L10: Networks and Cryptography

Hui Chen, Ph.D.
Dept. of Engineering & Computer Science
Virginia State University
Petersburg, VA 23806

Acknowledgement

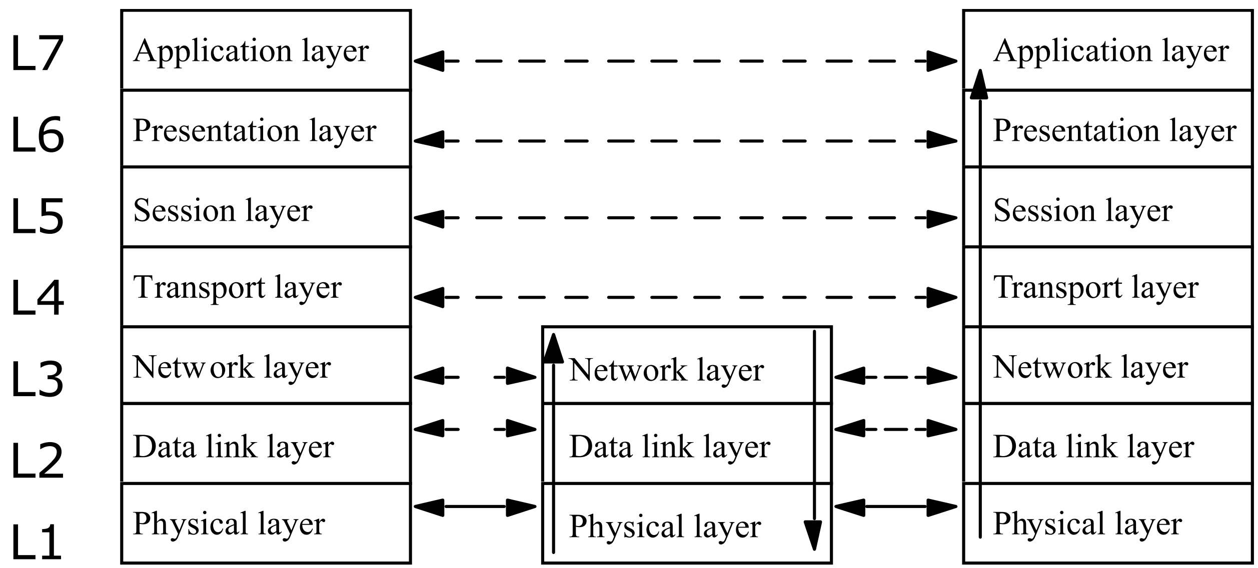
- Many slides are from or are revised from the slides of the author of the textbook
  - Matt Bishop, Introduction to Computer Security, Addison-Wesley Professional, October, 2004, ISBN-13: 978-0-321-24774-5. Introduction to Computer Security @ VSU's Safari Book Online subscription
  - http://nob.cs.ucdavis.edu/book/book-intro/slides/

Outline

- ISO/OSI 7-layer model
- Link and End-to-End protocols
- Concept of traffic analysis
- Two example protocols
  - PEM
  - IPSec

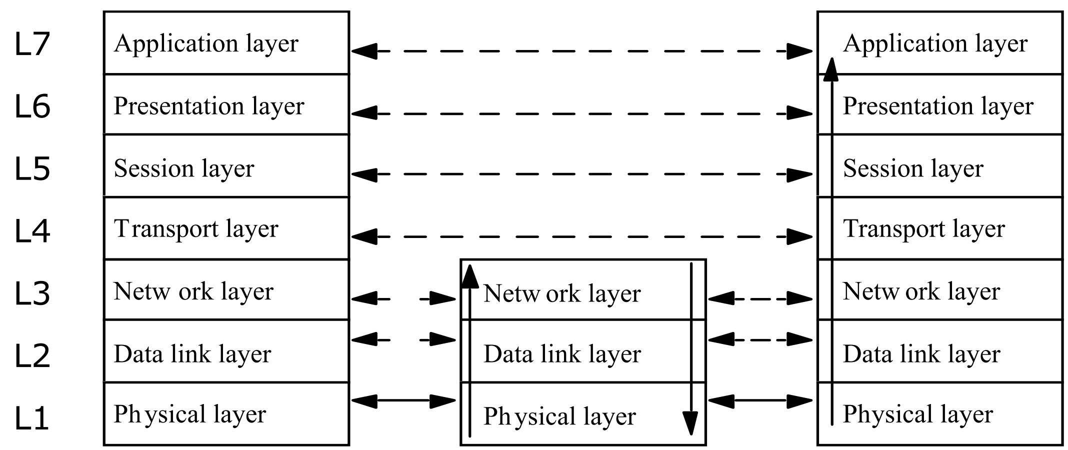
ISO/OSI Model

- *Conceptual* model for for digital communications and computer networks



ISO/OSI Model: Concepts

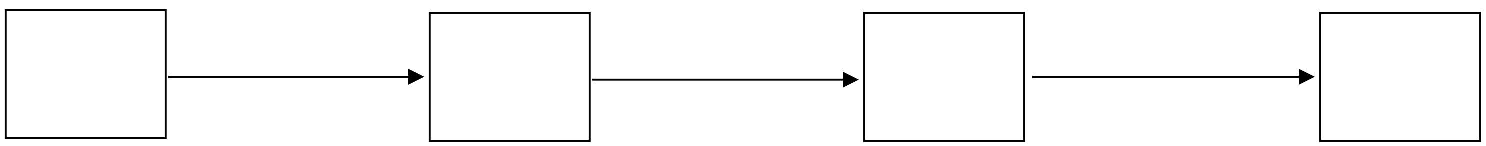
- Each host has a principal at each layer
- Principals at the same layer of different hosts are peers
- Peers communicate with peers at same layer
- Layer 1, 2, and 3 principals interact with peers at neighboring hosts (directly connected hosts)
- Layer 4, 5, 6, and 7 principals interact only with similar principals at the other end of the communication
- Use host to refer to the appropriate principal in the discussion that follows



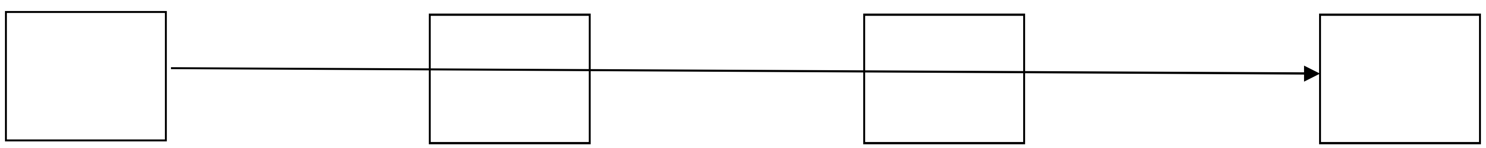
Link and End-to-End Protocols

- Hosts: C_0 … C_n and C_i and C_{i+1} are directly connected
- Link Protocol: C_j and C_{j+1} as comm. end points
- End-to-End Protocol: C_0 and C_n as comm. end points

Link Protocol



End-to-End (or E2E) Protocol



Encryption

- Link encryption
  - Each host enciphers message so host at “next hop” can read it
  - Message can be read at intermediate hosts
- End-to-end encryption
  - Host enciphers message so host at other end of communication can read it
  - Message cannot be read at intermediate hosts

Examples

- Secure Shell (SSH) protocol
  - Messages between client and server enciphered
  - Encipherment and decipherment occur only at these hosts
  - End-to-end protocol
- PPP Encryption Control Protocol
  - Host gets message, deciphers it
    - Figures out where to forward it
    - Enciphers it in appropriate key and forwards it
  - Link protocol

Cryptographic Considerations

- Link encryption
  - Each host shares key with neighbor
  - Can be set on per-host or per-host-pair basis
    - Consider 3 hosts, h1, h2, and h3
    - Per-host: each host has own keys, 3 keys
    - Per-host-pair: one key for a possible pair, one key for (h1, h2); one for (h2, h3), and one for (h1, h3)
- End-to-end
  - Each host shares key with destination
  - Can be set on per-host or per-host-pair basis
  - Message cannot be read at intermediate nodes

Traffic Analysis

- Deduce information from metadata (e.g., sender and recipient)
- Link encryption
  - Can protect headers of packets
  - Possible to hide source and destination
    - Note: may be able to deduce this from traffic flows
- End-to-end encryption
  - Cannot hide packet headers
    - Intermediate nodes need to route packet
  - Attacker can read source, destination

Traffic Analysis: Example

- All traffic are enciphered using end-to-end encryption in a company that has leaked proprietary data.
- Investigator Alice monitors senders and recipients of network traffic.
  - Connection from host *larry* always occur between midnight and four in the morning
  - In correlation with the time the leak occurred, Alice suggests that host *larry* is likely involved in the leak.
- Alice has not read any enciphered data in the network, only the metadata (in the clear)

Example Protocols

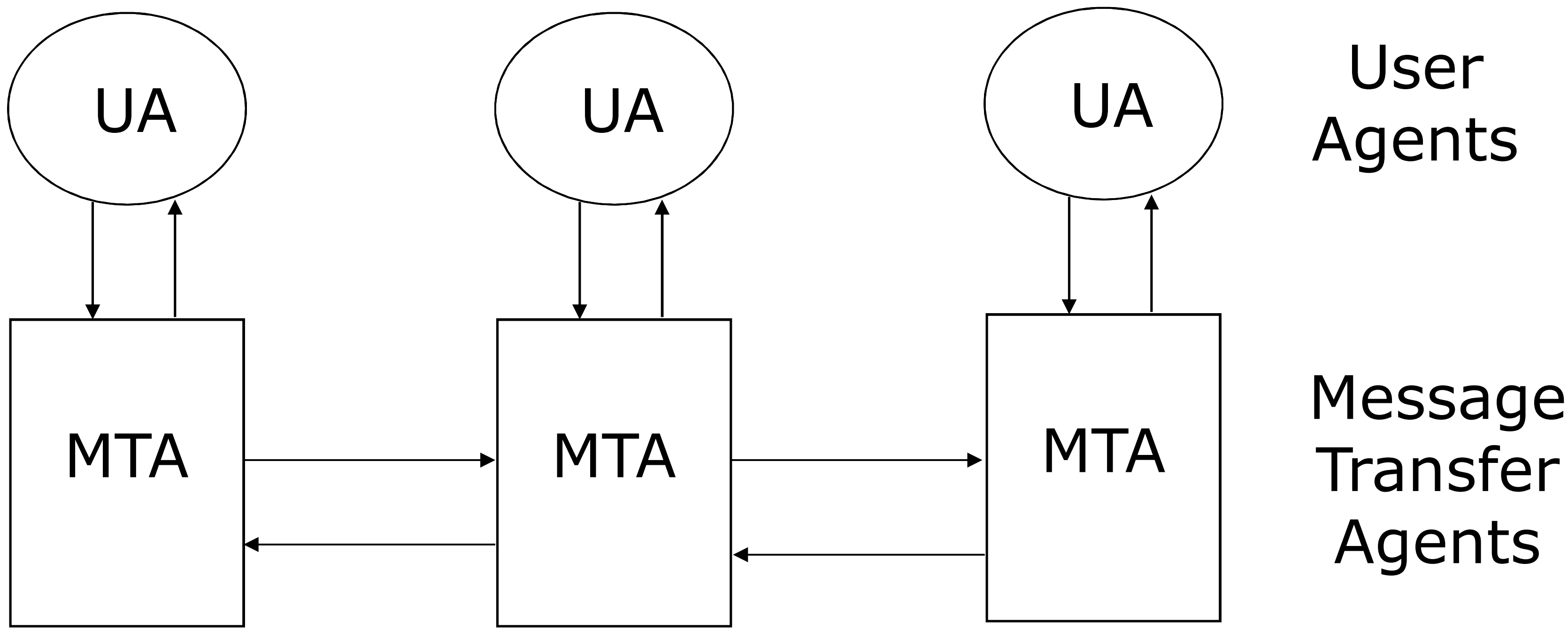
- Privacy-Enhanced Electronic Mail (PEM)
  - Applications layer protocol
- IP Security (IPSec)
  - Network layer protocol

Privacy-Enhanced Electronic Mail (PEM)

- Overview of E-mail service
- Threats to E-mail service
- Design goals of PEM
- Design for confidentiality
- Design for integrity and authentication
- Design for non-repudiation
- Practical considerations

Message Handling System

- Authentication is minimal and easily evaded



Threats to E-mail Services

- Violation of confidentiality
- Violation of Authentication
- Violation of message integrity
- Violation of non-repudiation

Goals of PEM

- To enhance E-mail service with
  - Confidentiality
    - Only sender and recipient(s) can read message
  - Origin authentication
    - Identify the sender precisely
  - Data integrity
    - Any changes in message are easy to detect
  - Non-repudiation of origin
    - Whenever possible …

Design Principles

- Do not change related existing protocols
  - Cannot alter SMTP
- Do not change existing software
  - Need compatibility with existing software
- Make use of PEM optional
  - Available if desired, but email still works without them
  - Some recipients may use it, others not
- Enable communication without prearrangement
  - Out-of-bands authentication and key exchange are problematic

Basic Design: Keys

- Two keys
  - *Interchange keys* tied to sender and recipients and are static (for some set of messages)
    - Must be available *before* messages sent
    - If symmetric ciphers are used, the keys must be exchanged out-of-bands
    - If public keys are used, the sender needs to obtain the certificate of the recipient
  - *Data exchange keys* generated for each message
    - Like a session key, session being the message

Basic Design: Confidentiality

- Confidentiality
  - m message
  - k_s data exchange key
  - k_B Bob's interchange key

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

Basic Design: Integrity

- Integrity and authentication:
  - m message
  - $h(m)$ hash of message m —Message Integrity Check (MIC)
  - k_A Alice's interchange key
- Non-repudiation: if k_A is Alice's interchange key, this establishes that Alice's interchange key was used to sign the message

$$\text{Alice} \xrightarrow{m \{ h(m) \}_{k_A}} \text{Bob}$$

Basic Design: Putting Together

- Confidentiality, integrity, authentication:
  - Notations as in previous slides

$$\text{Alice} \xrightarrow{\{\, m \,\}_{k_s} \,||\, \{\, h(m) \,\}_{k_A} \,||\, \{\, k_s \,\}_{k_B}} \text{Bob}$$

Design Goal: Non-Repudiation

- Non-Repudiation
  - Notations as in previous slides
  - If a public key cipher is bing used and k_A is Alice's private key, get non-repudiation

$$\text{Alice} \xrightarrow{\{\ m\ \}_{k_s} \ ||\ \{\ h(m)\ \}_{k_A} \ ||\ \{\ k_s\ \}_{k_B}} \text{Bob}$$

Practical Considerations

- Limits of SMTP
  - Only ASCII characters, limited length lines
- Use encoding procedure
  1. Map local character representation into canonical format
     - Format meets SMTP requirements
  2. Compute and encipher MIC over the canonical format; encipher message if needed
  3. Map each 6 bits of result into a character; insert newline after every 64th character
  4. Add delimiters around this ASCII message

Problem

- Recipient without PEM-compliant software cannot read it
  - If only integrity and authentication used, should be able to read it
- Mode MIC-CLEAR allows this
  - Skip step 3 in encoding procedure
  - Problem: some MTAs add blank lines, delete trailing white space, or change end of line character
  - Result: PEM-compliant software reports integrity failure

PEM vs. PGP

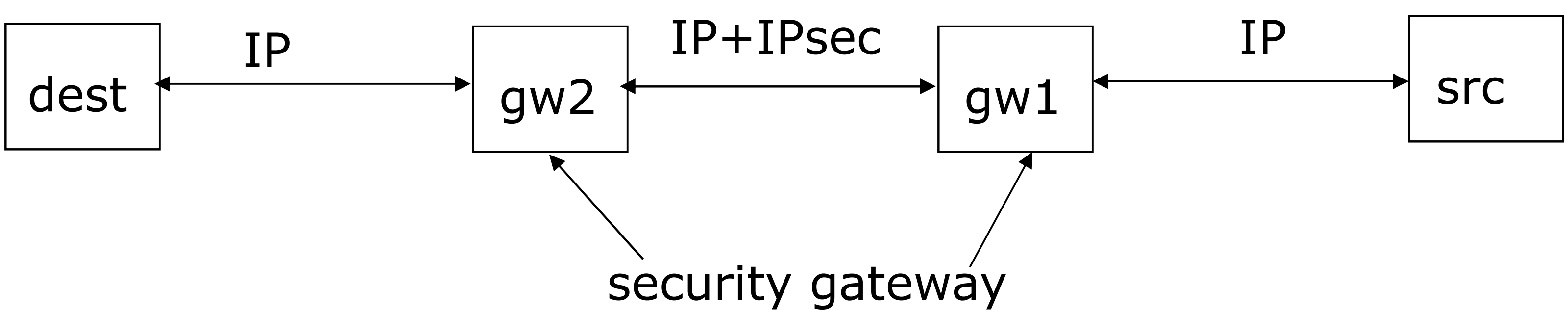
- Use different ciphers
  - PGP uses IDEA cipher
  - PEM uses DES in CBC mode
- Use different certificate models
  - PGP uses general “web of trust”
  - PEM uses hierarchical certification structure
- Handle end of line differently
  - PGP remaps end of line if message tagged “text”, but leaves them alone if message tagged “binary”
  - PEM always remaps end of line

IPsec

- Design goals
- Transport mode and tunnel mode
- IPsec architectures
- IPsec protocols

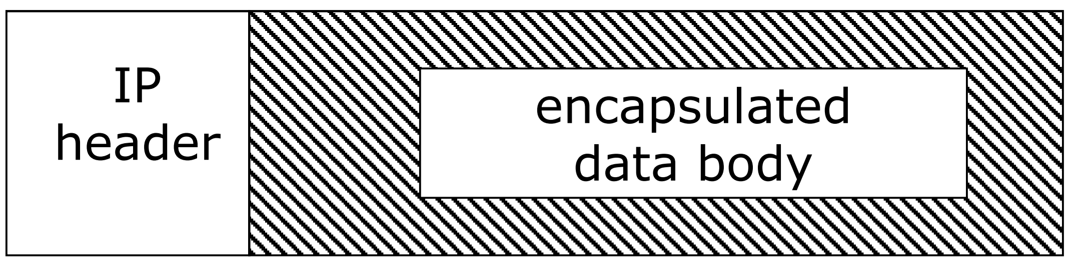
Design Goals

- Network layer security
  - Provides confidentiality, integrity, authentication of endpoints, replay detection
- Protects all messages sent along a path



IPsec Transport Mode

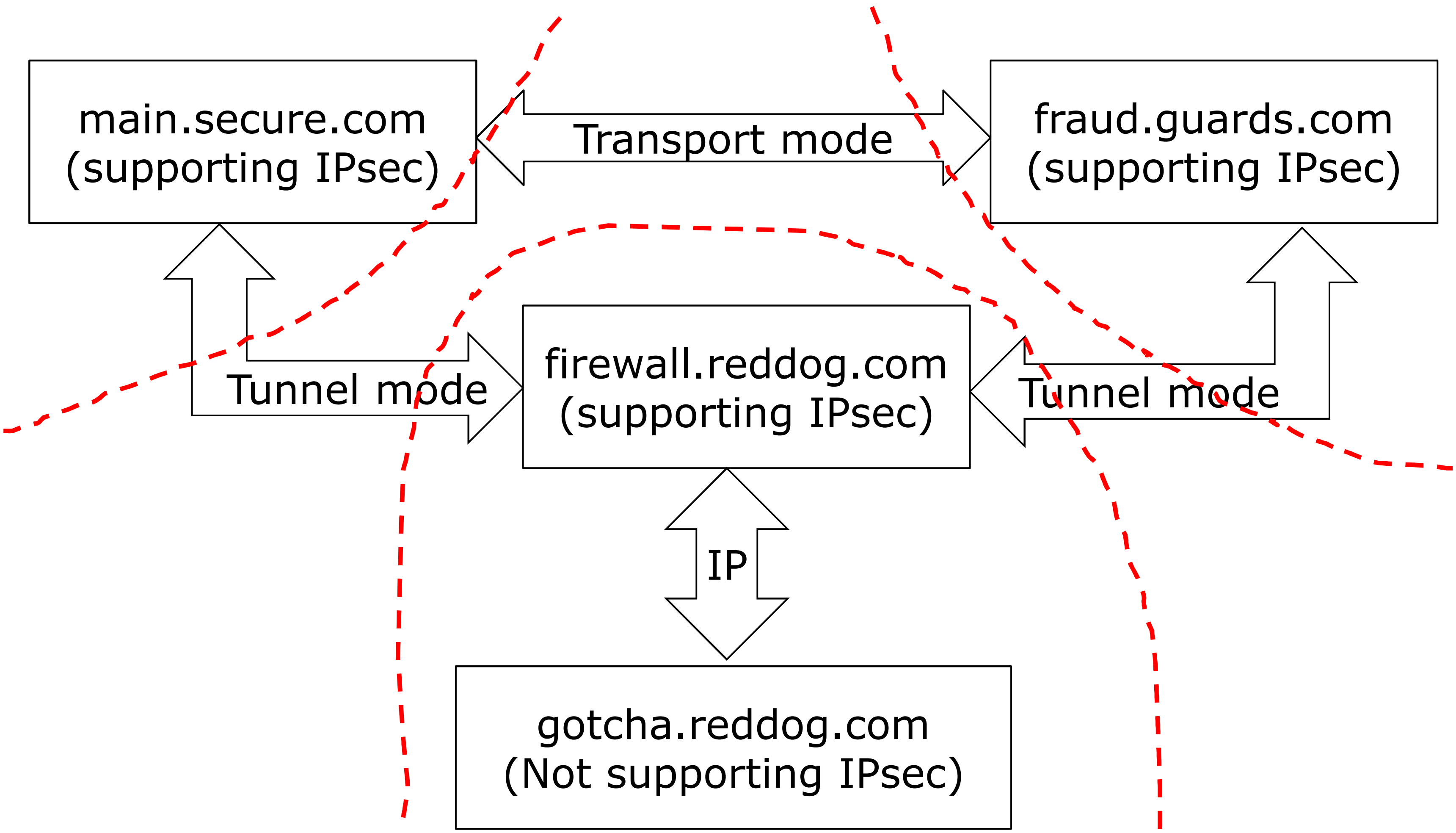
- Encapsulate IP packet data area (containing upper layer packet, e.g., TCP segments) to form IPsec-wrapped data packet
- Use IP to send IPsec-wrapped data packet
- Note: IP header not protected
- Used when both endpoints support IPsec



IPsec Tunnel Mode

- IP header not protected in IP transport mode
- Protect IP header using IP tunnel mode, i.e., encapsulate entire IP packet in an IPsec envelope and forward it using IP
- Used when either or both endpoints do not support IPsec but two intermediate nodes do

IPsec: Example Scenario



IPsec Protocols

- Authentication Header (AH) protocol
  - Message integrity
  - Origin authentication
  - Anti-replay
- Encapsulating Security Payload (ESP) protocol
  - Confidentiality
  - Others provided by AH
- Internet Key Exchange (IKE) protocol
  - Key management

IPsec Architecture: SPD

- Security Policy Database (SPD)
  - Determine how to handle messages (discard them, add security services, forward message unchanged)
  - SPD associated with network interface
  - SPD determines appropriate entry from packet attributes
    - Including source, destination, transport protocol

SPD: Example

- Goals
  - Discard SMTP packets from host 192.168.2.9
  - Forward packets from 192.168.19.7 without change
- SPD entries

```
src 192.168.2.9, dest 10.1.2.3 to 10.1.2.103, port 25, discard
src 192.168.19.7, dest 10.1.2.3 to 10.1.2.103, port 25, bypass
dest 10.1.2.3 to 10.1.2.103, port 25, apply IPsec
```

- Note: entries scanned in order
  - If no match for packet, it is discarded

IPsec Architecture: SA

- Security Association (SA)
  - Association between peers for security services
    - Identified uniquely by destination address, security protocol (AH or ESP), unique 32-bit number (security parameter index, or SPI)
  - Unidirectional
    - Can apply different services in either direction
  - SA uses either ESP or AH, but not both. If both required, use 2 SAs

SA Database (SAD)

- Entry describes SA; some fields for all packets:
  - AH algorithm identifier, keys
    - When SA uses AH
  - ESP encipherment algorithm identifier, keys
    - When SA uses confidentiality from ESP
  - ESP authentication algorithm identifier, keys
    - When SA uses authentication, integrity from ESP
  - SA lifetime (time for deletion or max byte count)
  - IPsec mode (tunnel, transport, either)

SAD Fields

- Antireplay (inbound only)
  - When SA uses antireplay feature
- Sequence number counter (outbound only)
  - Generates AH or ESP sequence number
- Sequence counter overflow field
  - Stops traffic over this SA if sequence counter overflows
- Aging variables
  - Used to detect time-outs

IPsec Architecture

- Packet arrives
- Look in SPD
  - Find appropriate entry
  - Get dest address, security protocol, SPI
- Find associated SA in SAD
  - Use dest address, security protocol, SPI
  - Apply security services in SA (if any)

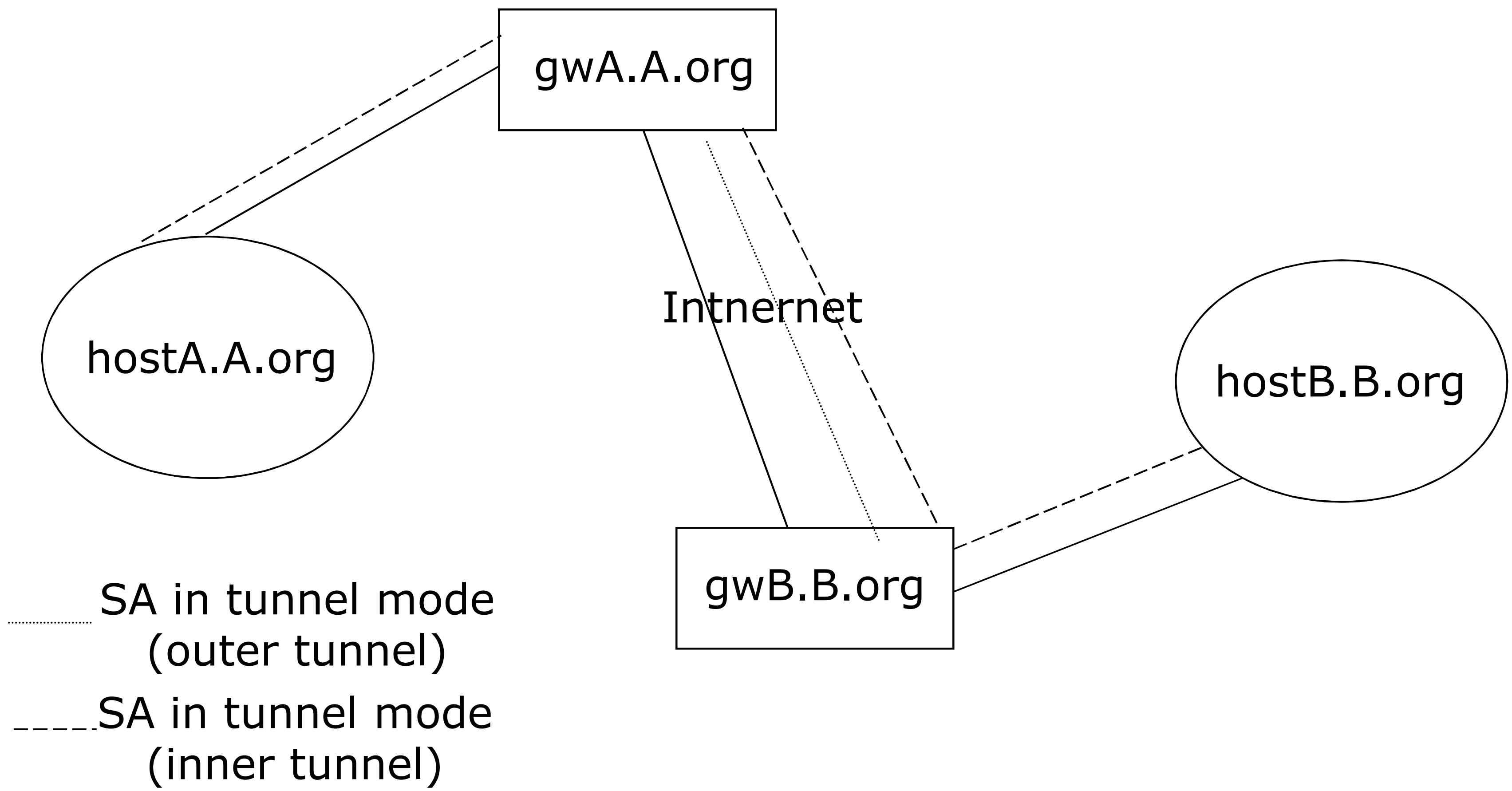
SA Bundles and Nesting

- Sequence of SAs that IPsec applies to packets
  - This is a *SA bundle*
- Nest tunnel mode SAs
  - This is *iterated tunneling*

Example: Nested Tunnels

- Group in A.org needs to communicate with group in B.org
- Gateways of A, B use IPsec mechanisms
  - But the information must be secret to everyone except the two groups, even secret from other people in A.org and B.org
- Inner tunnel: a SA between the hosts of the two groups
- Outer tunnel: the SA between the two gateways

Example: Systems



Example: Packets

- Packet generated on hostA
- Encapsulated by hostA's IPsec mechanisms
- Again encapsulated by gwA's IPsec mechanisms
  - Above diagram shows headers, but as you go left, everything to the right would be enciphered and authenticated, *etc*.

IP header from gwA	AH header from gwA	ESP header from gwA	IP header from hostA	AH header from hostA	ESP header from hostA	IP header from hostA	Transport layer headers, data

AH Protocol

- Parameters in AH header
  - Length of header
  - SPI of SA applying protocol
  - Sequence number (anti-replay)
  - Integrity value check
- Two steps
  - Check that replay is not occurring
  - Check authentication data

Sender

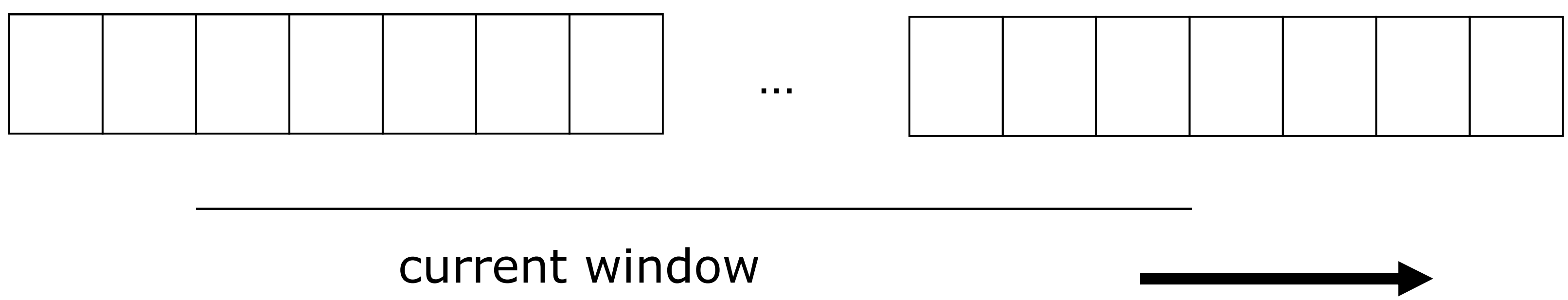
- Check sequence number will not cycle
- Increment sequence number
- Compute IVC of packet
  - Includes IP header, AH header, packet data
    - IP header: include all fields that will not change in transit; assume all others are 0
    - AH header: authentication data field set to 0 for this
    - Packet data includes encapsulated data, higher level protocol data

Recipient

- Assume AH header found
- Get SPI, destination address
- Find associated SA in SAD
  - If no associated SA, discard packet
- If antireplay not used
  - Verify IVC is correct
    - If not, discard

Recipient, Using Antireplay

- Check packet beyond low end of sliding window
- Check IVC of packet
- Check packet’s slot not occupied
  - If any of these is false, discard packet



AH Miscellany

- All implementations must support:
  HMAC_MD5
  HMAC_SHA-1
- May support other algorithms

ESP Protocol

- Parameters in ESP header
  - SPI of SA applying protocol
  - Sequence number (anti-replay)
  - Generic "payload data" field
  - Padding and length of padding
    - Contents depends on ESP services enabled; may be an initialization vector for a chaining cipher, for example
    - Used also to pad packet to length required by cipher
  - Optional authentication data field

Sender

- Add ESP header
  - Includes whatever padding needed
- Encipher result
  - Do not encipher SPI, sequence numbers
- If authentication desired, compute as for AH protocol *except* over ESP header, payload and *not* encapsulating IP header

Recipient

- Assume ESP header found
- Get SPI, destination address
- Find associated SA in SAD
  - If no associated SA, discard packet
- If authentication used
  - Do IVC, antireplay verification as for AH
    - Only ESP, payload are considered; *not* IP header
    - Note authentication data inserted after encipherment, so no deciphering need be done

Recipient

- If confidentiality used
  - Decipher enciphered portion of ESP heaser
  - Process padding
  - Decipher payload
  - If SA is transport mode, IP header and payload treated as original IP packet
  - If SA is tunnel mode, payload is an encapsulated IP packet and so is treated as original IP packet

ESP Miscellany

- Must use at least one of confidentiality, authentication services
- Synchronization material must be in payload
  - Packets may not arrive in order, so if not, packets following a missing packet may not be decipherable
- Implementations of ESP assume classical cryptosystem
  - Implementations of public key systems usually far slower than implementations of classical systems
  - Not required

More ESP Miscellany

- All implementations must support (encipherment algorithms):
  - DES in CBC mode
  - NULL algorithm (identity; no encipherment)
- All implementations must support (integrity algorithms):
  - HMAC_MD5
  - HMAC_SHA-1
  - NULL algorithm (no MAC computed)
- Both cannot be NULL at the same time

Which to Use: PEM, IPsec

- What do the security services apply to?
  - If applicable to one application *and* application layer mechanisms available, use that
    - PEM for electronic mail
  - If more generic services needed, look to lower layers
    - IPsec for network layer, either end-to-end or link mechanisms, for connectionless channels as well as connections
  - If endpoint is host, IPsec sufficient; if endpoint is user, application layer mechanism such as PEM needed

Key Points

- Key management critical to effective use of cryptosystems
  - Different levels of keys (session *vs*. interchange)
- Keys need infrastructure to identify holders, allow revoking
  - Key escrowing complicates infrastructure
- Digital signatures provide integrity of origin and content

  Much easier with public key cryptosystems than with classical cryptosystems

Summary

- ISO/OSI 7-layer model
- Link and End-to-End protocols
- Concept of traffic analysis
- PEM
- IPSec