DIMACS Security & Cryptography Crash Course – day 4

Internet Cryptography Tools, Part II: IP-Sec

Prof. Amir Herzberg

Computer Science Department, Bar Ilan University

http://amir.herzberg.name

Sources

- Partial but readable coverage in Stalling's book, `Cryptography and Network Security`
- IP-Sec is defined in Internet Engineering Task Force (**IETF**) RFC Documents:
  - Architecture – RFC 2401
  - Authentication Header (AH) – RFC 2402
  - Encapsulating Security Payload (ESP) – RFC 2406
  - IKE – Internet Key Exchange – RFC 2409

Outline

- Internet Layer Security
- IPsec Architecture
- The ESP Header
- The AH Header
- IPsec Modes of Operation
- IPsec and NAT
- IKE

Internet Layer Security

- **Characteristics**
  - Connectionless, Unreliable
  - IP addresses can be easily spoofed
  - Routers and gateway might be sniffed
- **Requirements**
  - Data source authentication
  - Integrity protection
  - Replay protection
  - Access control
  - Confidentiality and privacy
  - Clogging prevention (Availability)

IPsec - IP Security Protocol

Designed for IPv4 and IPv6

- Mandatory to implement in IPv6

Security Services

- Confidentiality
- Integrity
- Data source authentication (source IP address??)
- Replay protection
- Access Control

Can protect from…

- Syn attack
- Session hijacking

IPsec Protocols (RFC2401)

- Two **separate** layers
- IP-Sec Setup: IKE – Internet Key Exchange
  - Compare to SSL Handshake protocol; application layer
  - Negotiate and establish `Security Association`
  - Run once per `IP-sec connection` - not `real-time`
- IP-Sec sub-layer: traffic encapsulation & protection
  - Compare to SSL record protocol
  - Between IP and Transport layers
  - AH – Authentication Header (no secrecy)
  - ESP – Encapsulating Security Payload
  - Signal to IKE when detecting traffic that requires IP-sec but without established IP-sec connection

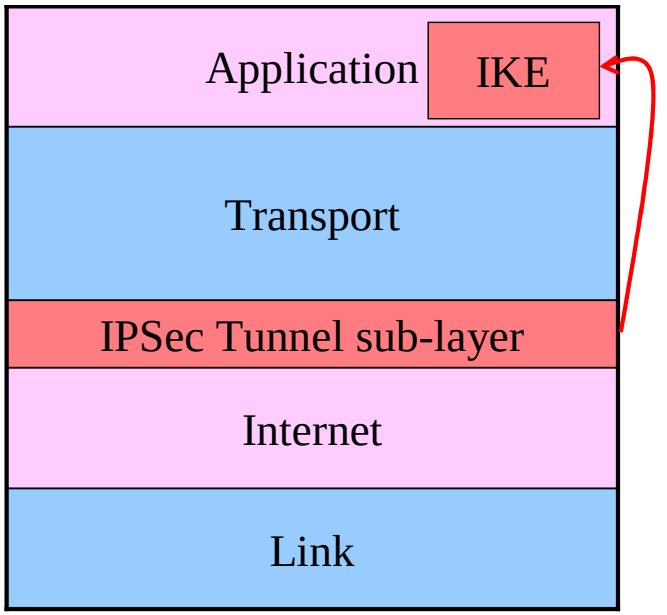
Adding Crypto-Security – Where?

Internet Layer – Secure IP?

Pros:

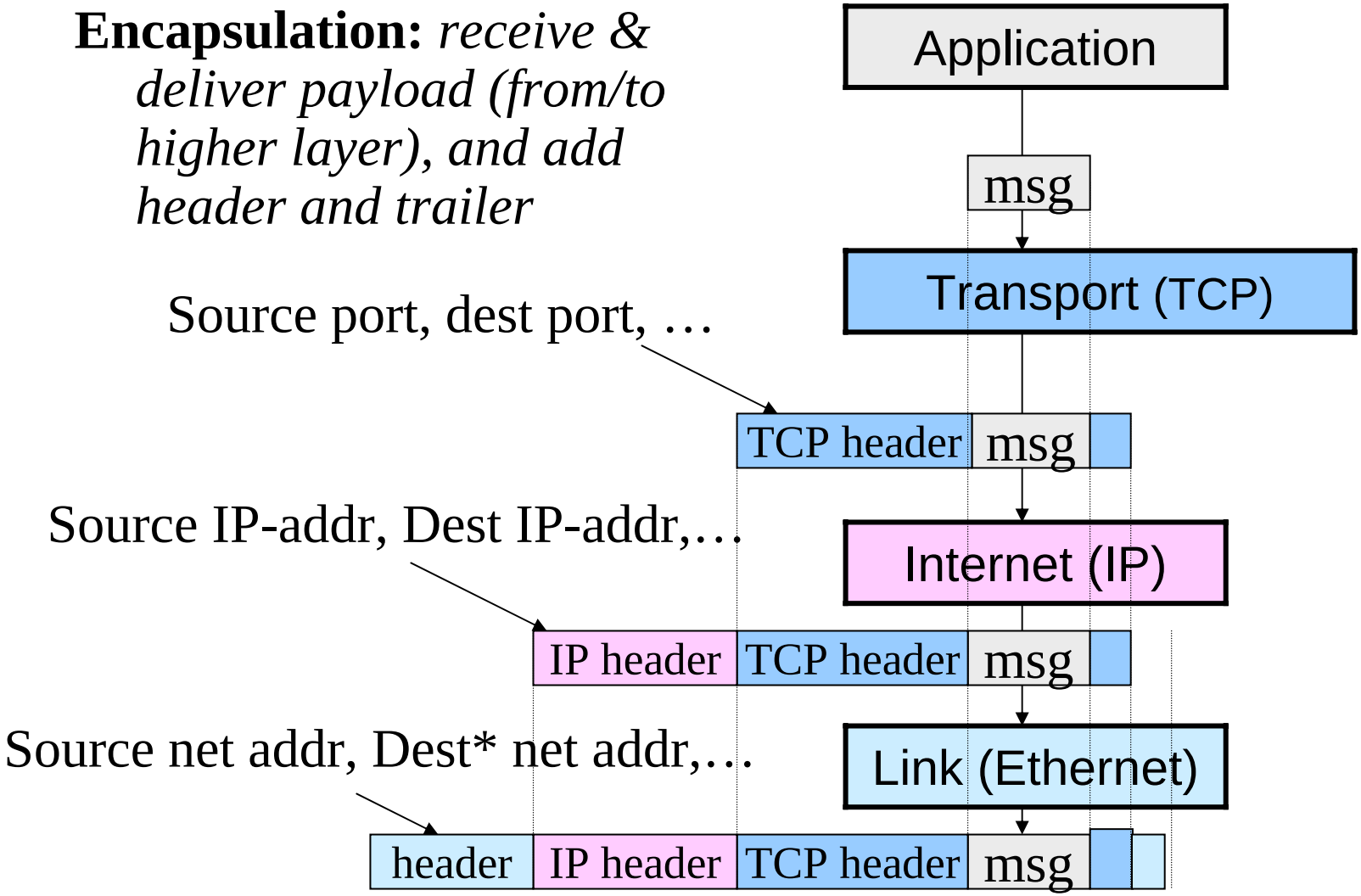
- Protection against DOS (clogging)
- Protect all applications, data
- Implemented by operating systems, Routers, …

- Cons:
  - Hard to implement
  - Rarely available at destination
    - Compatible algorithms
    - Key management

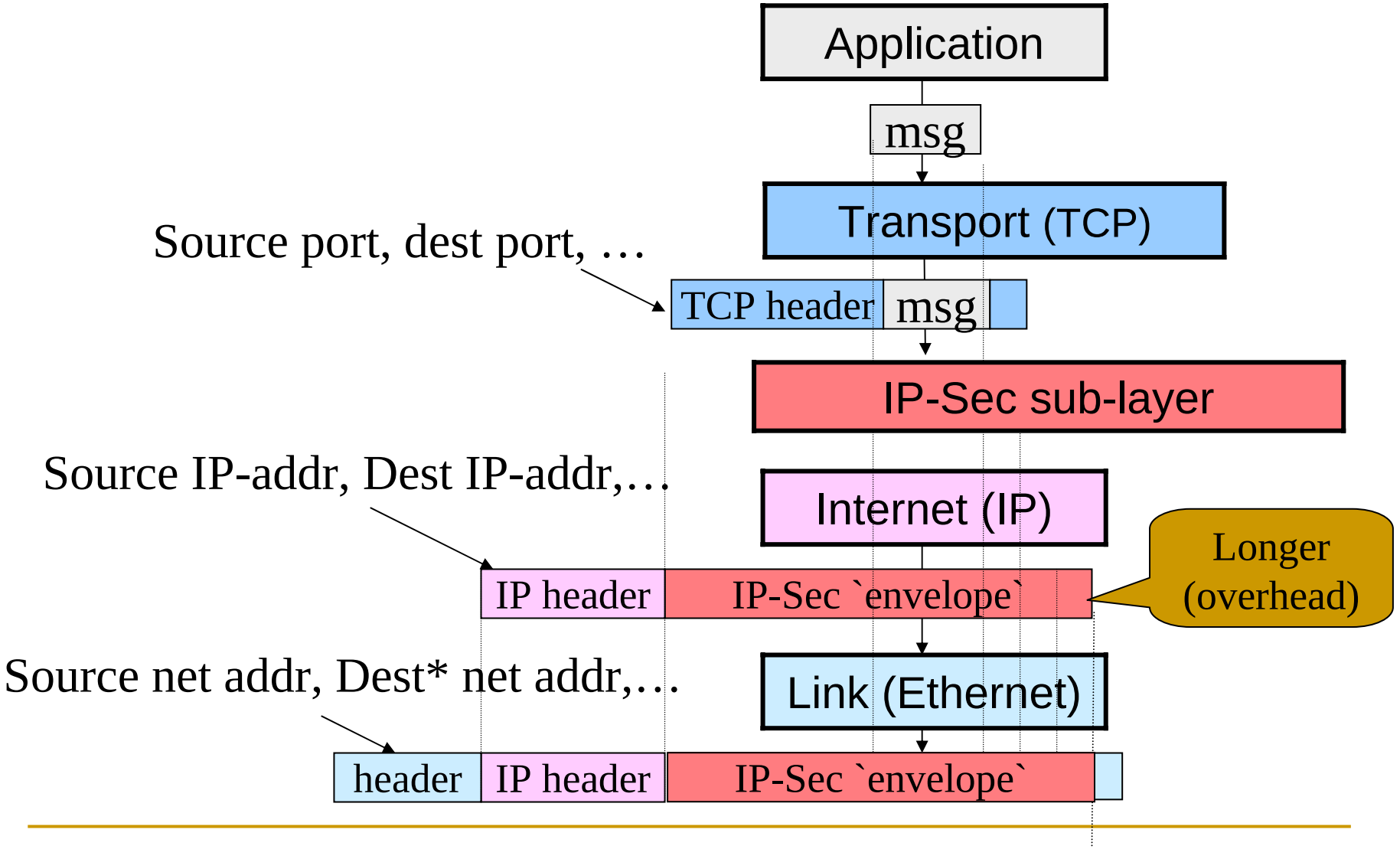


Standard TCP/IP Encapsulation

Encapsulation: *receive & deliver payload (from/to higher layer), and add header and trailer*



TCP/IP Encapsulation with IP-Sec



IP-Sec Implementation Options

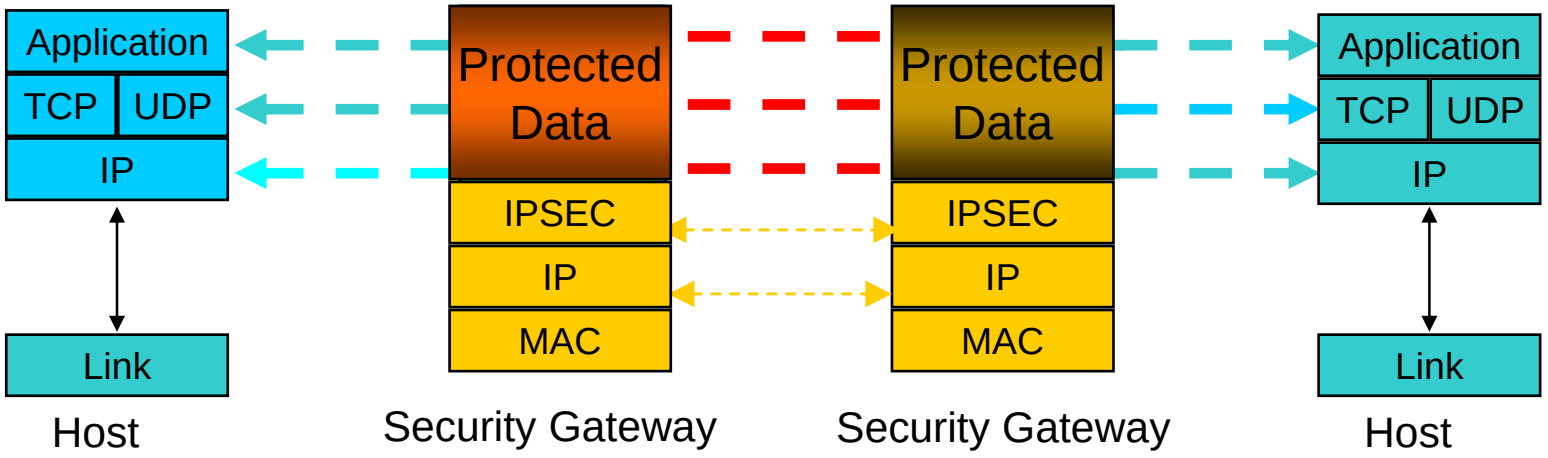
1. Native: implement IP-sec as part of IP implementation in Operating System
   - E.g. in Windows 2000, XP
2. BITS (Bump In The Stack) – intercept IP traffic to/from network driver
   - Implementations on host w/o changing OS
     - E.g. Checkpoint's firewall implementation
3. BITW (Bump In The Wire) – intercept IP traffic by tunneling via *security gateway*
   - Single gateway can protect multiple hosts
   - Only `tunnel mode` of IP-Sec…

IPsec Modes of Operation

- Tunnel Mode
  - IPsec adds its own IP header
  - IPsec encapsulation/decapsulation either by hosts or by gateways along the route
- Transport Mode
  - IPsec uses existing IP header, just changes protocol field to IP-Sec
  - End-to-end - IPsec encapsulation by source host, decapsulation by destination host (receiver)

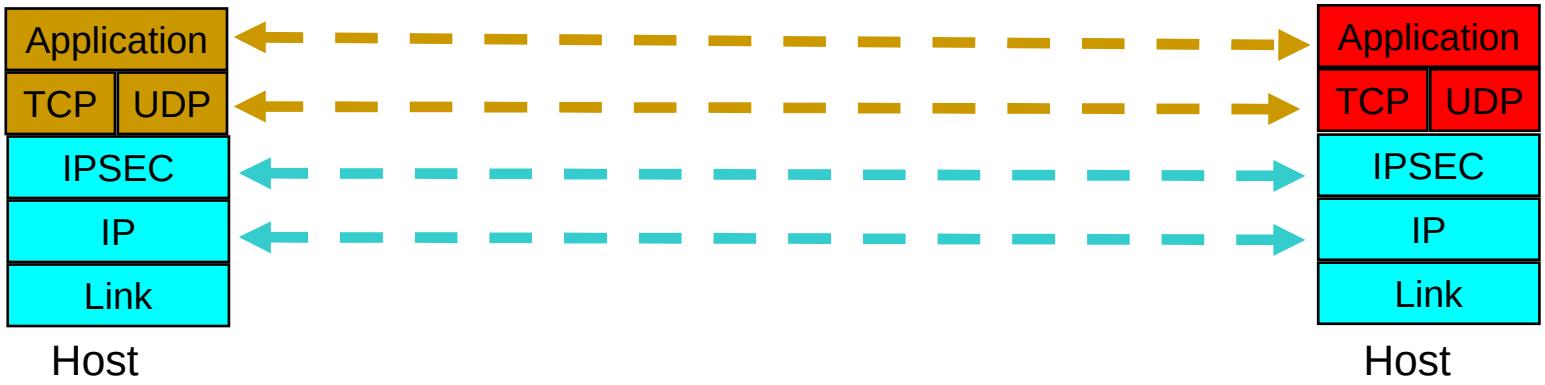
IPsec Tunnel Mode

- Can be applied by Security gateways
  - But also by hosts (at one or both ends)
  - Traffic may be IP-Sec protected already (nested)
- Entire IP packet is payload to IPSEC
- If provided by gateway, transparent to host
- If encryption used, hides hosts' IP address

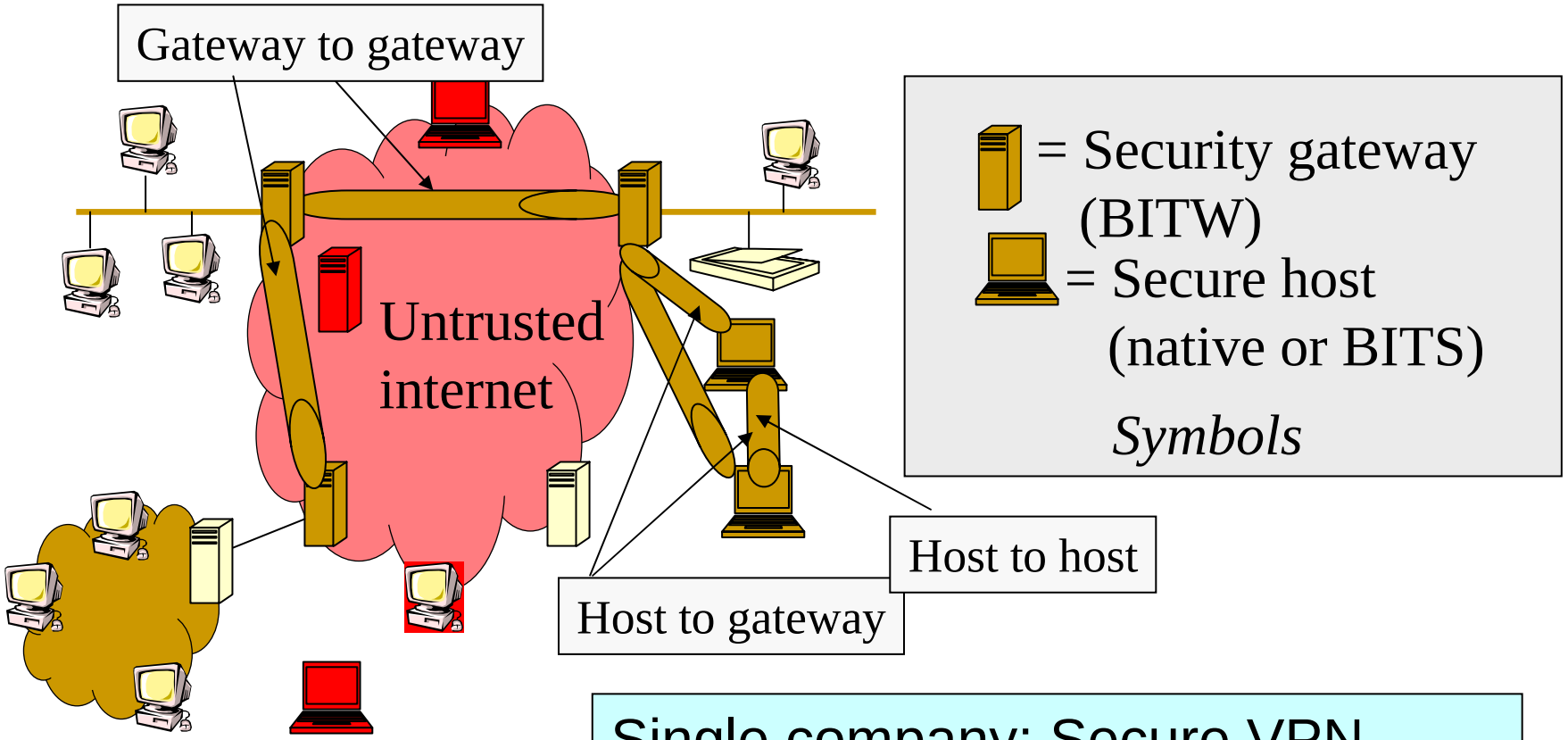


IPsec Transport Mode

- Supplies end to end security services
- Modifies IP Header, Payload
  - Does *not* add another header!
- Requires IP-Sec support by both hosts: native or (at least) BITS (bump in the stack)



IP-Sec transforms the Net to a Secure Virtual Private Network



Single company: Secure VPN
Many companies: Secure Extranet

IP-Sec Sub-Layer (Record) Protocols

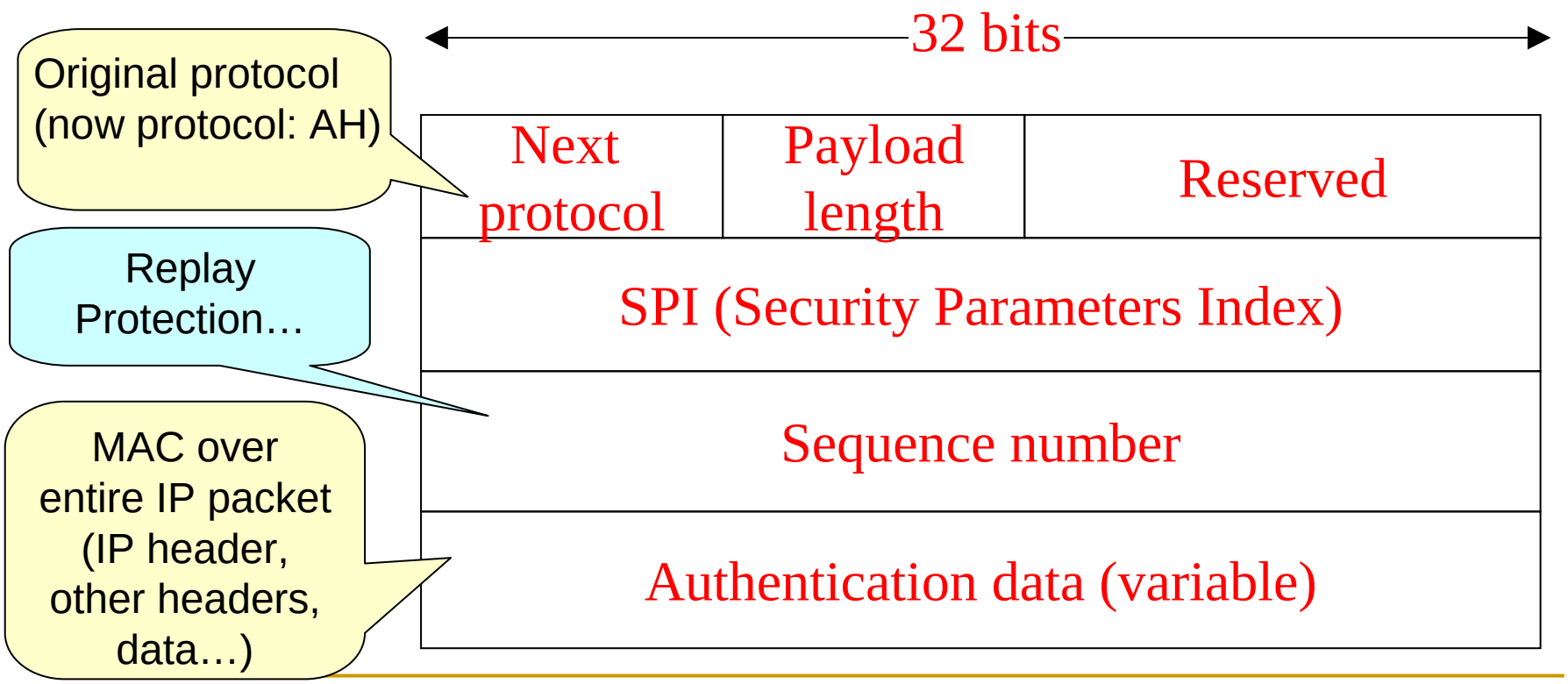
- AH – Authentication Header (no confidentiality)
- ESP – Encapsulating Secure Payload – encryption (can be `null`), authentication
- Do *not* use encryption w/o authentication [Be96]
- Both support multiple *security associations (SA)*
  - SA=Security parameters: keys, algorithms, counters…
  - Multiple SA btw same peers – different protocols/ports
    - Separate btw users to prevent chosen/known text attacks!
  - Simplifies key update – just change SA
  - Identify SA by Security Parameter Index (SPI)
    - Each party selects its SPI for each security association
    - Send recipient's SPI in each packet (32 bit)
- Both use sequence numbers for FIFO, no-replay

IPsec Replay Protection

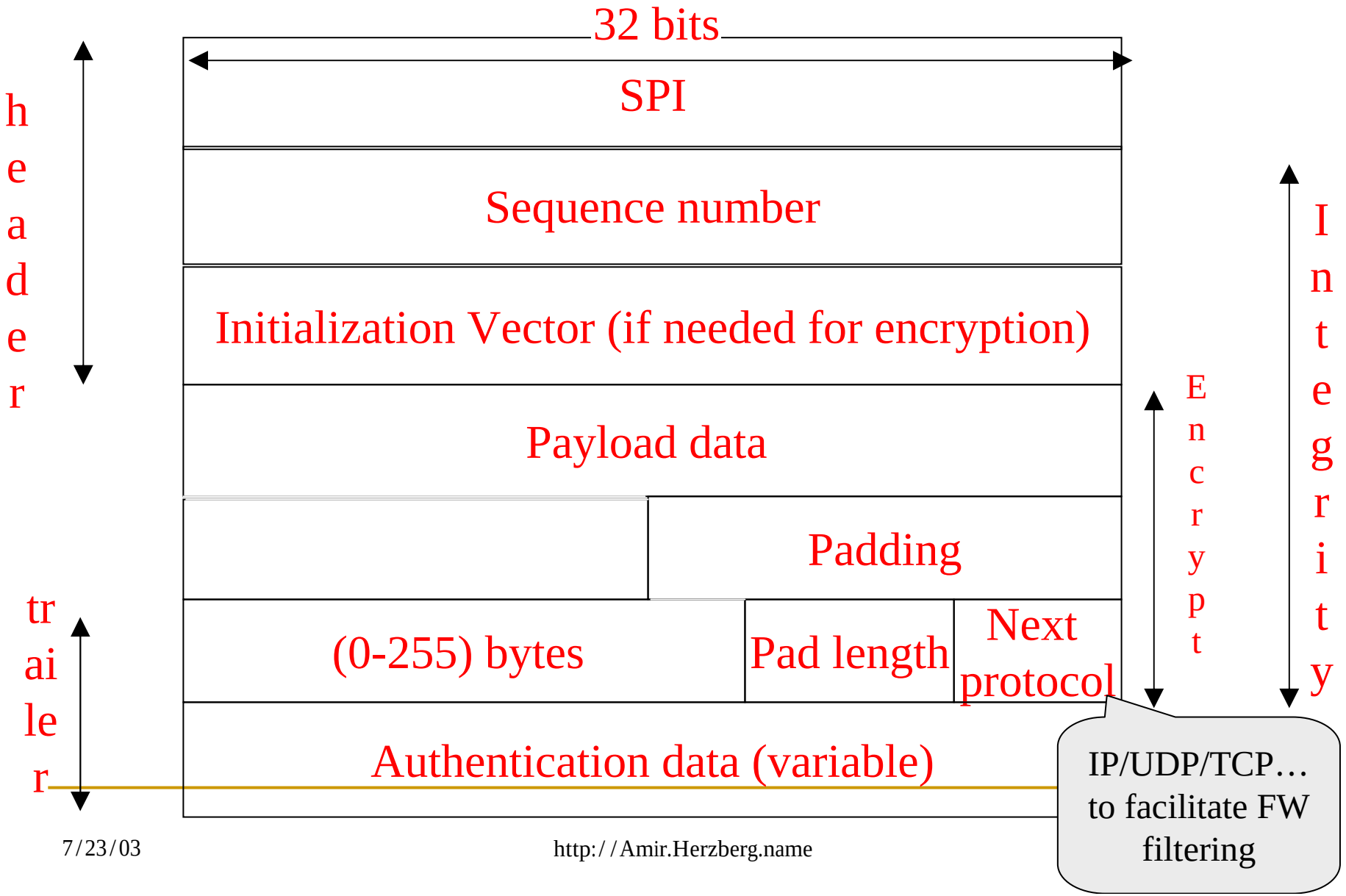
- Sequence number zero when SA established
- Increment per outgoing packet
- Receiver identify replay by repeated seq #
  - To avoid dropping legal packets arriving out of order, the receiver maintains in the SA a sliding window (minimal size 32)
- Sequence number field is sent and included in the MAC computation
- Must not wrap during a single key lifetime
  - Keys must be changed after 2^{32} packets

Authentication Header (AH)

- Inserted after the IP header
- AH protocol number (in IP header) is 51



ESP - Encapsulating Security Payload



IPsec Data Structures

- IPsec is using two data structures (define in the IPsec architecture RFC)
- SAD – Security Association Data contains all the active Security Associations (SAs)
  - Incoming: access via SPI in packet
    - SPI in each direction selected by recipient (for efficiency)
  - Outgoing: access via `selectors` in packet
    - IP addresses, TCP/UDP ports and more
  - Built manually or by key management (IKE)
- SPD – Security Policy Data contains user defined policy. The user defines which security services, at which level are offered to each IP datagram

The Security Policy Data (SPD)

Contains a list of rules: <select, action>

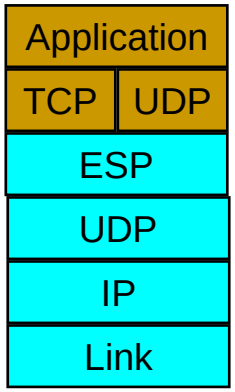
- Selectors: IP addresses (or range), TCP/UDP ports and more
- Actions:
  - Discard
  - Bypass IPsec
  - Apply IPsec, specifying either (or both):
    - Security services, protocol, and algorithms
    - Pointer to the entry of matching active SA in the SAD
- Packet-filtering (firewall) functionality

IPsec and NAT Incompatibilities

1. AH MAC calculation includes IP header, changed by NAT ➔MAC verification fails
2. UDP/TCP checksum: TCP and UDP checksum cover the IP addresses; NAT devices recalculate checksum, but can't after IP-Sec ➔ receiving IP stack drop the packet
   - The problem doesn't occur in Tunnel mode, because only the outer IP header can be modified, while the TCP/UDP checksum is calculated over the inner (encrypted) IP header

UDP Encapsulation of IPsec

- UDP encapsulation of IPsec packets solves the Checksum problem



- The UDP ports are selected by the key management protocol
- Common solution to NAT interoperability

IP-Sec Setup: IKE (Internet Key Exchange)

- Two phases
- 1st phase: setup ISAKMP SA(**I**nternet **S**ecurity **A**ssociation and **K**ey **M**anagement **P**rotocol)
  - Algorithms, keys, etc. – to be used by IKE (not AH/ESP!)
  - Perfect forward secrecy (PFS): exposure of all keys does not expose past traffic [using Diffie-Hellman]
- 2nd phase: Generate IP-Sec SA
  - Protected using the ISAKMP SA
  - Many 2nd phases may share ISAKMP SA (1st phase)
    - E.g. one 1st phase for gateways, then many 2nd phase for each pair of hosts using these gateways
  - More efficient than 1st phase; PFS optional

Why Two IKE Phases?

- To fulfill the PFS requirement, every phase I exchange, performs a DH exchange
- In Phase II, DH execution is optional – phase II and the IPsec keys can be derived from phase I exchange
- Phase II is more efficient; many two phase II exchanges can use the same set of phase I keys

Identification of Peer in IKE

- 1st phase – identify using one of:
  - Manually pre-shared secret key
  - Exchanged public key certificates, and…
    - Public key signatures, or
    - Public key encryption of challenge (two variants)
- 2nd phase – trusts identification in 1st phase
  - Uses identities from SPD

Why derive many session keys?

- Why not establish & use one `master key`?
- Ensure reliable separation of sessions
- Restrict use of a single key
  - Make cryptoanalysis harder – less available ciphertext
  - Restrict damage of *known key attack*: session key exposure does not expose past or future messages, session keys, or master key
  - *Forward secrecy* – exposing *all* keys now does not expose past session keys
  - *Proactive secrecy* – security *recovered* after *all* keys exposed

Conclusion

- IP Security protocol protects all Internet traffic
- Tunnel mode allows gateways to protect many hosts
- Transport mode allows efficient host-to-host security
  - Possible interoperability problems w/ NAT (can tunnel over UDP)
- Support for authentication (AH/ESP) and encryption (ESP)
- Tunneling of IPsec protected traffic is possible
  - E.g. to hide identities of source/destination hosts behind gateway
- Flexible policy for security
  - Block, allow (unprotected) or protect traffic
  - Defined on host/port basis – not per user!
- Resiliency to clogging (in IKE via cookies)
- Requires no change in applications
- Hard to implement, interoperate

Extras

IKE DOS Protection

- Goal : protect against a DoS where the attacker perform IP spoofing. The attacker floods the victim with IKE requests, and forces him to perform expensive computations
- Solution : before performing expensive computations (e.g. DH), verify that the other party is indeed located in the IP address that appears in the header
- Note: requires the `main mode` of IKE (6 flows, cf. to `aggressive mode of 3 flows)

The Cookies Mechanism

- Any participant in the protocol sends a pseudo random string (Cookie) to the other party
- The other party return the cookie, proving it can receive from its IP address
- Compute cookie by hash (e.g. MD5) of IP addr, UDP ports, local secret value, date and hour
- Efficient generation, memory less verification
- Expensive calculations will be performed only after the other party cookie is received

Exercises

- An organization connects to the Internet from multiple offices, but concerned about:
  - Denial of service attacks from the Internet
  - Protect data on few key applications (mostly web) from unauthorized exposure
  - Efficiency and cost of solution
- IP-Sec uses connections. In what ways are these connections not reliable?
- Consider extranet btw 3 companies, using mostly web services. Present and compare SSL and IPSec designs.