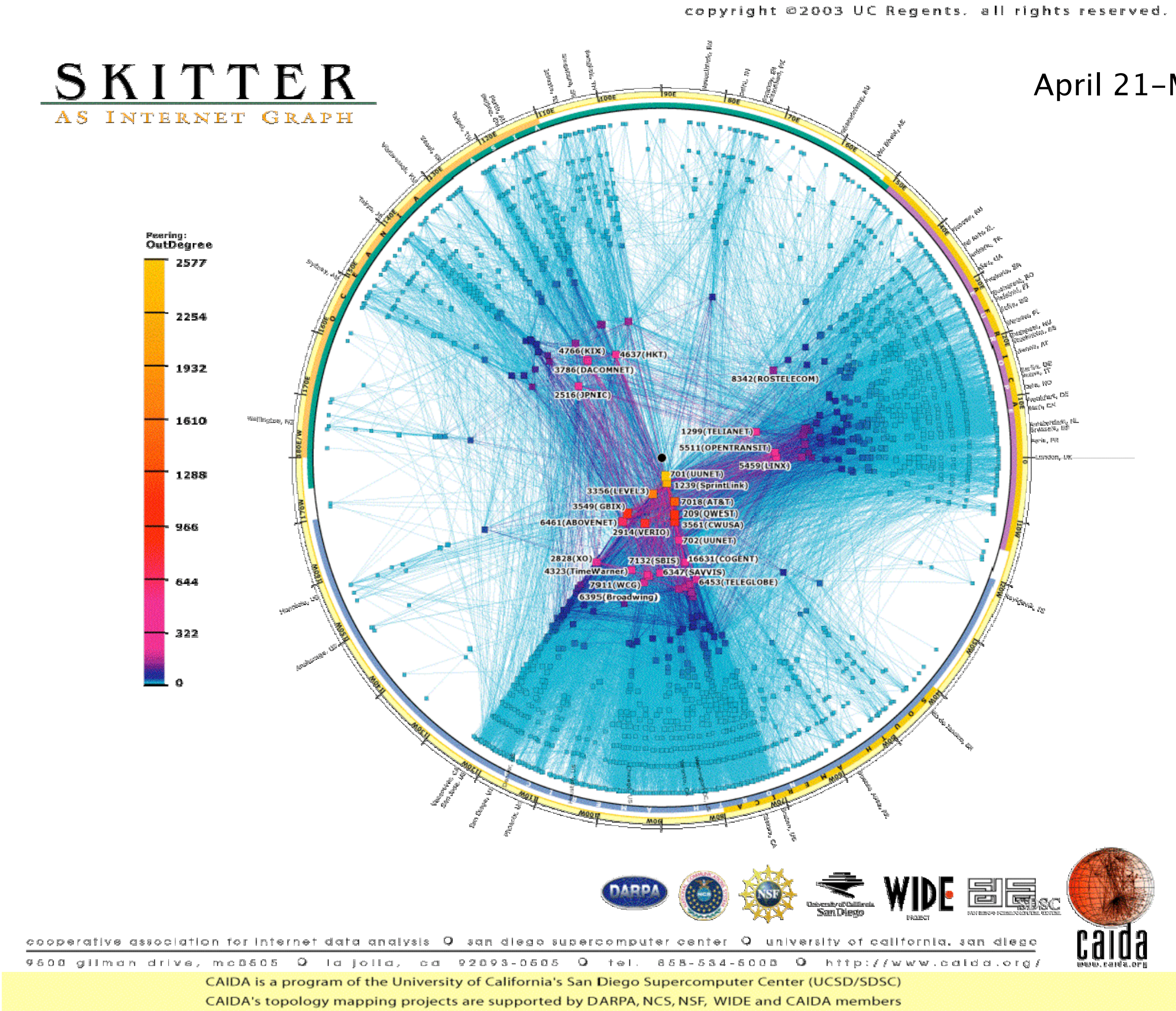
cs526: Information Security

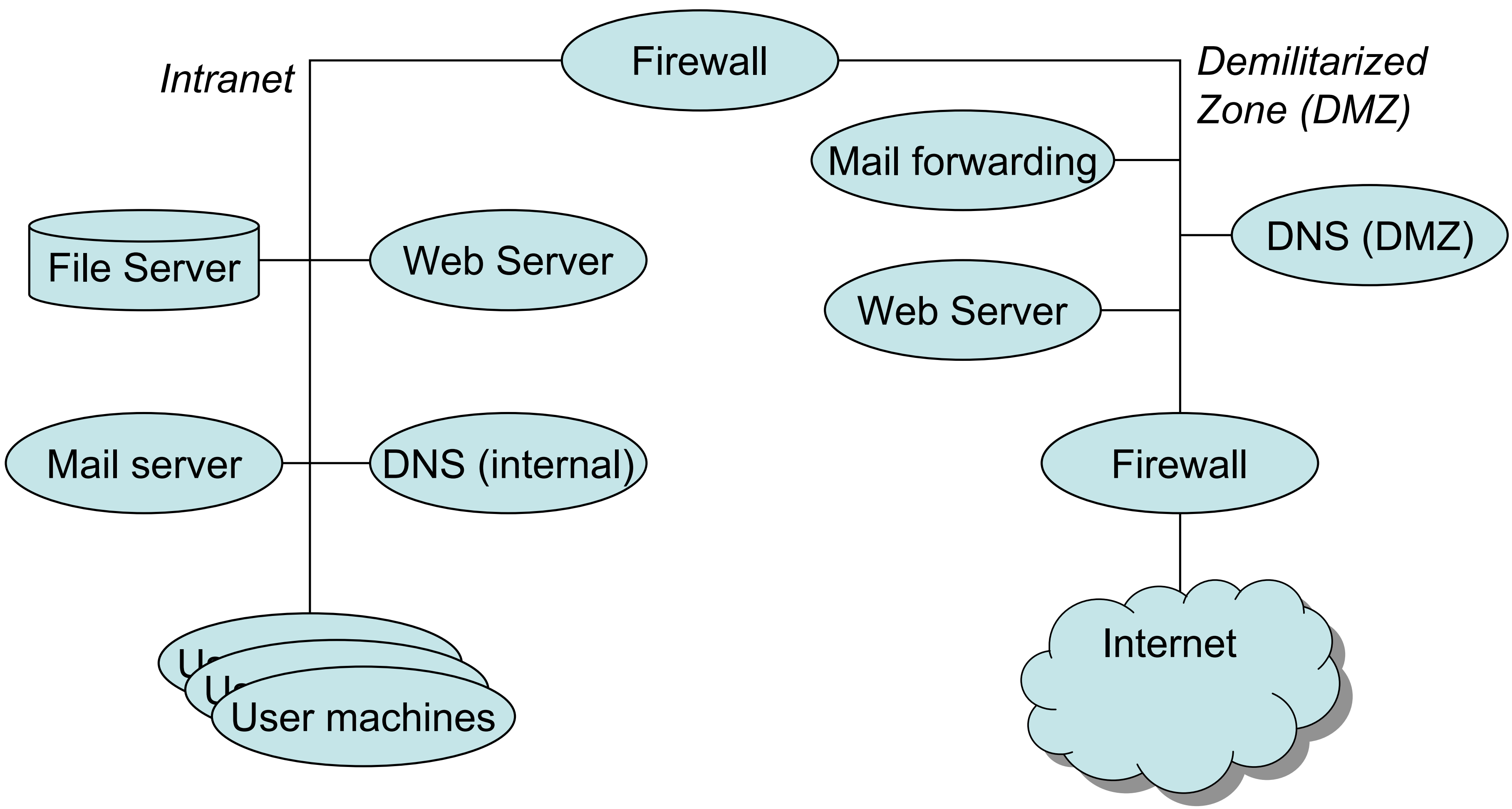
Cristina Nita-Rotaru

Internet

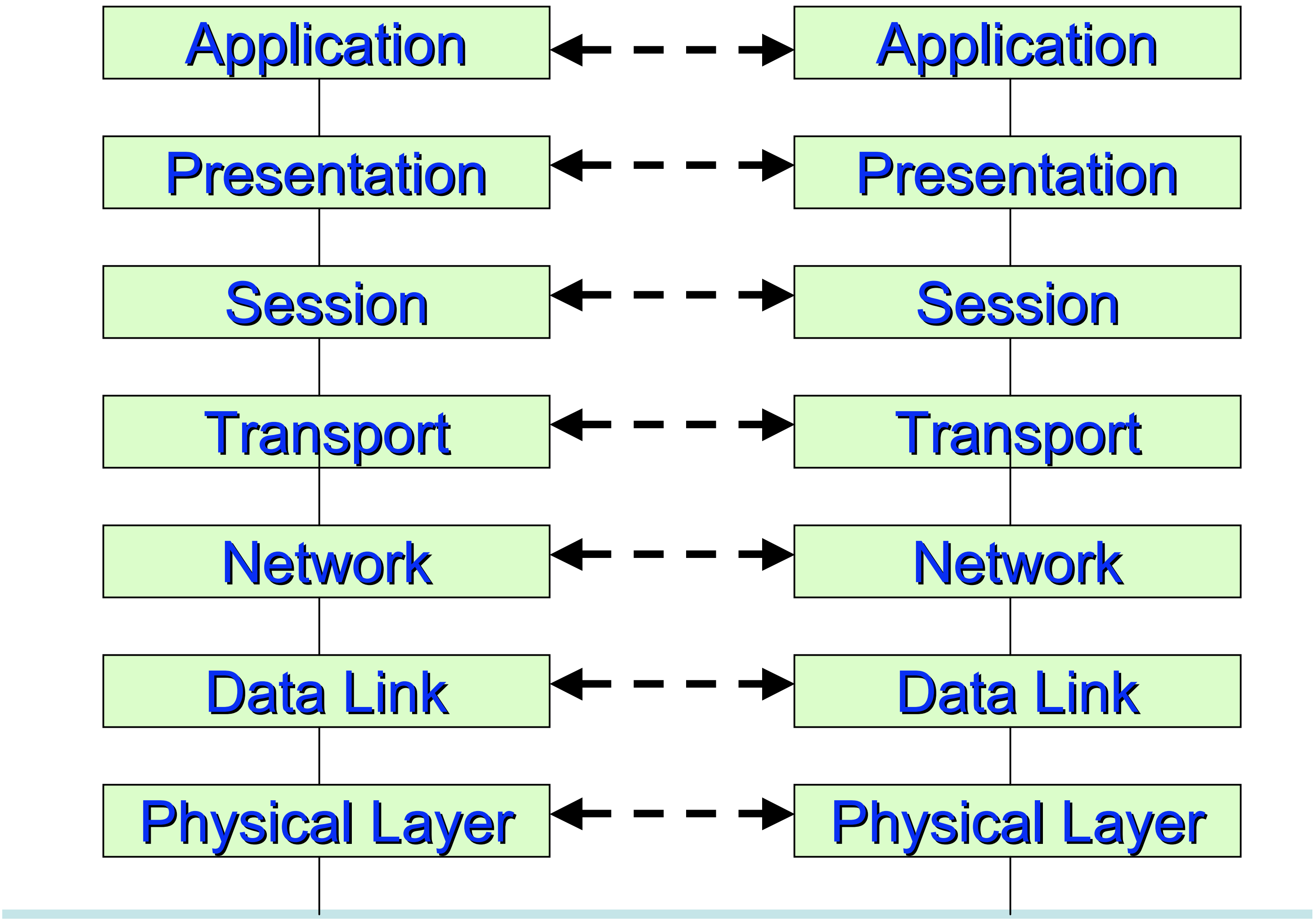
April 21–May 8, 2003.



“Typical” corporate network

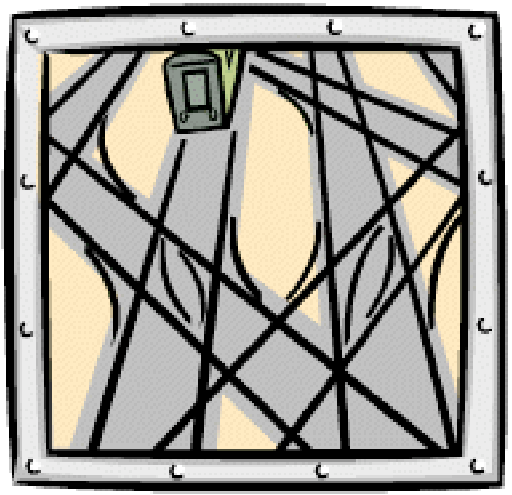


OSI/ISO Model



Infrastructure Protocols

- Transport: IP (between computers), TCP, UDP (between processes)
- Routing:
  - inter-domain:BGP (path vector)
  - intra-domain:OSPF (link state), RIP (distance vector)
  - WLAN: AODV (distance vector), DSR(path vector)
- Service: DNS



Security Services

- **1) Confidentiality:** information is available for reading only to authorized parties.
- **2) Authentication:**
  - Data source authentication: the data is coming from an authorized party.
  - Entity authentication: the entity is who it says it is.
- **3) Integrity:** data was not modified from the source to the destination.

Security Services (2)

- **4) Non-repudiation:** neither the sender, nor the receiver of a message are able to deny the transmission.
- **5) *Access* control:** only authorized parties can use specific resources.
- **6) *Availability*:** resources/services available to authorized parties.

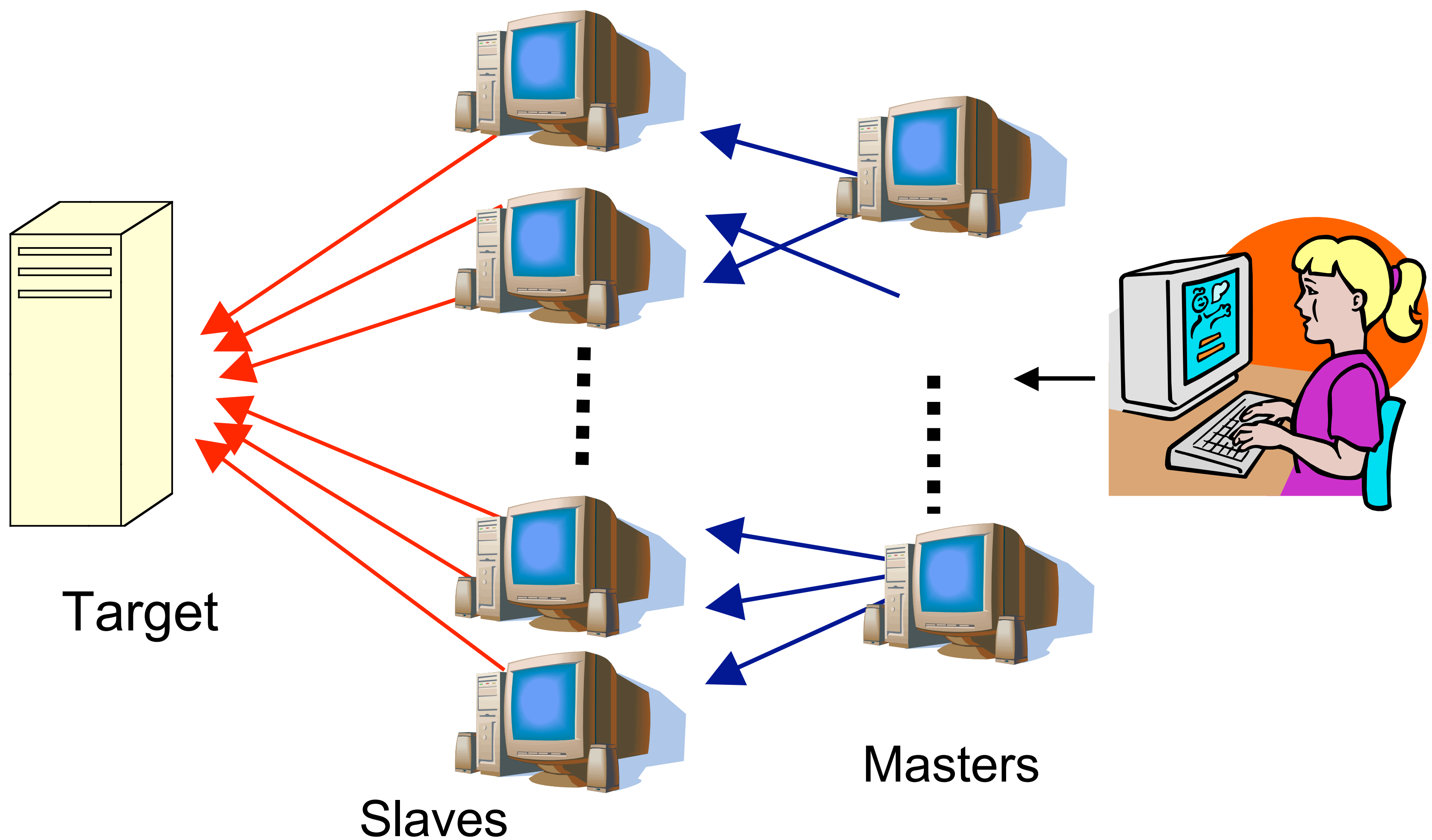
What is Network Security About?

- Protection of networks and their services from unauthorized modification, destruction, or disclosure, and provision of assurance that the network performs its critical functions correctly.
- WHERE ARE WE NOW?
  - Solving confidentiality + integrity+authentication for 2 party protocols: standards like IPSEC, SSL;
  - Authentication and access control: Kerberos, now integrated in Windows 2000
  - Availability: Here we're in trouble!!!!!

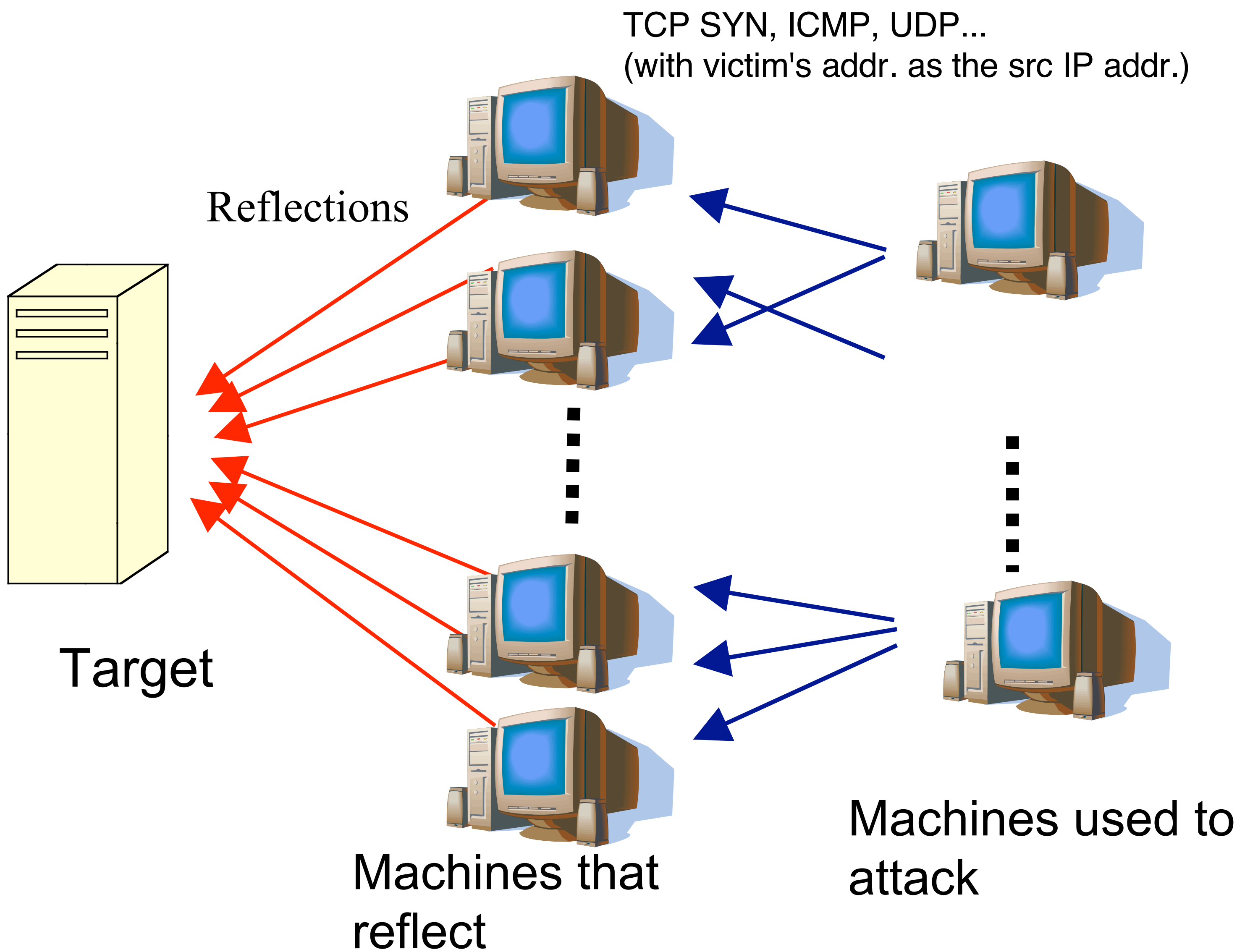
Distributed Denial of Service

- Exploit protocols vulnerabilities or design
- Flooding-based attacks:
  - Single-source attacks
  - Multiple-source attacks
  - Reflector attacks (particular case of multiple-source attacks)
- DDoS: distributed attack, employs a combination of attacks, can attack one machine or more
- DDOS Tools: collection of attacks, attacks is coordinated, master-slaves architecture. Tools like:
  - Trin00
  - Tribe Flood Network.
  - Stacheldracht

Direct DDOS Attack



Reflector DDOS Attack

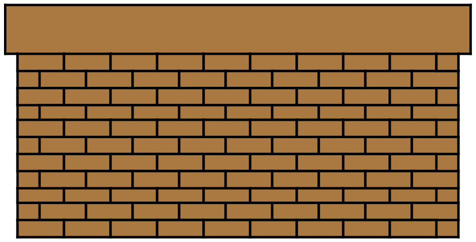


How to Defend?

- The players:
  - Victim network
  - Intermediate network
  - Source network
- Attack is usually **observed close to victim** but **must be stopped close to the source**
- Intermediate network used to **traceback** the attack (probabilistic marking) or to **slow down** the attacks (distributed filters)
- **Reactive methods** (after the attack) and **proactive methods** (prevents the attack) methods

Defence Methods at Victim

- Intrusion detection and firewall: detect packets that look like attacks (known attack signature).
- Make the attack costly for attacker: for example have clients to solve puzzles if they want to access server's resources
- Increase server resources: distributed clusters, load balancing



Intermediate Network

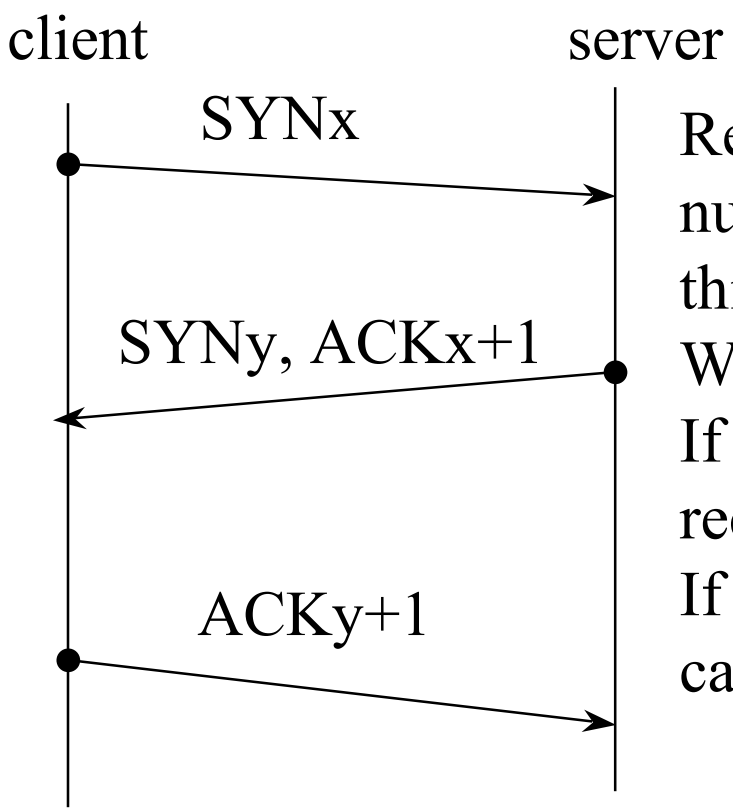
- IP traceback: where is the attack coming from (forensics)
- Push-back mechanism
- Route-based packet filtering

Attacks Against TCP

Transmission Control Protocol - TCP

- Connection oriented protocol for a user process: establishes a connection (channel) between two end-points
- **Reliable**, full-duplex channel: acknowledgements, retransmissions, timeouts
- Congestion control mechanisms
- The packets are delivered in the same order in which they were sent.

TCP Handshake



Resources allocated; There is a max. number of connections that can be in this state (SYN_RECVD state)

Wait for the ACK (75 seconds)

If timeout expires or RST received, data deallocated

If ACK received, connection established, can also contain data.

SYN Attack

- An attacker sends many SYN with source address spoofed packets to a target.
- If the limit is reached, target machine will refuse any incoming connections till the timeout expires.
- Spoofed address chosen to be a non-existent one (If the spoofed address belongs to a machine, then SYN+ACK packet will reach that machine and trigger a RST answer that will close the connection).

Basis of the Attack

- There is no authentication of the source of the packets
- Addresses can be spoofed
- The protocol requires asymmetric allocation of resources

Configuration Optimizations

- System configuration
  - Reduce the timeout to 10 seconds
  - Increase the size of the queue
  - Disable non-essential services, reducing the number of ports to be attacked
- Router configuration
  - Block outside coming packets that have source addresses from the internal network
  - Block packets to the outside that have source addresses from outside the internal network

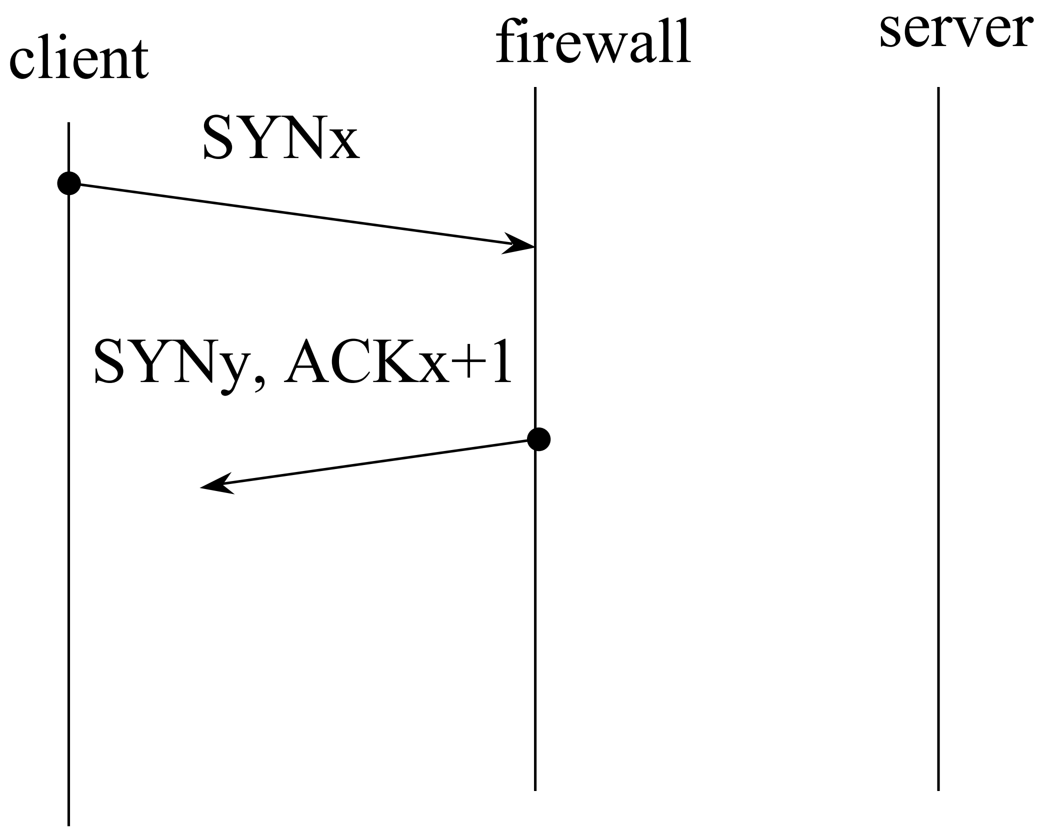
Infrastructure Improvements

- If addresses prefixes separate clear the inside from the outside, then router configuration can be improved.
- Example: routers that attach an organization or an ISP to a backbone network.

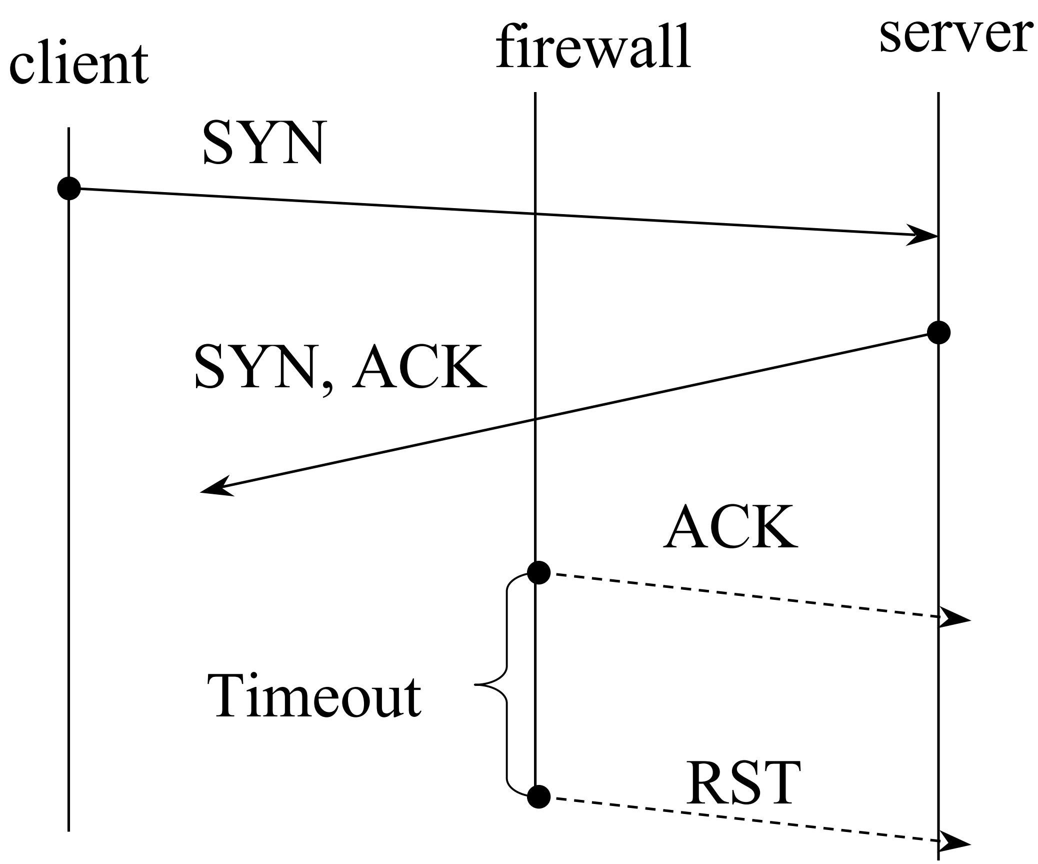
Firewall Approach

- Main idea: each packet for inside network if first examined by the firewall
- Additional delays
- Two approaches:
  - Firewall as a relay
  - Firewall as a gateway

Firewall as a Relay: Attack Scenario



Firewall as a Semi-transparent Gateway: Attack Scenario



Active Monitoring

- Monitor the TCP traffic within a local area network and figure out which ones are illegitimate connections.
- Send RST for the illegitimate connections (this closes the connection).
- Does not require protocol stack modification.
- Monitor can be tricked to classify bad addresses as good addresses

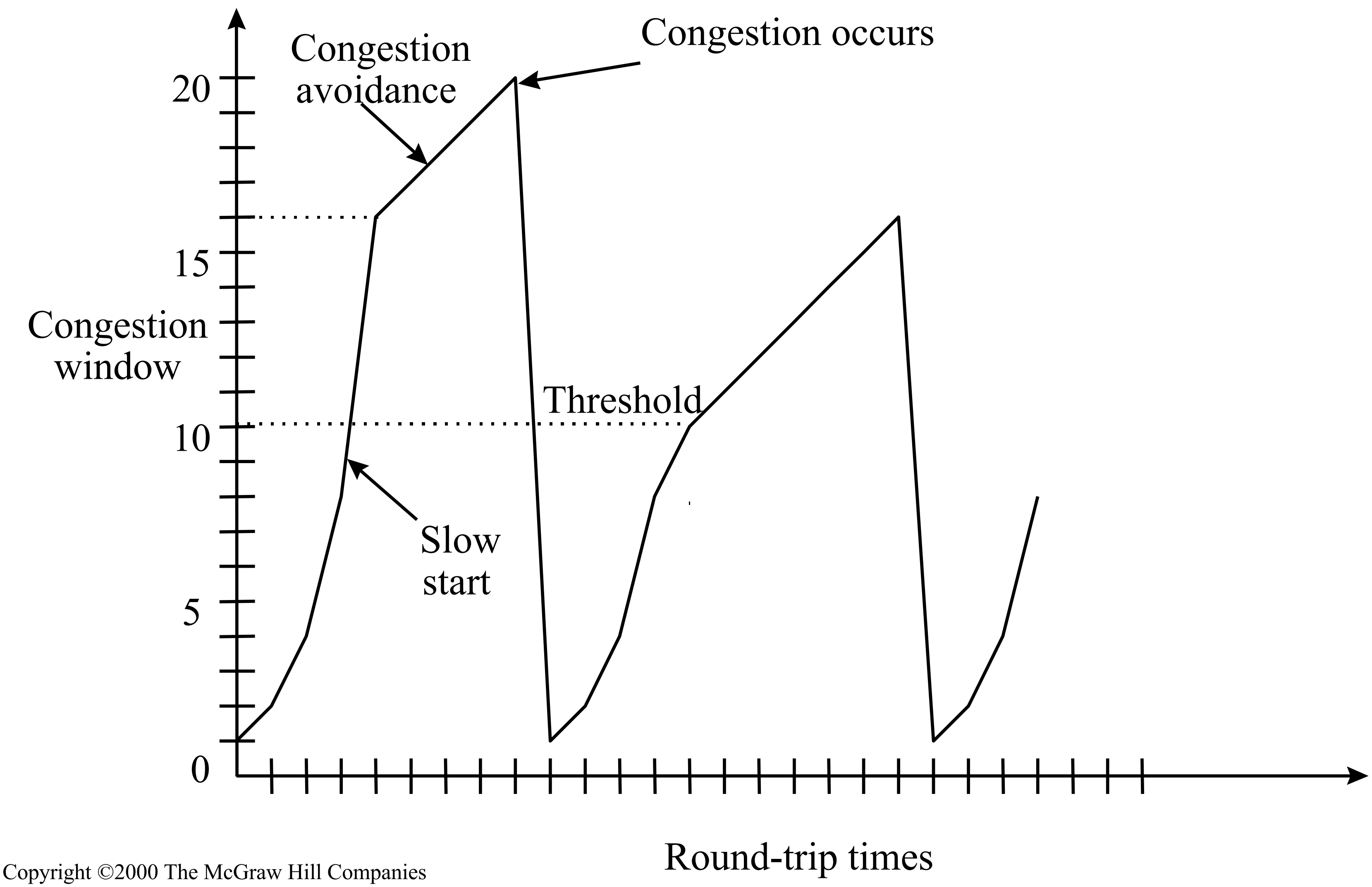
So Far…

- Attacks require **high-rate** transmission (flood of SYN packets), unusual network traffic, **attackers are relatively easy to detect and filter**.
- However ….TCP can be attacked by using TCP friendly traffic (exploit congestion control mechanism), **low rate**, therefore it can cause **maximal damage without detection.**

TCP Congestion Control

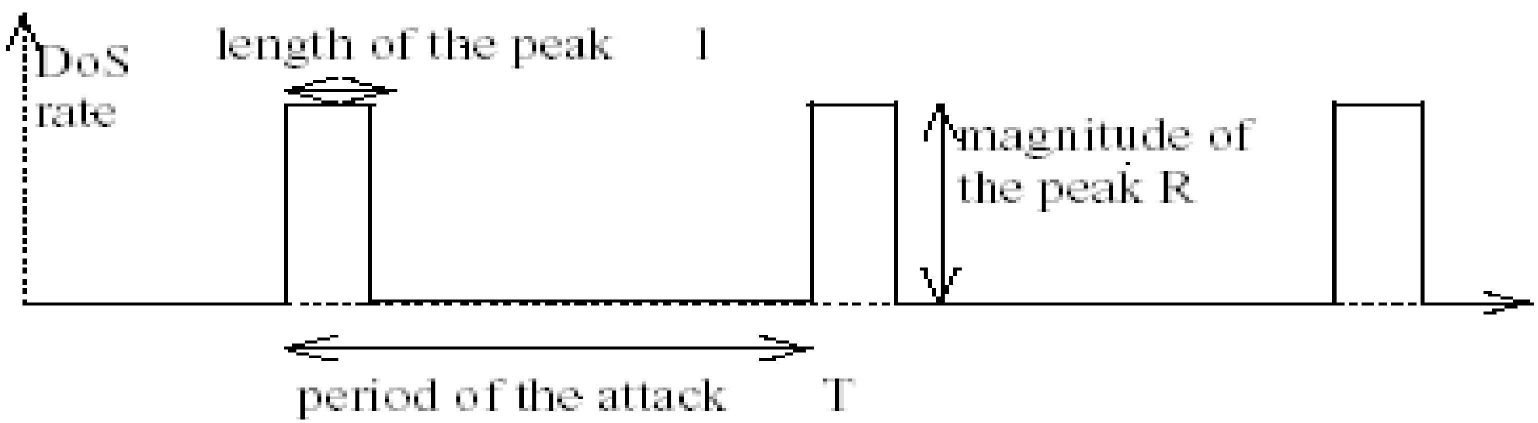
- Source determines how much bandwidth is available for it to send, it starts slow and increases the window of send packet based on ACKS.
- ACKS are also used to control the transmission of packets.
- Uses Additive Increase Multiplicative Decrease (AIMD)
- Uses Retransmission Timeout (RTO) to avoid congestion
- TCP Fairness: if k TCP sessions share same bottleneck link of bandwidth B, each should have average rate of B/k

TCP Congestion Control



The Attack

- All the attacker needs to do is generate a TCP flow to force the targeted TCP connection to repeatedly enter a retransmission timeout state
- Very effective, TCP throughput degrades significantly
- Sending high-rate, RTT scale short duration bursts and repeating periodically at RTO scale period.



Basis of the Attack

- Protocol is homogenous and deterministic
  - protocols react in a pre-defined way
  - tradeoff of vulnerability vs. predictability
- Periodic outages synchronize TCP flow states and deny their service
- Slow time scale protocol mechanisms enable low-rate attacks
  - outages at RTO scale, pulses at RTT scale imply low average rate

Proposed Solutions

- Factors: **randomization, connectivity, accountability**
- Router-Assisted Mechanisms: Routers identify and regulate the misbehaving flows
  - Router-Based algorithms
  - Random early detection with preferential dropping (queue management)
- End-point minRTO Randomization
- They mitigate the attack, but can not eliminate it

Protecting IP: IPSEC

IPSec Overview

- Transparent to applications (below transport layer (TCP, UDP)
- Facilitate direct IP connectivity between sensitive hosts through untrusted networks
- Provides:
  - access control
  - integrity
  - data origin authentication
  - rejection of replayed packets
  - confidentiality
- IETF IPSEC Working Group
- Documented in RFCs and Internet drafts

Security Mechanism

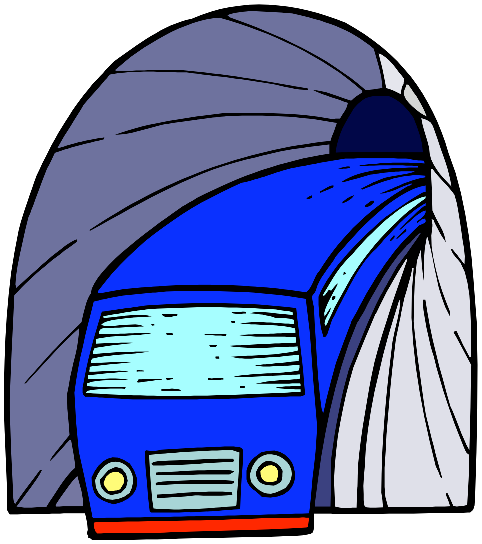
- **Authentication Header (AH):** provides integrity and authentication without confidentiality
- **Encapsulating Security Payload (ESP)**: provides confidentiality and can also provide integrity and authentication
- Operates based on security associations
- **Transport-mode**: encapsulates an upper-layer protocol (e.g. TCP or UDP) and prepends an IP header in clear
- **Tunnel-mode**: encapsulates an entire IP datagram into new packet adding a new IP header

Transport Mode

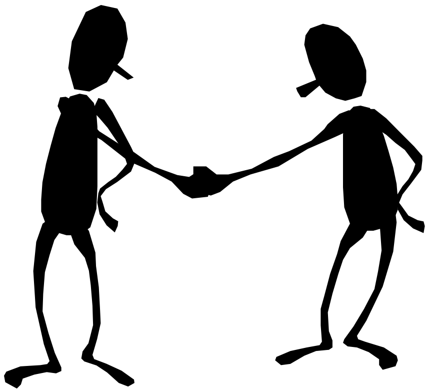
- **ESP in Transport Mode:** encrypts and optionally authenticates the IP payload (data), but not the IP header.
- **AH in Transport Mode:** authenticates the IP payload and selected portions of the IP header.

Tunnel Mode

- **ESP in Tunnel Mode:** encrypts and optionally authenticates the entire inner IP packet, including the inner IP header.
- **AH in Tunnel Mode:** authenticates the entire inner IP packet and selected portions of the outer IP header.



Security Associations (SA)



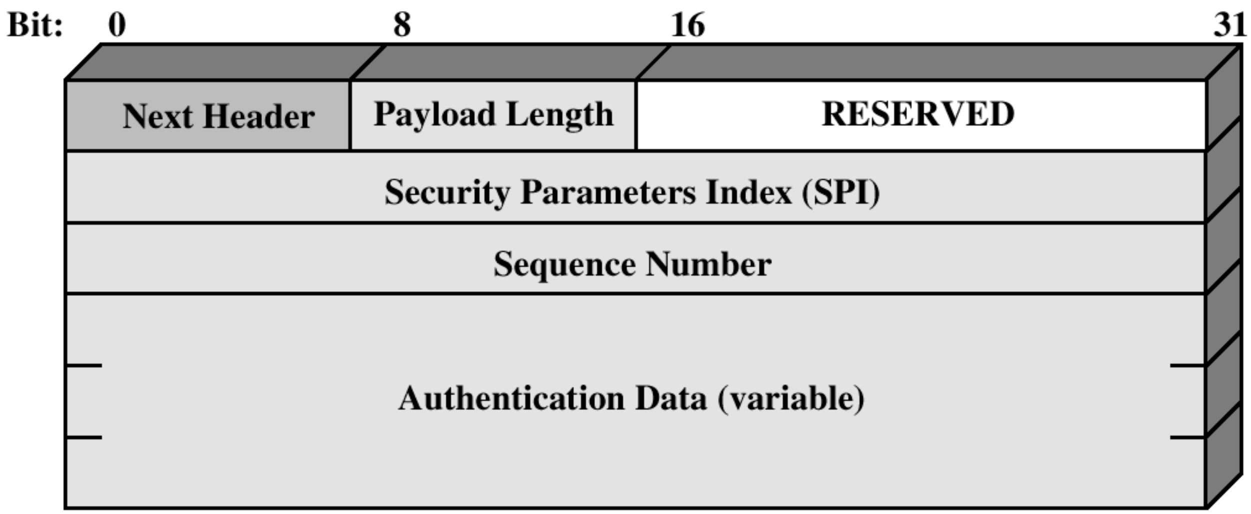
- A relationship between a sender and a receiver.
- Identified by three parameters:
  - Security Parameter Index (SPI)
  - IP Destination address (IP of the destination SA, can be a host, a firewall or a router)
  - Security Protocol Identifier (ESP or AH)
- SPI + IP destination address uniquely identifies a particular Security Association.
- SAs are unidirectional, sender supplies SPI to receiver.

Parameters of a Security Association

- Sequence number counter: used to generate a sequence number in AH and ESP headers
- Sequence counter overflow: how should sequence counter overflow be handled
- Anti-replay window: used to determine if an inbound AH or ESP packet is a replay
- AH information: auth. keys, key lifetime
- ESP information:encryption, auth., key, key lifetime, initial values
- Lifetime of the Security Association
- Protocol Mode: tunnel, transport

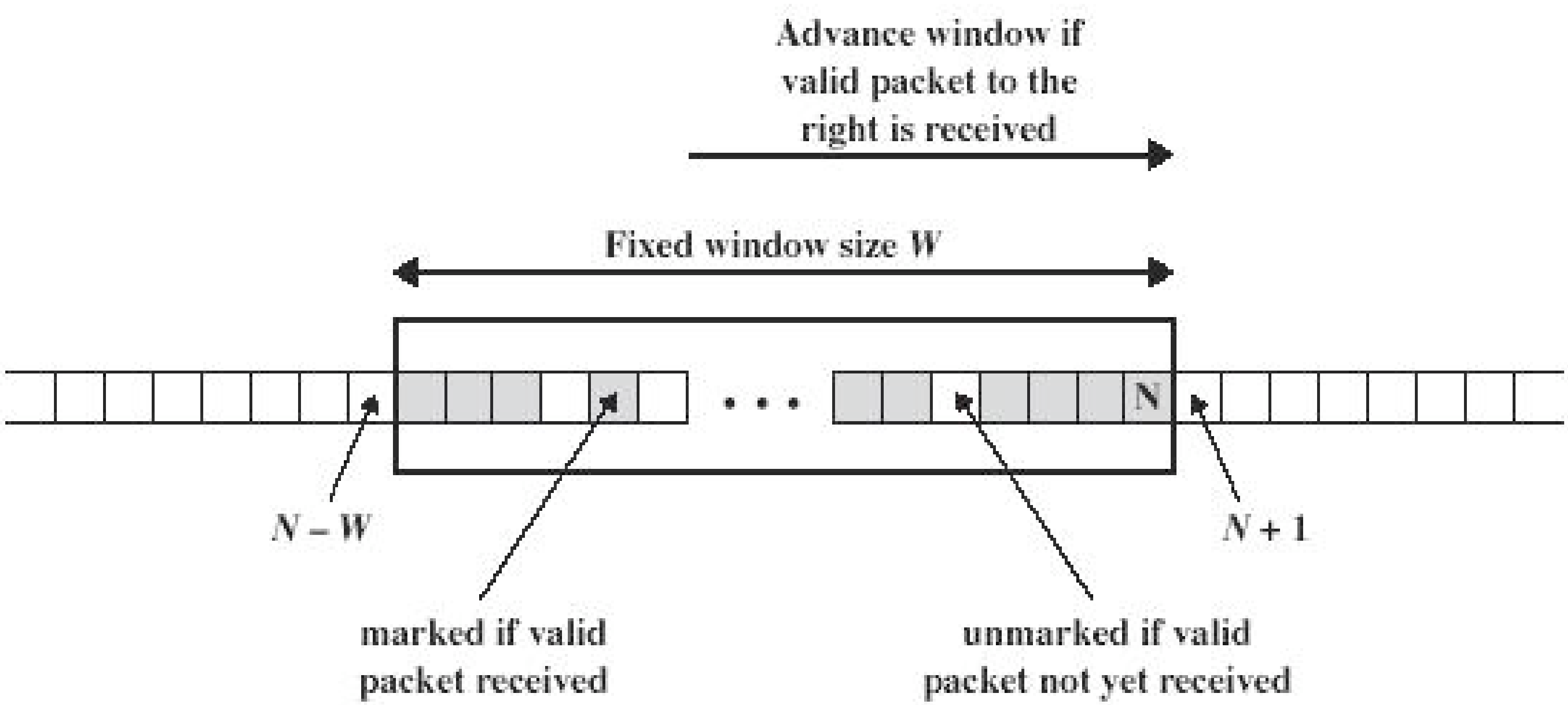
Authentication Header

- Provides support for data integrity and authentication (MAC) of IP packets, using HMAC based on MD5 or SHA1.
- Defends against replay attacks (sequence number).

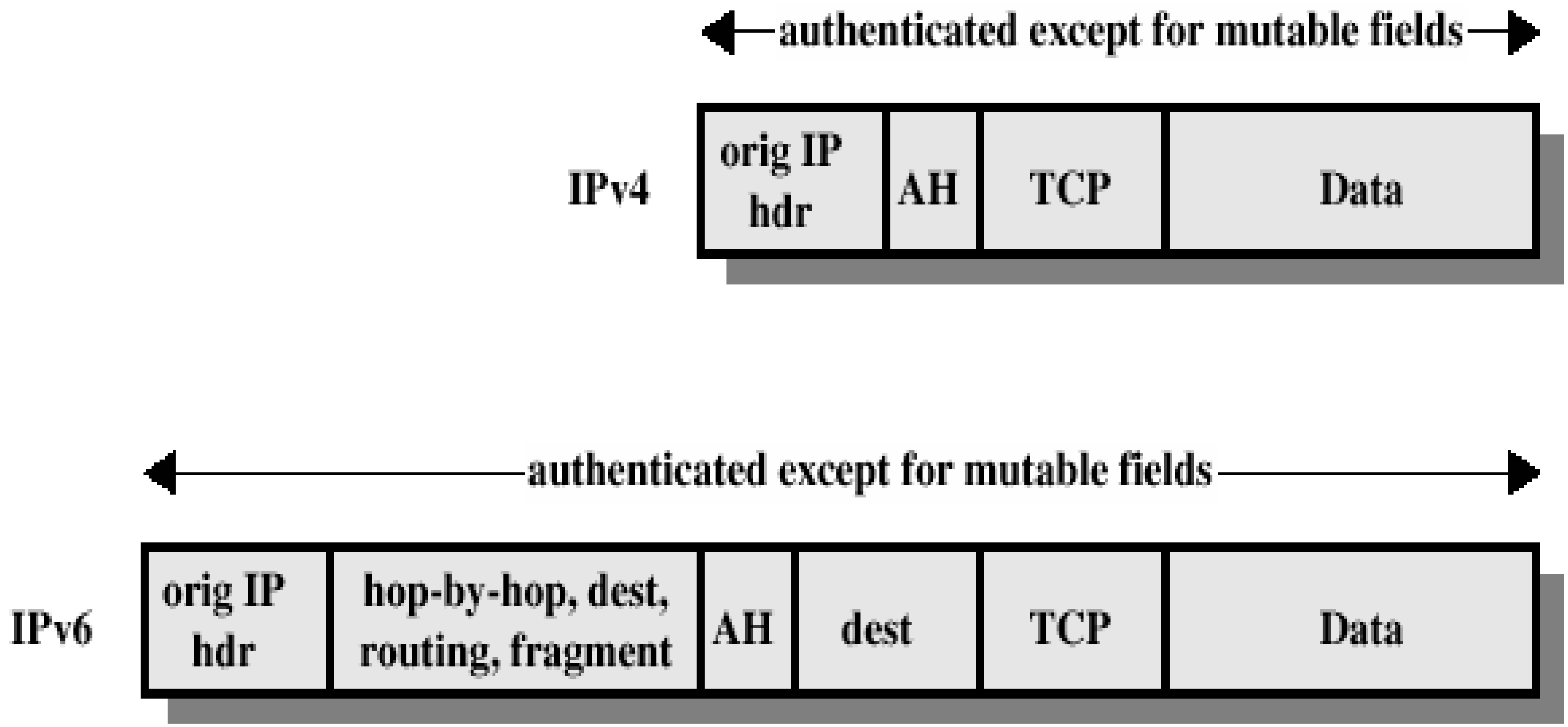


AH: Preventing Replay

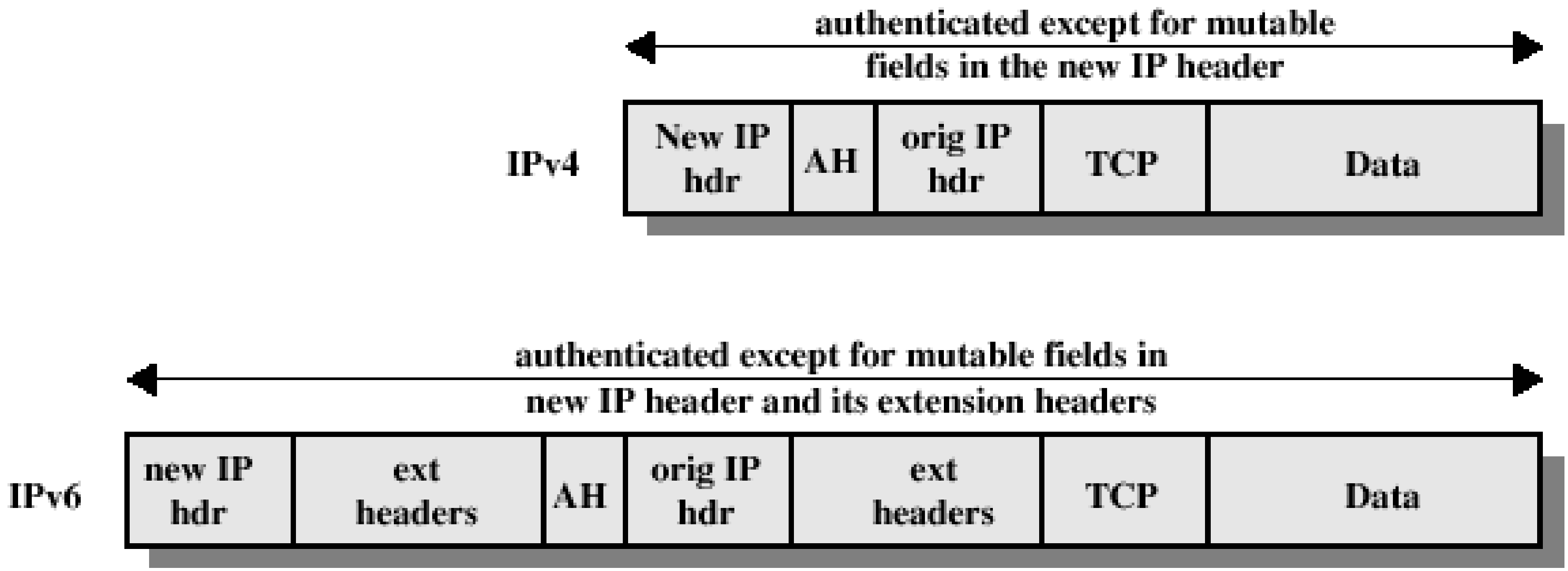
- When a SA is established, sender initializes sequence counter to 0.
- Every time a packet is sent the counter is incremented and is set in the sequnce number in the AH header.
- When sequence number $2^{32} - 1$ is reached, a new SA should be negociated.



AH Authentication: Transport Mode



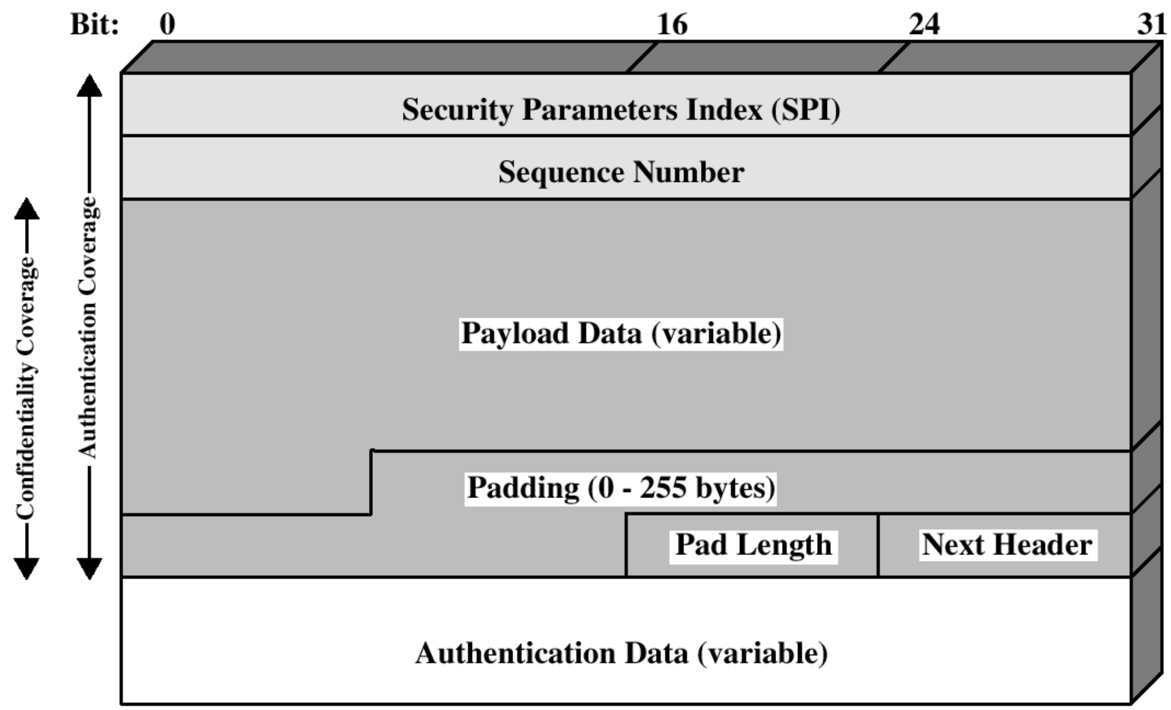
AH Authentication: Tunnel Mode



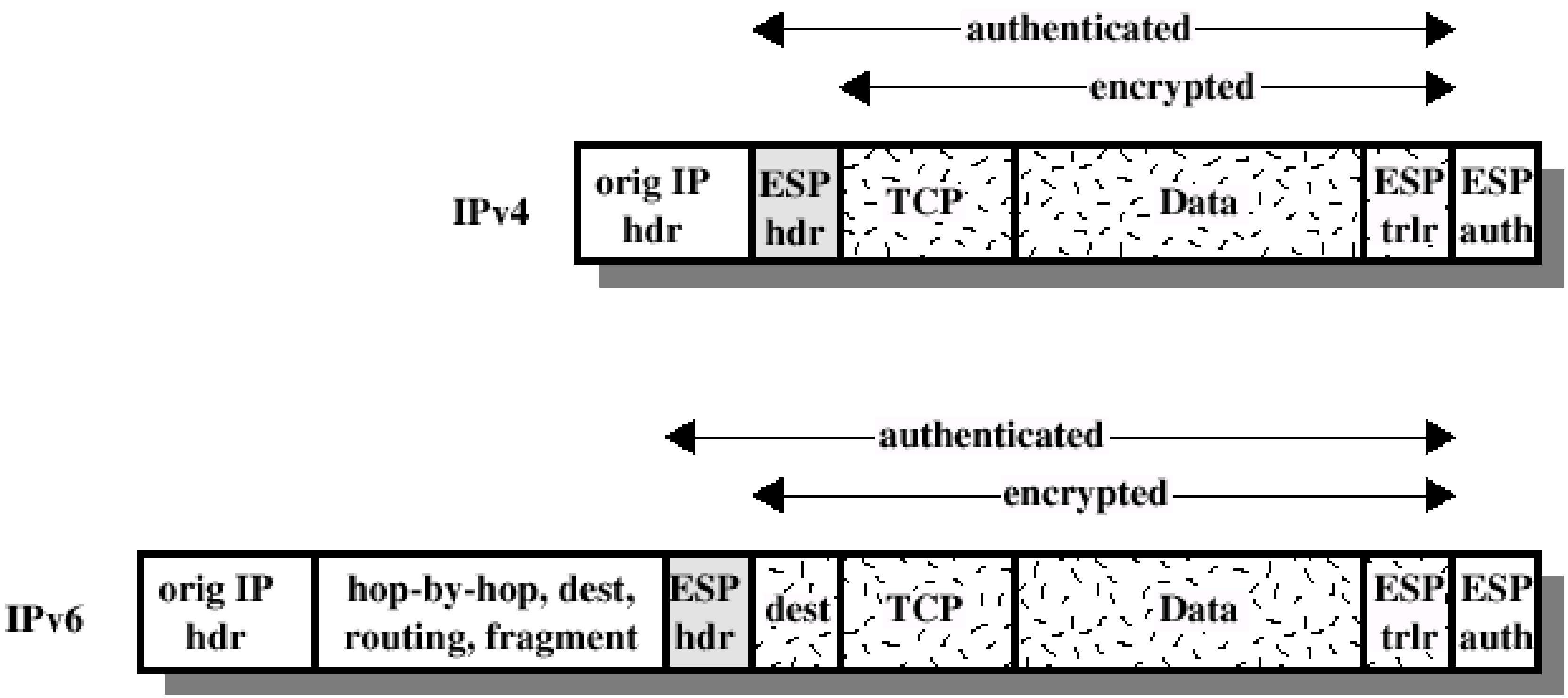
The new IP header contains different IP addresses than the ultimate destination and source

Encapsulating Security Payload

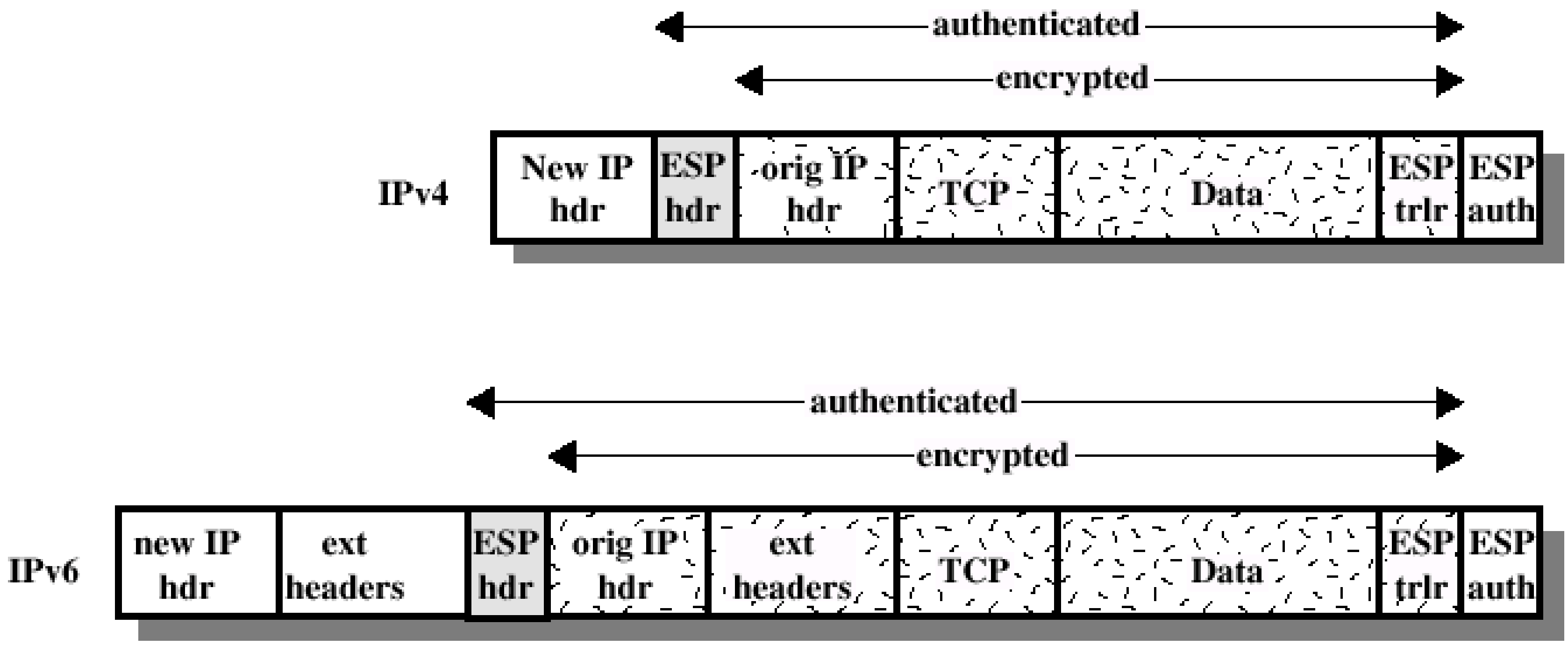
- ESP provides confidentiality services, optionally can provide the same services as AH
- Encryption: 3DES, Blowfish, CAST, IDEA, 3IDEA



ESP Encryption and Authentication: Transport Mode

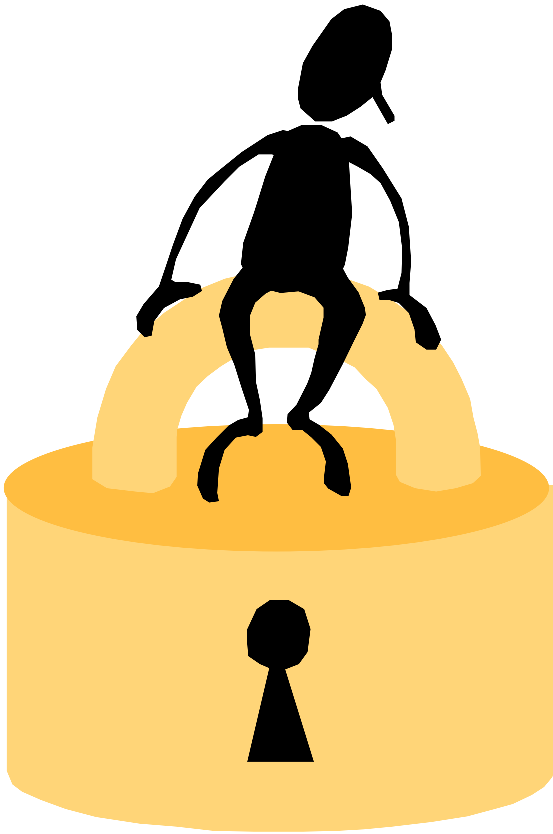


ESP Encryption and Authentication: Tunnel Mode



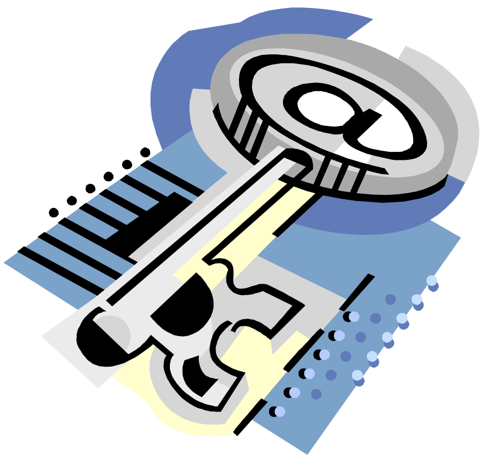
Cryptographic Algorithms

- Encryption:
  - Three-key triple DES
  - RC5
  - IDEA
  - Three-key triple IDEA
  - CAST
  - Blowfish
- Authentication:
  - HMAC-MD5-96
  - HMAC-SHA-1-96



A Word About Key Management

- Manual: system adminstrator configures the keys for hosts
- Automated: on-demand creation of keys
    - Oakley Key Determination Protocol (based on Diffie-Hellman): autheniticated, prevents replays, negociates global parameters
    - Internet Security Association and Key Management Protocol (ISAKMP): Internet key management and negociation, defines procedures and packet formats to establish, negotiate, update, and destroy SAs
- Oakley:
    - Digital signatures
    - Public-key encryption
    - Symmetric-key encryption



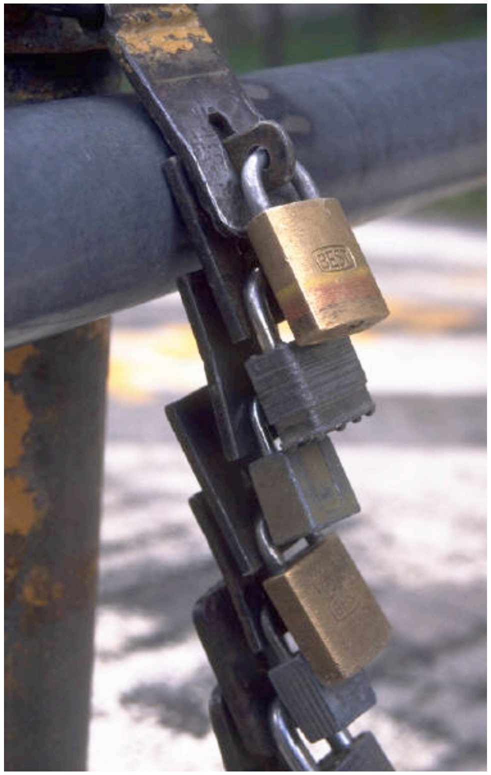
Secure Reliable End-to-End Communication: TLS/SSL

What is Transport Layer Security

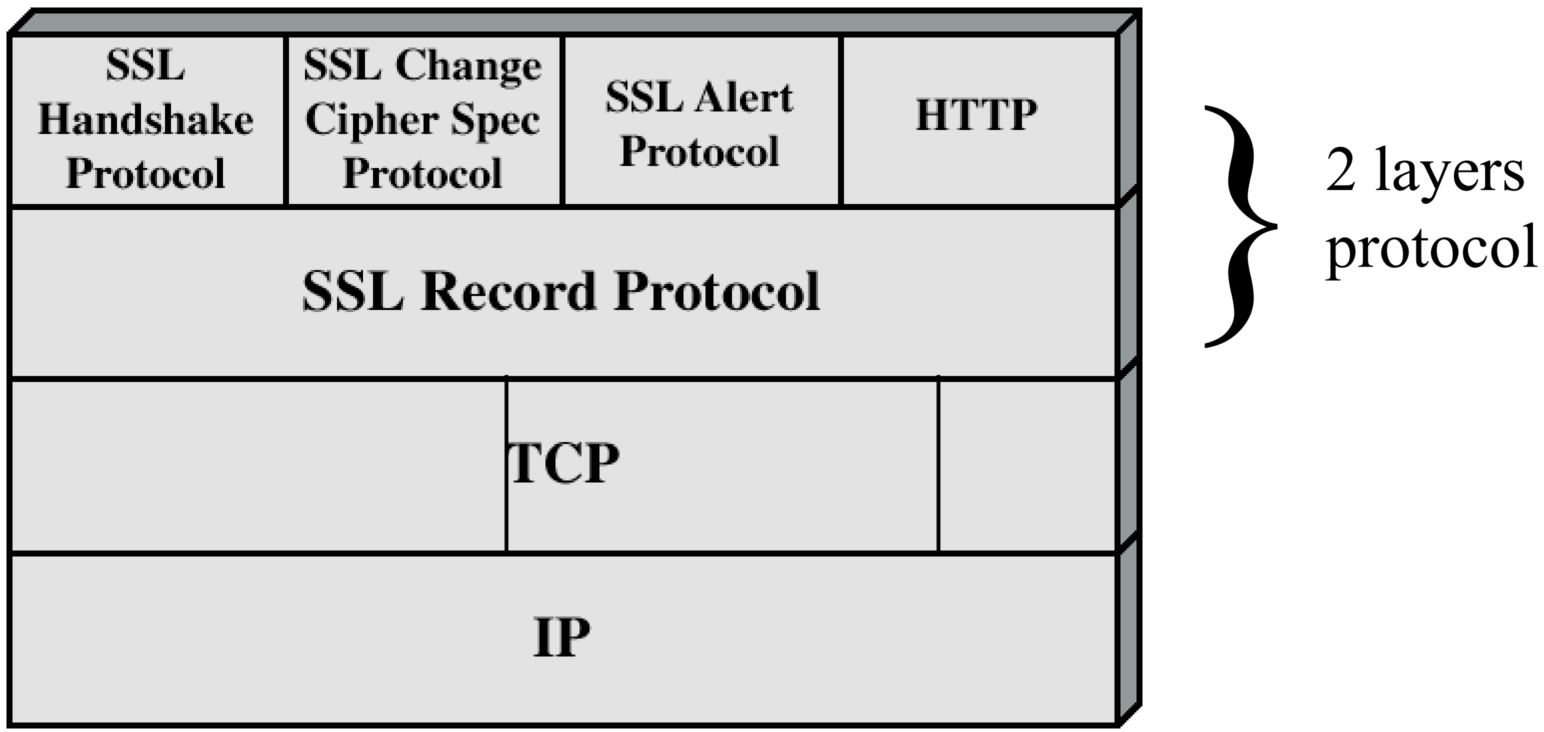
- Protocol that allows to establish an end-to-end secure channel, providing: confidentiality, integrity and authentication
- Defines how the characteristics of the channel are negotiated: key establishment, encryption cipher, authentication mechanism
- Requires reliable end-to-end protocol, so it runs on top of TCP
- It can be used by other session protocols (such as HTTPS)
- Several implementations: for example SSLeay, open source implementation (www.openssl.org)

TLS (cont.)

- **Confidentiality**: Achieved by encryption using DES, 3DES, RC2, RC4, IDEA.
- **Integrity**: Achieved by computing a MAC and send it with the message; MD5, SHA1.
- **Key exchange**: relies on public key encryption for this.



TLS: Protocol Architecture



Session and Connection

- **Session**:
  - association between a client and a server;
  - created by the Handshake Protocol;
  - defines secure cryptographic parameters that can be shared by multiple connections.
- **Connection**:
  - end-to-end reliable secure communication;
  - every connection is associated with a session.

Session

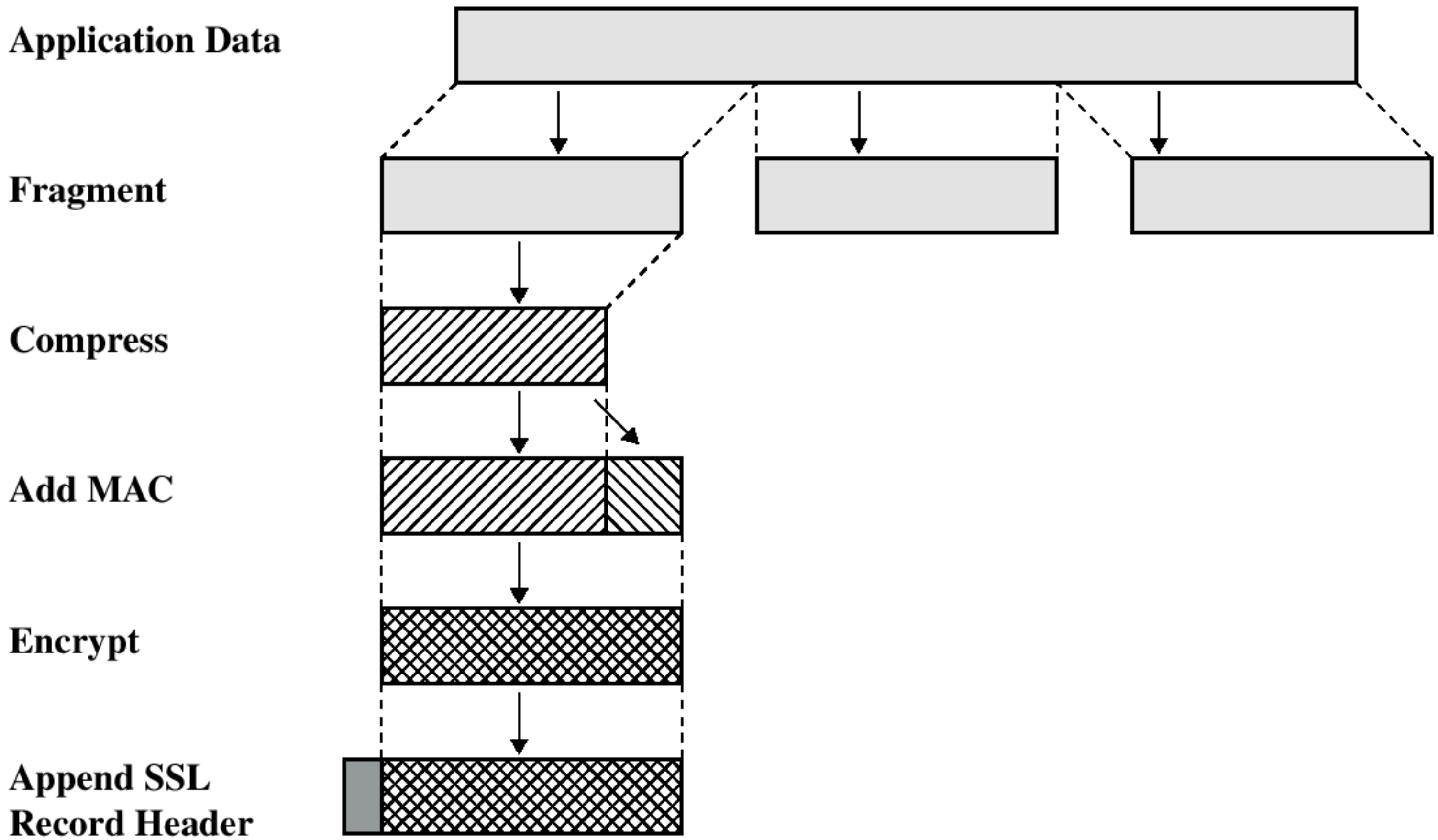
- Session identifier: generated by the server to identify an active or resumable session.
- Peer certificate: X 509v3 certificate.
- Compression method: algorithm used to compress the data before encryption.
- Cipher spec: encryption and hash algorithm, including hash size.
- Master secret: 48 byte secret shared between the client and server.
- Is resumable: indicates if the session can be used to initiate new connections.

Connection

- Server and client random: chosen for each connection.
- Server write MAC secret: shared key used to compute MAC on data sent by the server.
- Client write MAC secret: same as above for the client
- Server write key: shared key used by encryption when server sends data.
- Client write key: same as above for the client.
- Initialization vector: initialization vectors required by encryption.
- Sequence numbers: both server and client maintains such a counter to prevent replay, **cycle is 2^{64} - 1**.

TLS: SSL Record Protocol

- Provides confidentiality and message integrity using shared keys established by the Handshake Protocol

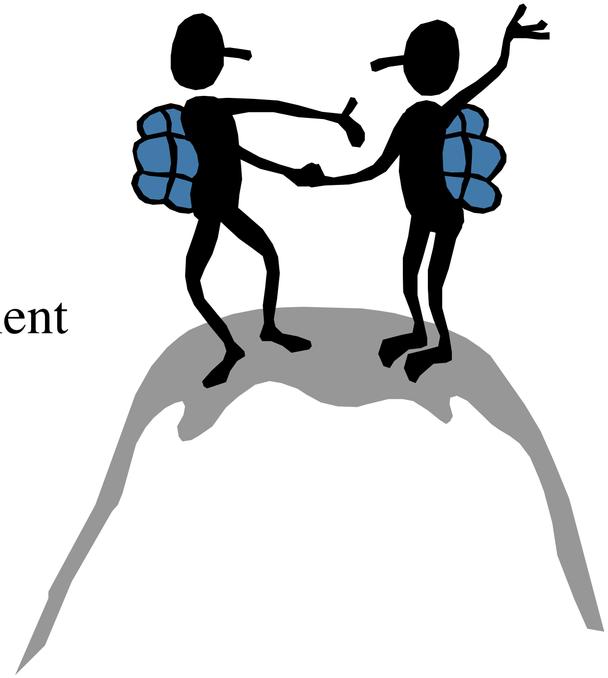


Alert Protocol

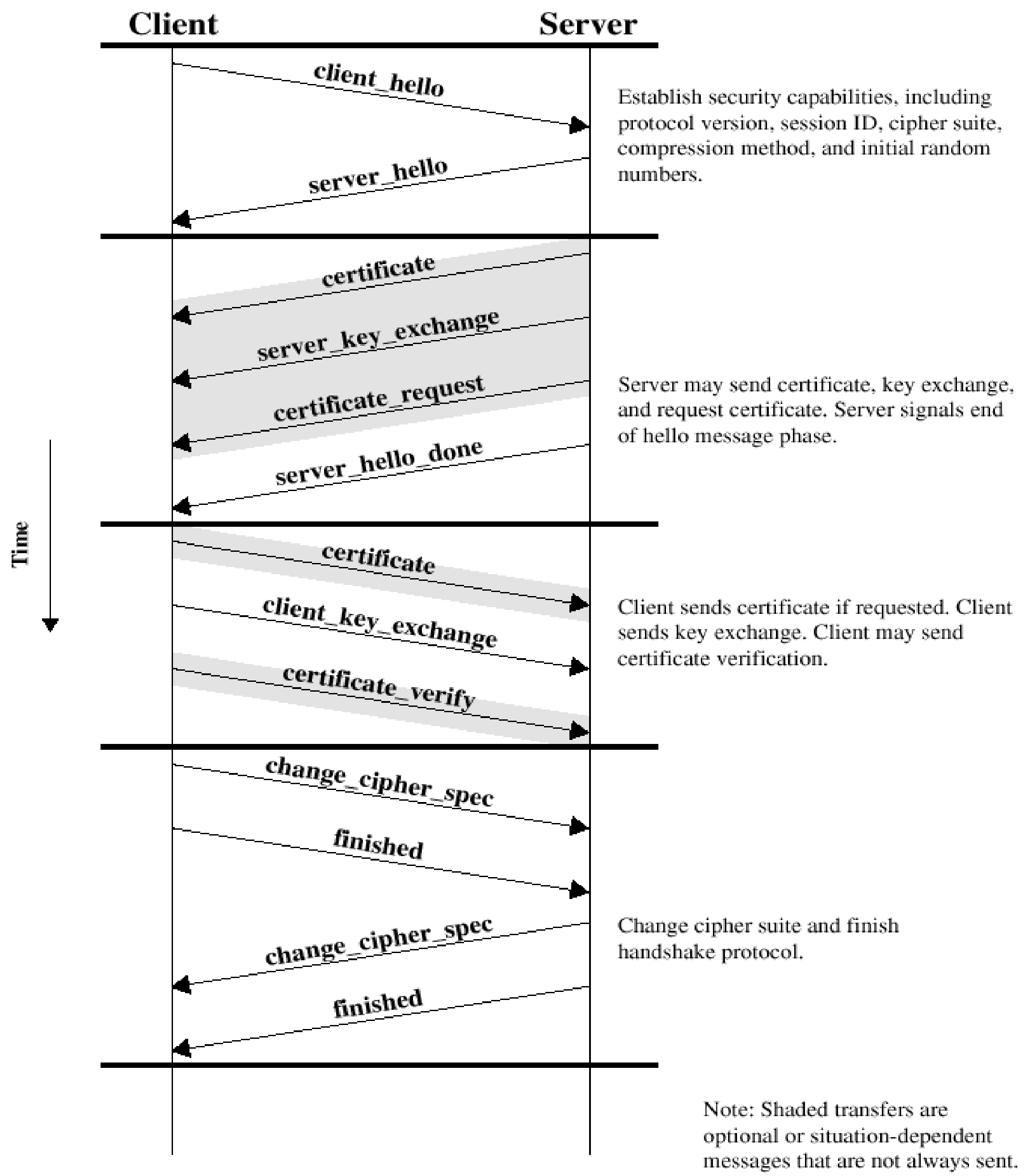
- Used to send TLS related alerts to peers
- **Alert messages are compressed and encrypted**
- Message: two bytes, one defines fatal/warnings, other defines the code of alert
- Fatal errors: decryption_failed, record_overflow, unknown_ca, access_denied, decode_error, export_restriction, protocol_version, insufficient_security, internal_error
- Other errors: decrypt_error, user_cancelled, no_renegotiation

TLS: Handshake Protocol

- Negotiate Cipher-Suite Algorithms
  - Symmetric cipher to use
  - Key exchange method
  - Message digest function
- Establish the shared master secret
- Optionally authenticate server and/or client



Handshake Protocol



Handshake Protocol: Hello

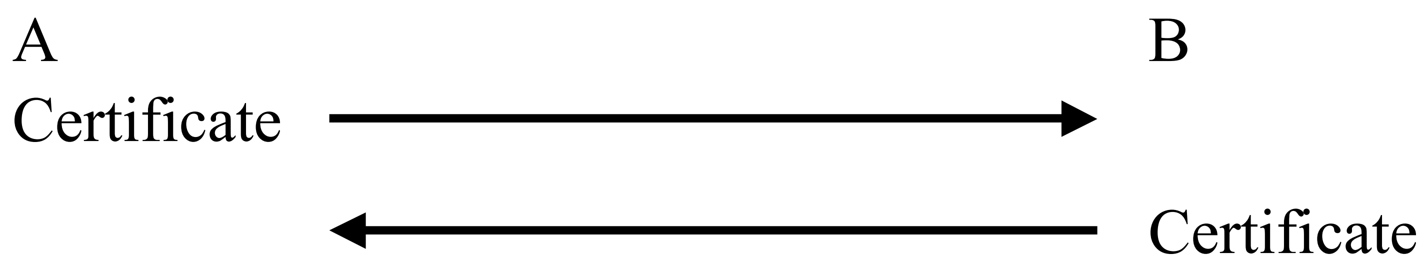
- Client_hello_message has the following parameters:
  - Version
  - Random: timestamp + 28-bytes random
  - Session ID
  - CipherSuite: cipher algorithms supported by the client, first is key exchange
  - Compression method
- Server responds with the same
- Client may request use of cached session
  - Server chooses whether to accept or not

Handshake Protocol: Key Exchange

- Supported key exchange methods:
  - RSA: shared key encrypted with RSA public key
  - Fixed Diffie-Hellman; public parameters provided in a certificate
  - Ephemeral Diffie-Hellman: the best; Diffie-Hellman with temporary secret key, messages signed using RSA or DSS
  - Anonymous Diffie-Hellman: vulnerable to man-in-the-middle

TLS: Authentication

- Verify identities of participants
- Client authentication is optional
- Certificate is used to associate identity with public key and other attributes



TLS: Change Cipher Spec/Finished

- Change Cipher Spec completes the setup of the connections.
- Announce switch to negotiated algorithms and values
- The client sends a message under the new algorithms, allows verification of that the handshake was successful.

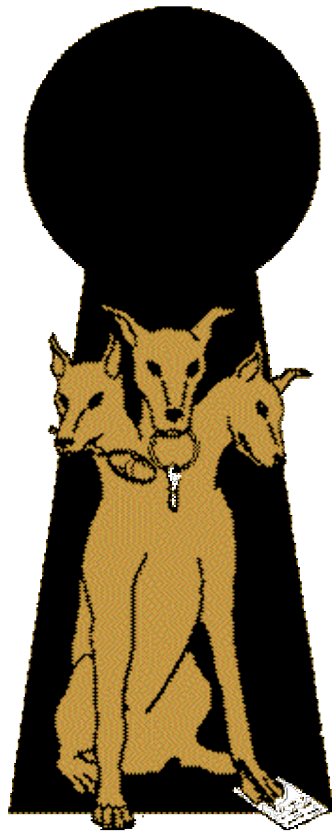
TLS vs. IPSEC

- Security goals are similar
- IPSec more flexible in services it provides, decouples authentication from encryption
- Different granularity: IPSec operates between hosts, TLS between processes
- Performance vs granularity

Distributed Authentication: KERBEROS

What is Kerberos?

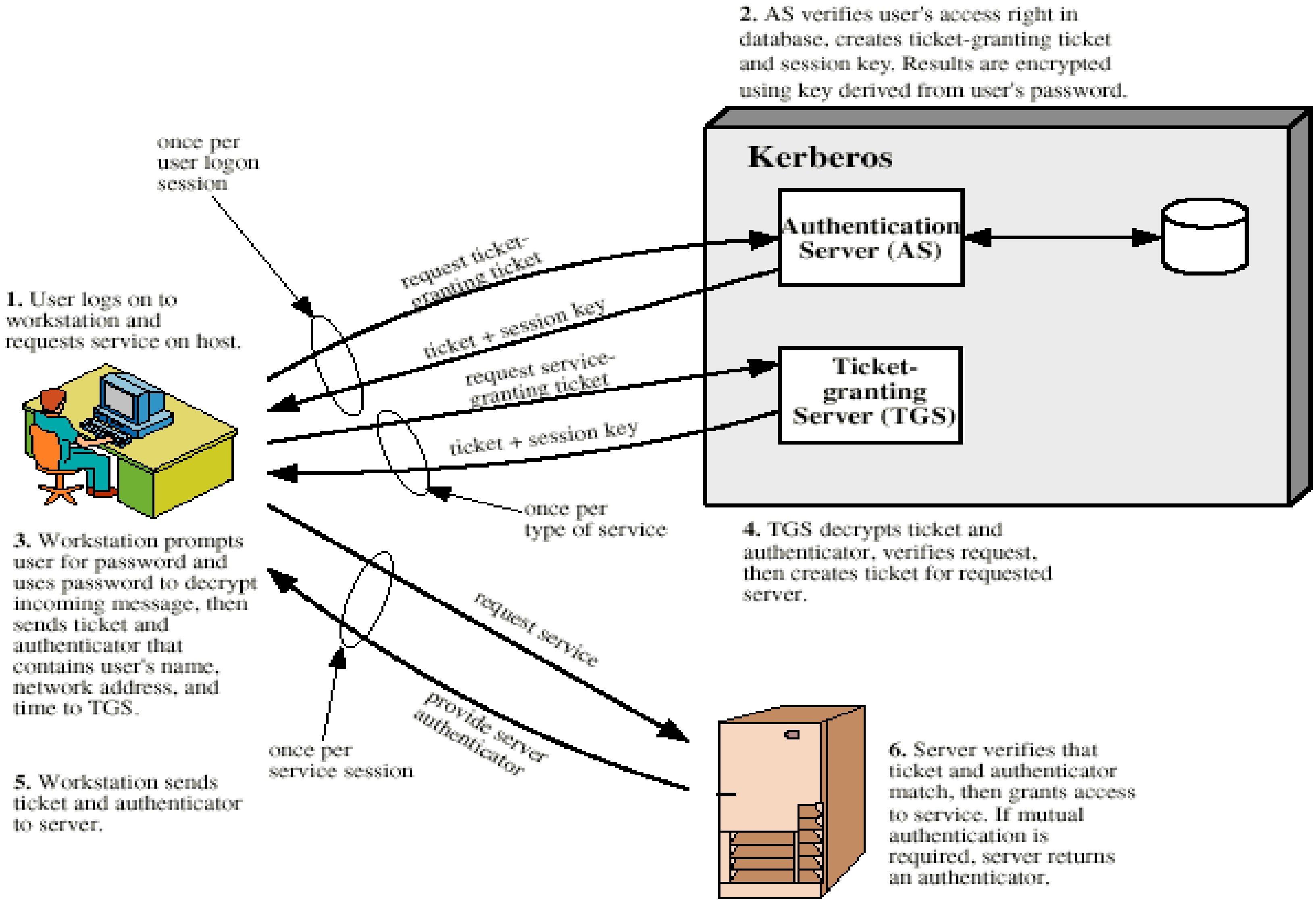
- Kerberos is a **network authentication protocol**
- Provides authentication for client-server applications, and data integrity and confidentiality
- Relies entirely on **symmetric cryptography**
- Developed at MIT: two versios, Version 4 and Version 5 (specified as RFC1510)
- http://web.mit.edu/kerberos/www



Tickets

- Client wants service from a particular server
- An Authentication Server allows access based on tickets
- **Ticket**: specifies that a particular client (authenticated by the Authentication Server) has the right to obtain service from a specified server S
- **Realm**: network under the control of an Authentication Server
- Use two type of tickets with two different lifetimes:
  - One ticket grants to right to ask for service; performed once per login session $Ticket_{tgs}$
  - For each type of service, use a ticket that grants the right to use that particular service $Ticket_S$
  - Every time that service is needed, used the ticket $Ticket_S$

Overview of Kerberos



V4: Authentication Service Exchange

Goal: Obtain Ticket-Granting Ticket

C $\rightarrow$ AS: $ID_c \| ID_{tgs} \| TS_1$
AS $\rightarrow$ C: $E_{Kc}[K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}]$

$Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2]$

ID_{tgs} denotes the identifier of the Ticket Granting Server (TGS)
TS1 and TS2 are timestamps
K_C is the key shared by the AS and client C
$K_{C,\ tgs}$ is the key shared by the TGS and client C
K_{tgs} key known by AS and the TGS
$Ticket_{tgs}$ …is the ticket
Lifetime is the validity of the ticket
AD is address identifier

V4: Ticket-Granting Service Exchange

Goal: Obtain Service-Granting Ticket

C $\rightarrow$ TGS: $ID_S \parallel Ticket_{tgs} \parallel Authenticator_C$

TGS $\rightarrow$ C: $E_{K_{c,tgs}}$ [$K_{C,S} \parallel ID_S \parallel TS_4 \parallel Ticket_S$]

$Ticket_{tgs} = E_{K_{tgs}}$ [$K_{C,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2$]

$Ticket_S = E_{K_S}$ [$K_{C,S} \parallel ID_C \parallel AD_C \parallel ID_s \parallel TS_4 \parallel Lifetime_4$]

$Authenticator_C = E_{K_{C,\ tgs}}$ [$ID_C \parallel AD_C \parallel TS_3$]

K_S is the key shared by the TGS and server S

V4: Client-Server Authentication Exchange

Goal: Obtain Service

$C \rightarrow S$: $Ticket_S \parallel Authenticator_C$

$S \rightarrow C$: $E_{K_{C,S}}[\ TS_5 + 1\ ]$

$Ticket_S = E_{K_S}[\ K_{C,S} \parallel ID_C \parallel AD_C \parallel ID_s \parallel TS_4 \parallel Lifetime_4\ ]$

$Authenticator_C = E_{K_{C,S}}[\ ID_C \parallel AD_C \parallel TS_5\ ]$

Request for Service in Another Realm

- Authenticate to local AS and obtain ticket to local TGS
- Ask local TGS for ticket for remote TGS, obtain ticket for remote TGS
- Ask remote TGS for ticket for remote server S, obtain ticket for remote server S
- Ask for service from remote server S

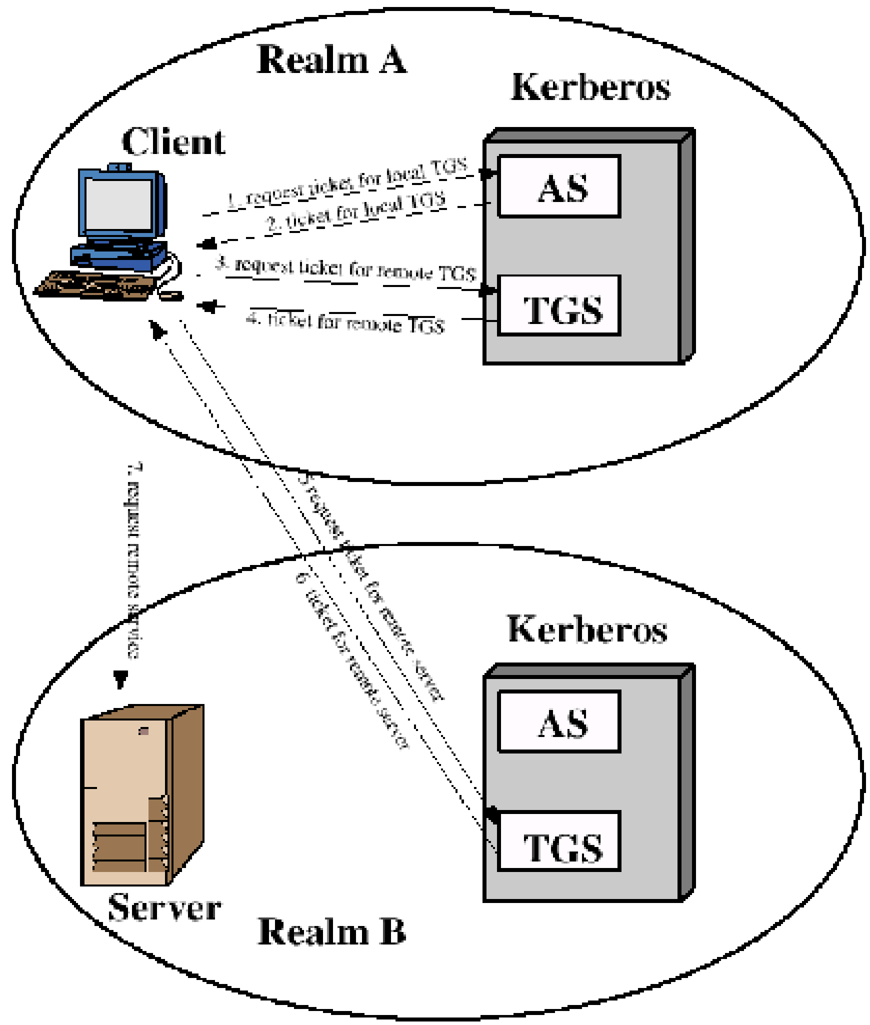


Figure 4.2 Request for Service in Another Realm

Kerberos Version 4 vs. Version 5

- Version 5 overcomes certain deficiencies in Version 4: environment and technical (1994)
- Environment:
  - V4 was using DES as encryption and there were restrictions; not general enough
  - Depending on IP, modify such that any network type address can be used
  - Message byte ordering; move the standards that provide unambiguous byte ordering
  - Ticket lifetime: V5 allows arbitrary lifetimes
  - Inter-realm authentication: V4 requires N^2 keys; V5 is better

Kerberos Version 4 vs. Version 5

- Technical:
  - V5 eliminates one unnecessary encryption
  - V4 was using a non-standard DES encryption mode that was found vulnerable; V5 uses CBC mode.
  - Use sub-session keys
  - Include a pre-authentication protocol that makes password attacks more difficult

V5: Authentication Service Exchange

Goal: Obtain Ticket-Granting Ticket

C → AS: $Options \parallel ID_c \parallel Realm_C \parallel ID_{tgs} \parallel Times \parallel Nonce_1$

AS → C: $Realm_C \parallel ID_C \parallel Ticket_{tgs} \parallel E_{Kc}\,[K_{c,tgs} \parallel Times \parallel Nonce_1 \parallel Realm_{tgs} \parallel ID_{tgs}]$

$Ticket_{tgs} = E_{K_{tgs}}\,[Flags \parallel K_{c,tgs} \parallel Realm_C \parallel ID_C \parallel AD_C \parallel Times]$

V5: Ticket-Granting Service Exchange

Goal: Obtain Service-Granting Ticket

C $\rightarrow$ TGS: Options $\|$ ID_S $\|$ Times $\|$ $Nonce_2$ $\|$ $Ticket_{tgs}$ $\|$

$Authenticator_C$

TGS $\rightarrow$ C: $Realm_C$ $\|$ ID_C $\|$ $Ticket_S$ $\|$ $E_{K_{c,tgs}}$ [$K_{C,S}$ $\|$ Times $\|$

$Nonce_2$ $\|$ $Realm_S$ $\|$ ID_S]

$Ticket_{tgs} = E_{K_{tgs}}$ [Flags $\|$ $K_{C,tgs}$ $\|$ $Realm_C$ $\|$ ID_C $\|$ AD_C $\|$ Times]

$Ticket_S = E_{K_S}$ [Flags $\|$ $K_{C,S}$ $\|$ $Realm_C$ $\|$ ID_C $\|$ AD_C $\|$ Times]

$Authenticator_C = E_{K_{C,\ tgs}}$ [ID_C $\|$ $Realm_C$ $\|$ TS_1]

V5: Client-Server Authentication Exchange

Goal: Obtain Service

C S: Options || $Ticket_S$ || $Authenticator_C$

S C: $E_{Kc,s}$ [TS_2 || Subkey || Seq#]

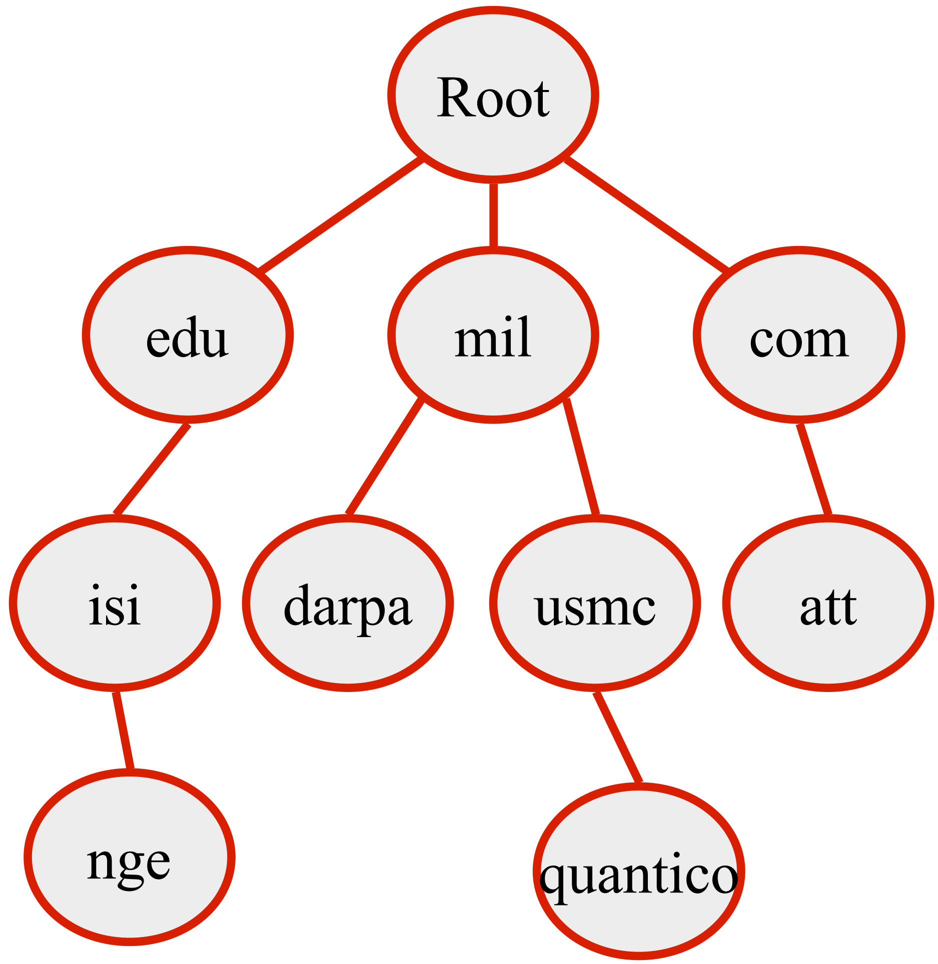
$Ticket_S = E_{K_S}$ [Flags || $K_{C,S}$ || $Realm_C$ || ID_C || AD_C || Times]

$Authenticator_C = E_{K_{C,S}}$ [ID_C || $Realm_C$ || TS_2 || Subkey || Seq#]

DNS: VULNERABILITIES AND COUNTERMEASURES

DNS

- Tree structure
  - Divided into zones
  - Delegating responsibilities
- ICANN oversees the domain name assignments
- Name servers
  - Authoritative information (hints to whom might be able to answer the request)
  - Cached data updated periodically



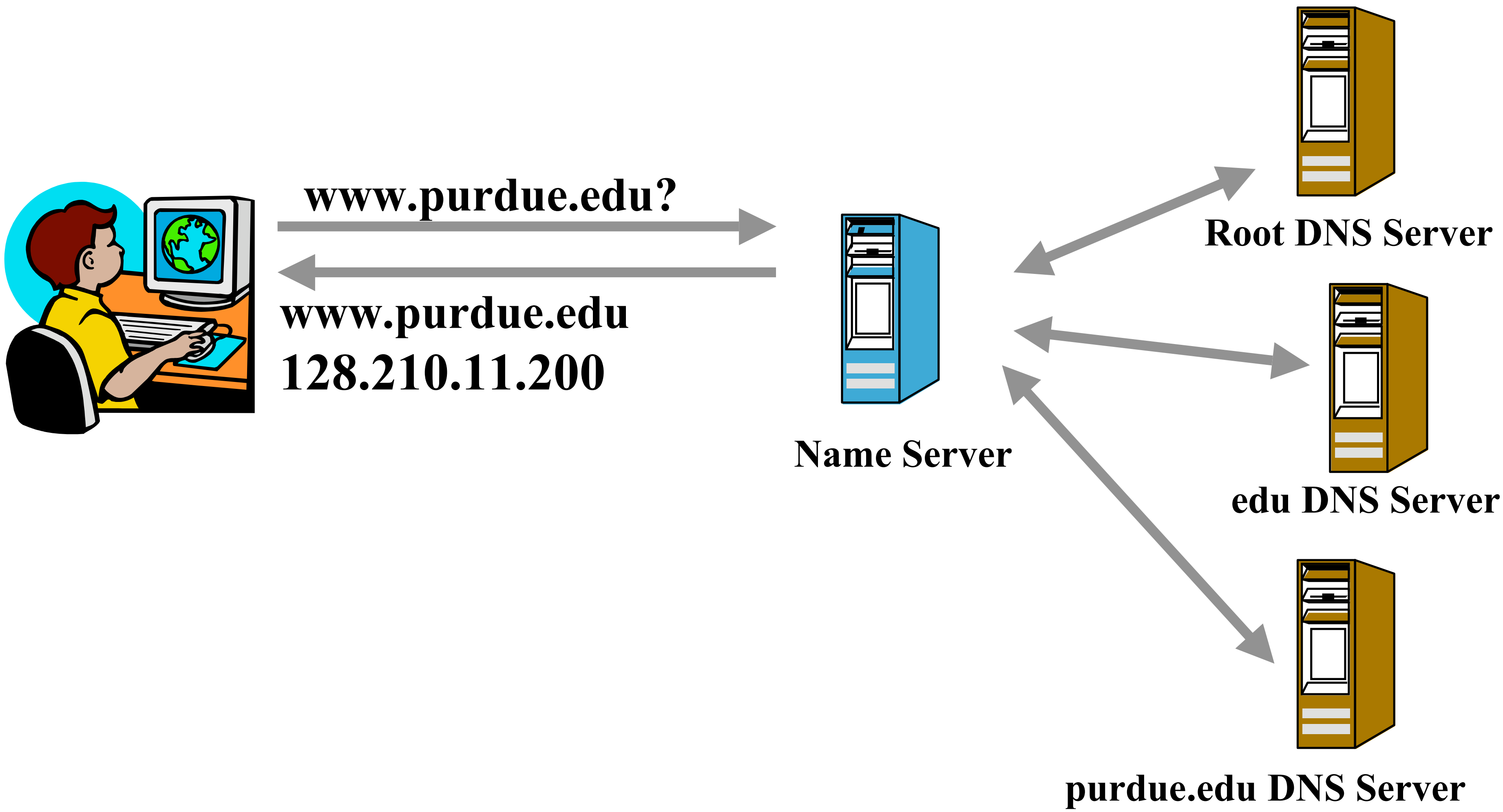
Hierarchical Structure

- 13 root servers: **10 in USA, 1 in Sweden, 1 in UK and 1 in Japan**.
- What kind of traffic? Oct 2002, 24 hours on f.root-servers.net root, 14GB, 152,744,325 queries, **1768 queries per second**
- Top Level Domain (TLD) operate ".com", ".edu", etc
- Name servers

Name Server

- Each zone has a name server that maintains database of host information for its zone
- Contact the authoritative NS of that zone to get host information (such as IP)
- Information needs to be updated when host info changes in the zone
- Dynamic updates change DNS data without having to rebuild any other part of the DNS tree

How Does it Work?



DNS Vulnerabilities

- **Denial of service**: servers bombarded with requests
  - Defective implementations RFC1918 (private addresses) that propagate requests/updates that were not supposed to happen (blackhole servers now collect and drop this traffic)
  - Malicious attacks: Oct. 2002, DDoS, 9 of the root servers were affected (about 1 hour, ICMP flooding);
- **Defense against denial of service:**
  - Clustering and load balancing
  - Queries rate controlled, each source address is limited to a 10KBits/sec and queue size of 3 packets.

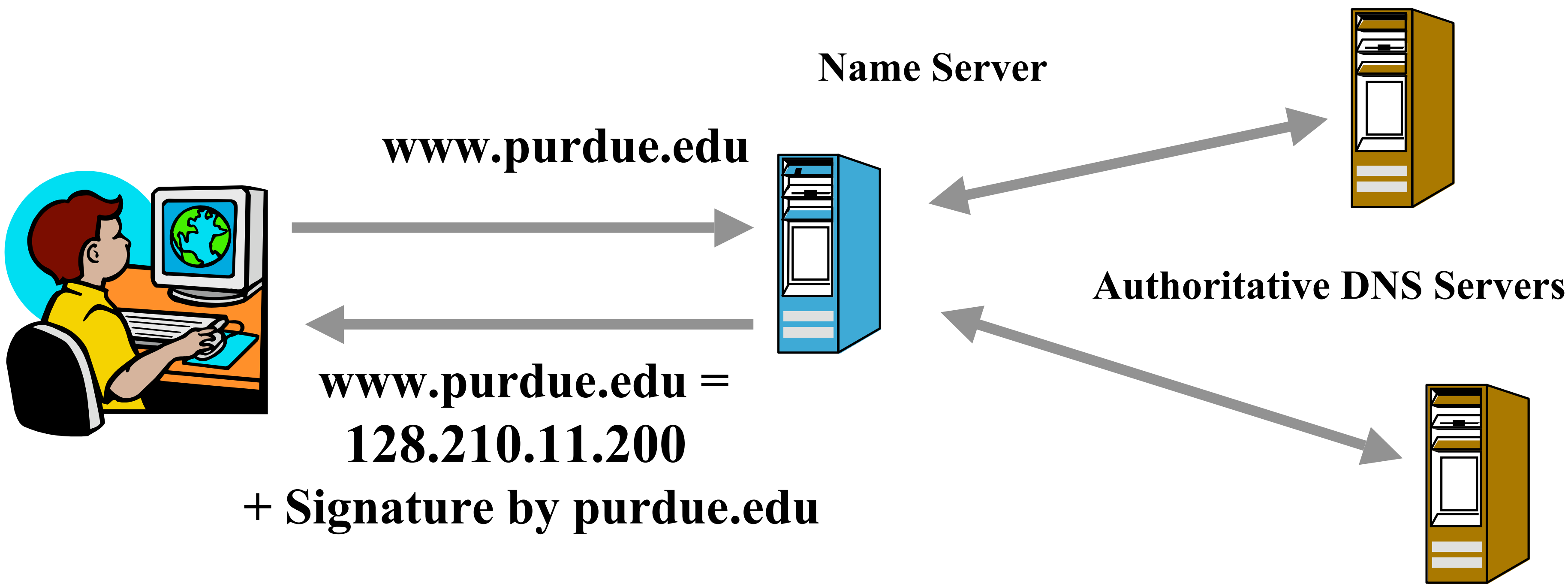
DNS Vulnerabilities (cont.)

- **DNS Spoofing**:
  - Guessing DNS queries Ids (man in the middle)
  - Compromise the DNS servers itself
- **Cache Poisoning**: False IP with a high TTL, which the DNS server will cache for a long time
- **Email Spoofing**: Registration with ICANN often done via email and authenticated by the email address. Return addresses can be falsified
- **Mis-configuration**: Administrator enters the DNS information incorrectly

DNSSEC

- **Proposed solution: addressing authentication and integrity (digital signatures)**
- Each DNS zone signs its data using its private key (signing can be done offline, in advance)
- Query for a record will return the requested resource and a digital signature of the requested resource record set
- Resolver will authenticate the response using the corresponding public key of the zone

Secure DNS


Name Server
www.purdue.edu
Authoritative DNS Servers
www.purdue.edu =
128.210.11.200
+ Signature by purdue.edu

What are the Issues?

- How to obtain the public key to verify the digital signature (chicken-and-egg problem)
- Key management is critical (connected with flexibility, original design (RFC 2535) was fatally flawed because did not consider carefully key management)
- Denial of existence : prove a domain for which a query was made, does not exist
- Incremental deployment, flexible to add new domains
- Cryptography alone adds new DoS due to crypto errors and attacks

Key Validation

- How to obtain certified public keys of zones, to verify the digital signatures
- New DNS records KEY, signed by servers in other zones
- Approaches
  - Tree structure: each parents signs the keys of children
  - PGP-style web of trust
  - Mesh: combination between the above, specifies how to find a path of trust

DNS: Summary

- DNS is a fundamental service suffering from numerous vulnerabilities
- DNSSEC proposed to provide authentication and integrity, based on digital signatures
- Main issues: deployment, how to obtain the public key to verify signatures, key rollover, authentication of denial of existence
- Not addressed
  - Denial of service
  - Mis-configuration, validation of the content