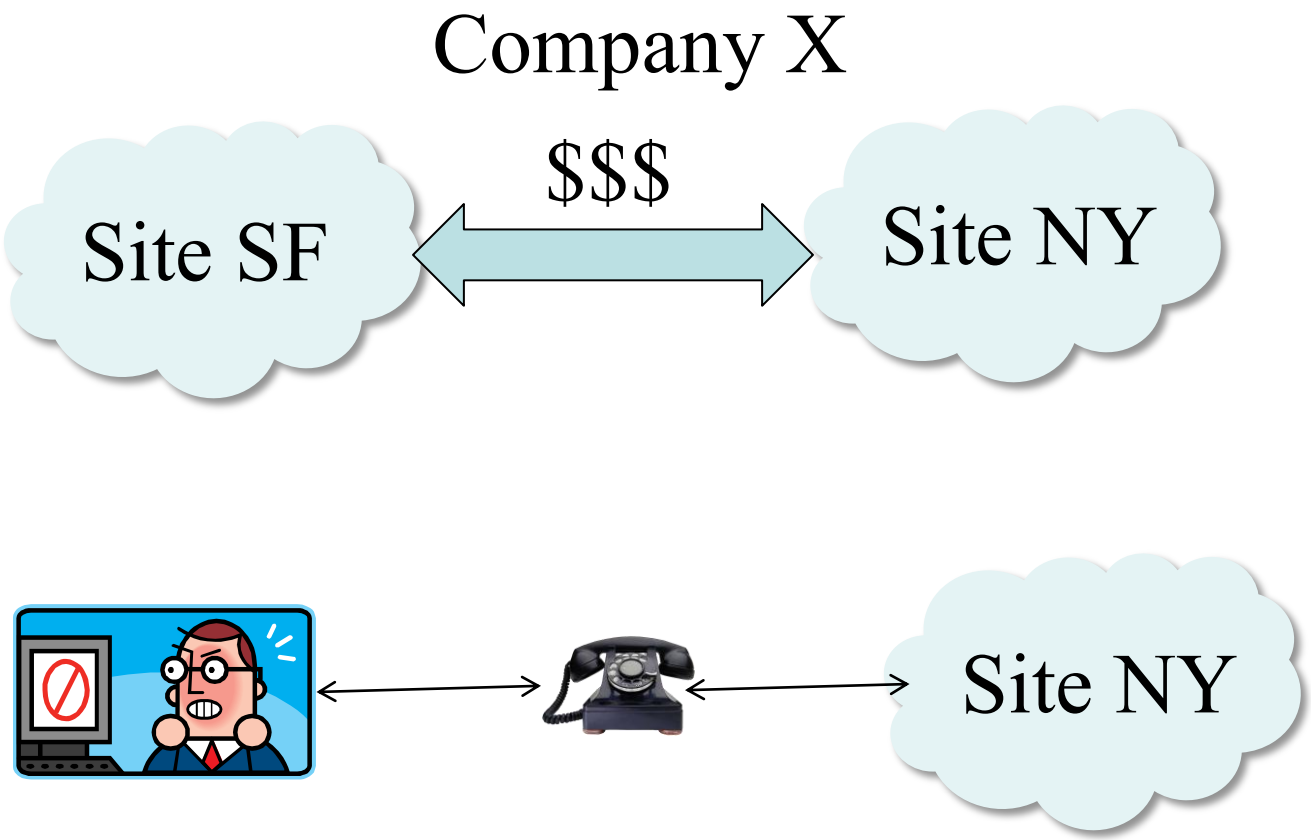
CSE509: (Intro to) Systems Security

Fall 2012

Invited Lecture by Vyas Sekar

IPSec

Security in Real Life: Motivation

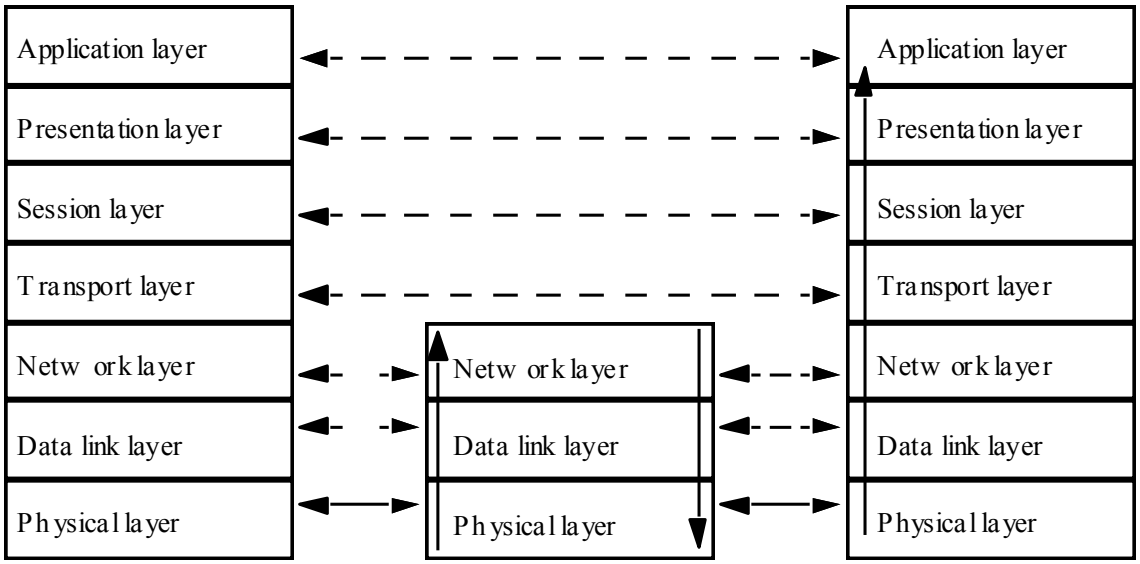


Security in Real Life: Why don't you run it on IP?

IP provides no security!
→Spoofing
→Replay
→Modify
→…

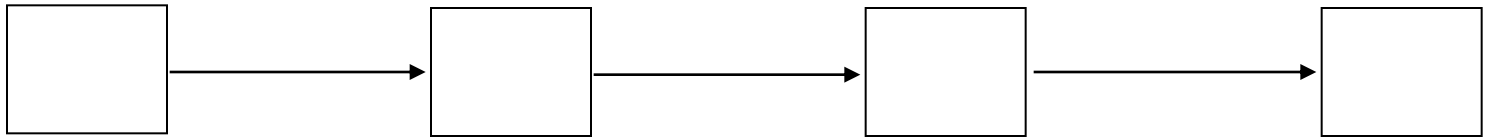
Networks and Cryptography

- ISO/OSI model
- Conceptually, each host has peer at each layer
  - Peers communicate with peers at same layer

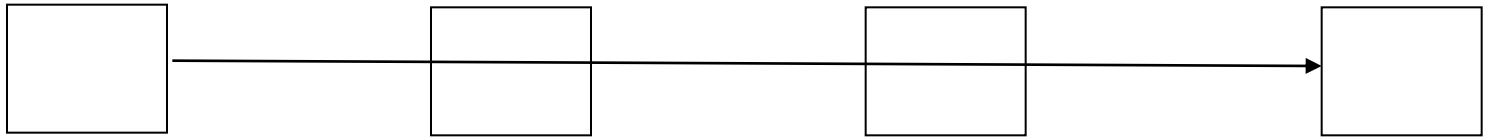


Link and End-to-End Protocols

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

Cryptographic Considerations

- Link encryption
  - Each host shares key with neighbor
- End-to-end
  - Each host shares key with destination
  - Message cannot be read at intermediate nodes

Eve: Traffic Analysis

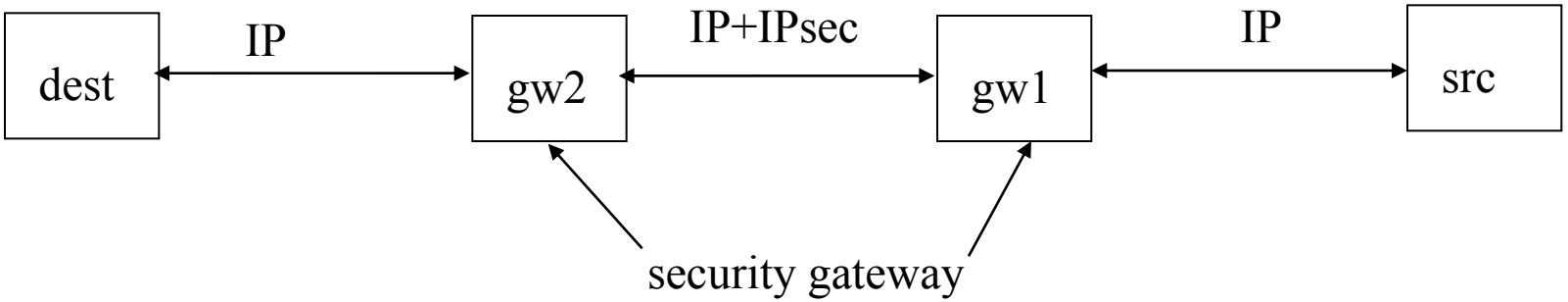
- Link encryption
  - Can protect headers of packets
  - Possible to hide source and destination
    - Note: may be able to deduce this from traffic flows
- End-to-end encryption
  - Cannot hide packet headers
    - Intermediate nodes need to route packet
  - Attacker can read source, destination

Real Life: Secure Communication

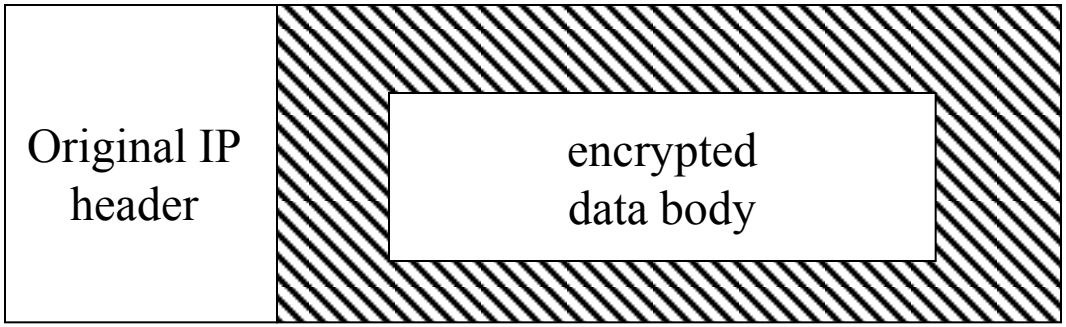
- IP Security (IPSec)
  - Network layer protocol
  - VPNs
- Secure Socket Layer (SSL)
  - Transport layer protocol

IPsec (RFC 4301-4309)

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

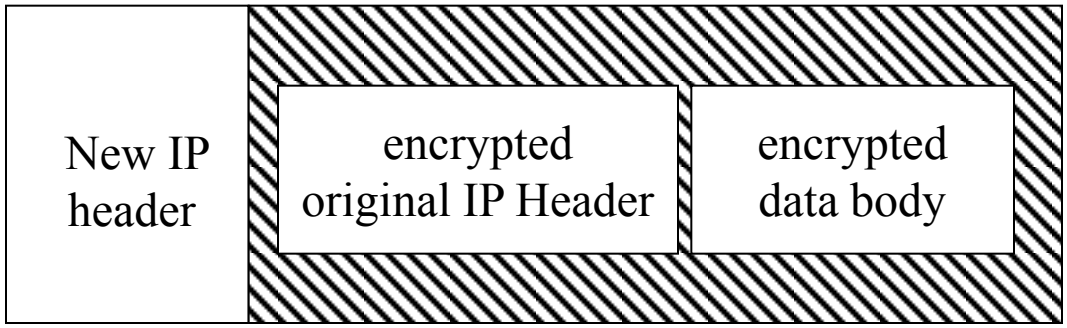


IPsec Transport Mode



- Encapsulate IP packet data area
- Use IP to send IPsec-wrapped data packet
- Note: IP header not protected
- Used: when both endpoints support IPSec

IPsec Tunnel Mode



- Encapsulate IP packet (IP header *and* IP data)
- Use IP to send IPsec-wrapped packet
- Note: IP header protected
- Used: When only two intermediate nodes support IPSec

IPsec Sub-Protocols

- Authentication Header (AH) Protocol
  - Adds authentication to an IP datagram: crypto-hash/MAC (e.g. SHA) of all unchangeable or predictable fields
  - Message integrity
  - Origin authentication
  - Anti-replay (!) – sequence numbers, 32 slot packet window at receiver
- Encapsulating Security Payload (ESP)
  - Confidentiality – encrypts IP payload
  - Supports: tunnel/transport modes
- IPComp
  - Compress BEFORE encryption (why not after ?! ☺)
- IKE/IKEv2
  - Internet Key Exchange

Where do Keys come from ?

- IKE (Internet Key Exchange) – RFC 2409
  - Compliant with **ISAKMP**: Internet Security Association and Key Management Protocol (RFC 2408)
  - Key Exchange: e.g. Oakley (Diffie-Hellman)
- Manual keys

IPsec Architecture: Associations

- Security Association (SA)
  - Association between peers for security services
    - Identified uniquely by dest address, security protocol (AH or ESP), unique 32-bit number (security parameter index, or SPI – allows receiver to lookup secret key for packet)
    - ISAKMP invention
  - Unidirectional
    - Can apply different services in either direction
  - SA uses either ESP or AH; if both required, 2 SAs needed

SA Database (SAD)

- Entry describes SA. Fields:
  - AH algorithm identifier, keys (e.g. “SHA”)
    - When SA uses AH
  - ESP encryption algorithm identifier, keys (e.g., 3DES)
    - 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)

IPsec Architecture: Policies

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

Example

- Goals
  - Discard SMTP (mail) 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/applied in order
  - If no match for packet, it is discarded

Packet Processing

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

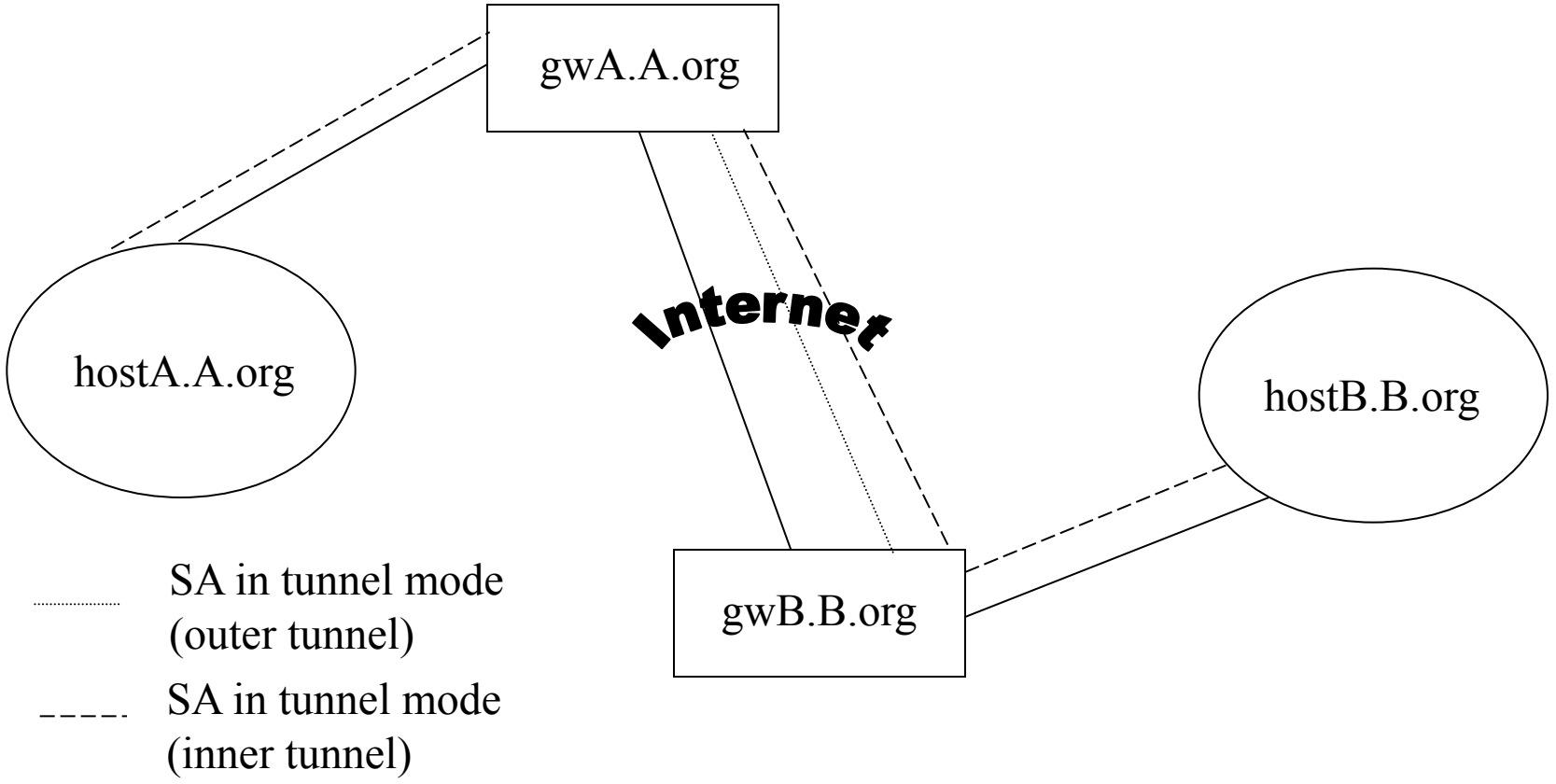
SA Bundles and Nesting

- SA Bundle: sequence of SAs that IPsec applies to packets
- Nest tunnel mode SAs
  - This is *iterated tunneling*

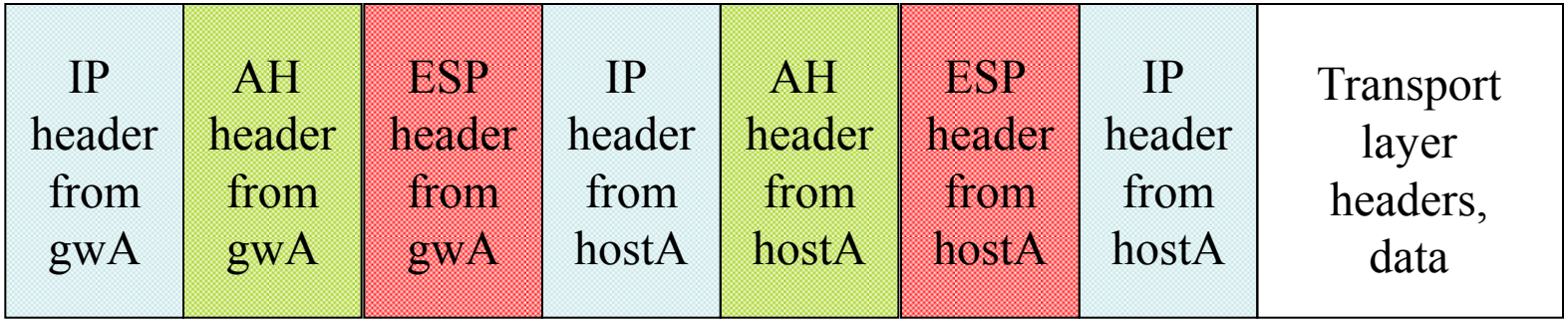
Example: Nested Tunnels

- Group in A.org communicates with group in B.org
- Gateways of A, B use IPsec mechanisms
  - But information must be secret to everyone (even other people in A.org and B.org) except the two groups
- Inner tunnel: a SA between 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*.

AH Protocol

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

Sender

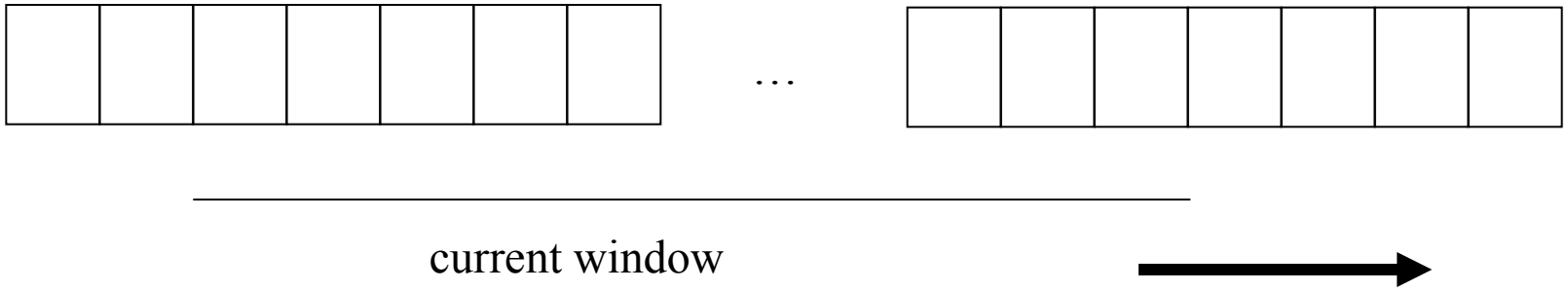
- Check sequence number will not cycle
- Increment sequence number
- Compute IVC of packet
  - Includes IP header, AH header, packet data
    - IP header: include all fields that will not change in transit; assume all others are 0
    - AH header: authentication data field set to 0 for this
    - Packet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



ESP Protocol

- Parameters in ESP header
  - Identification (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
    - Introduced in later versions for efficiency (instead of using AH)

ESP Protocol

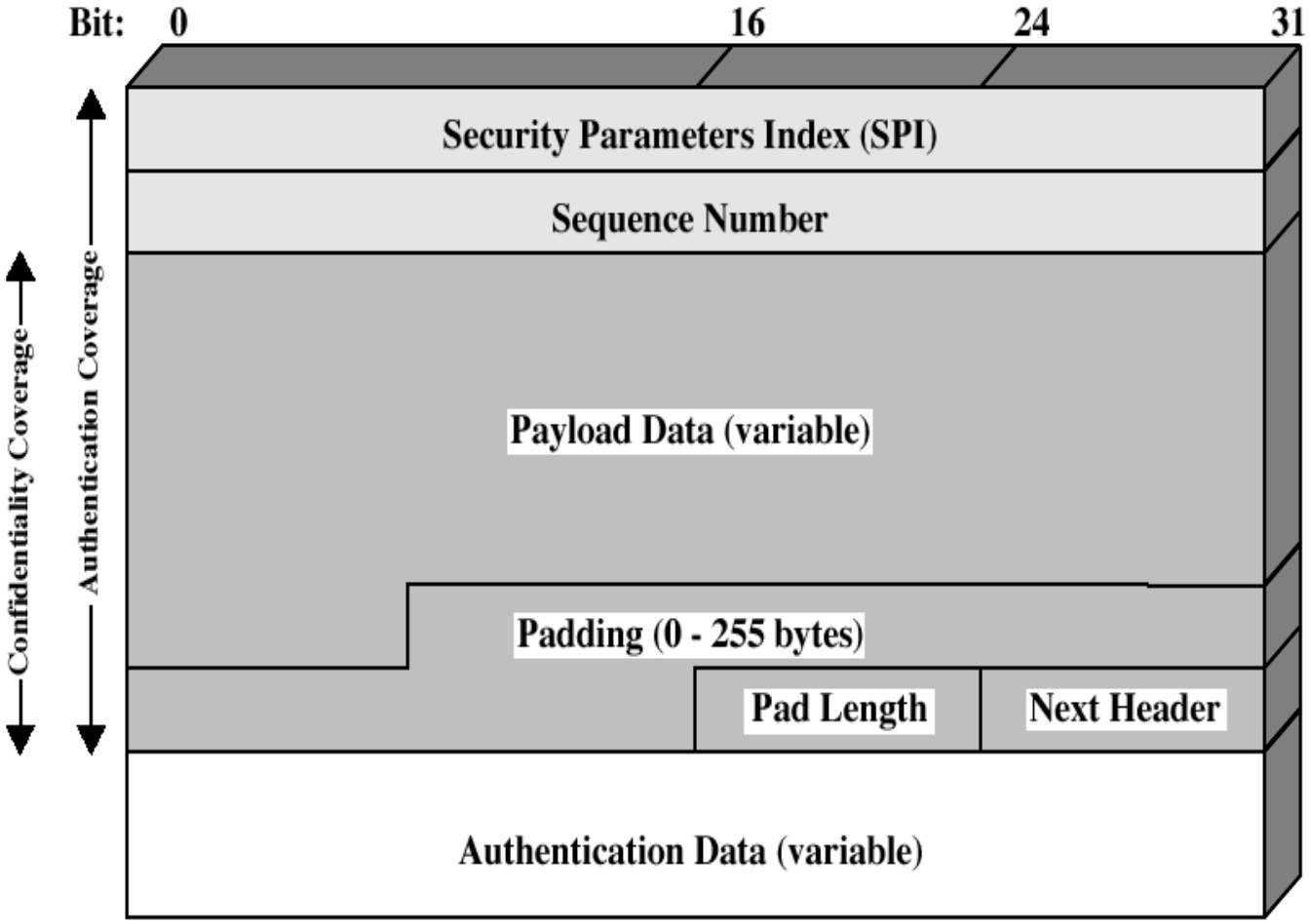


figure © unknown author online

Sender

- Add ESP header
  - Includes whatever padding needed
- Encipher result
  - Do not encipher SPI, sequence numbers
- If authentication desired, compute as for AH protocol: include ESP header, payload; *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 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 header
  - Process padding
  - Decipher payload
  - If SA is transport mode, IP header and payload combined into original IP packet
  - If SA is tunnel mode, payload is an encapsulated IP packet and so is treated as original IP packet

ESP: Transport Mode

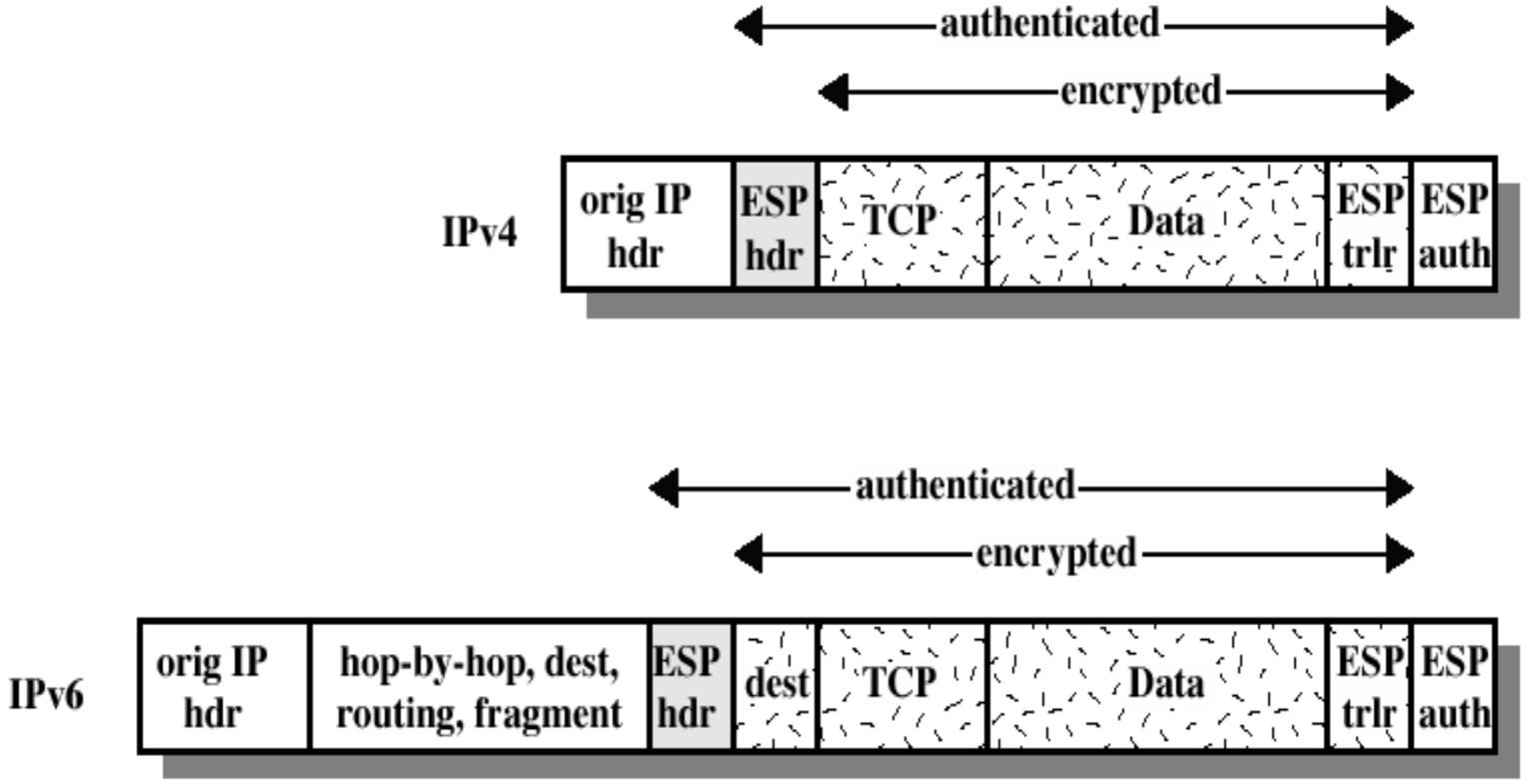
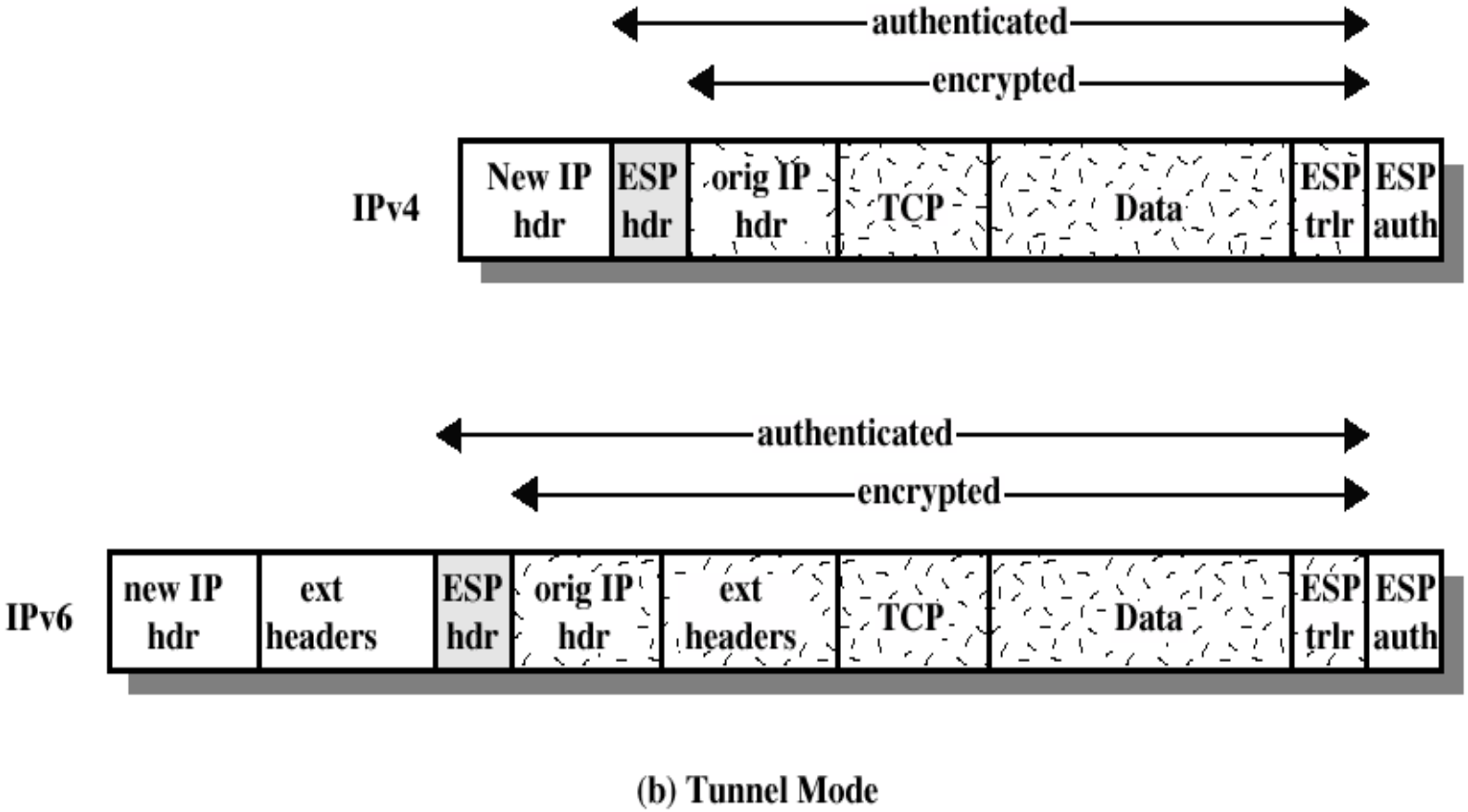


figure © unknown author online

ESP: Tunnel Mode



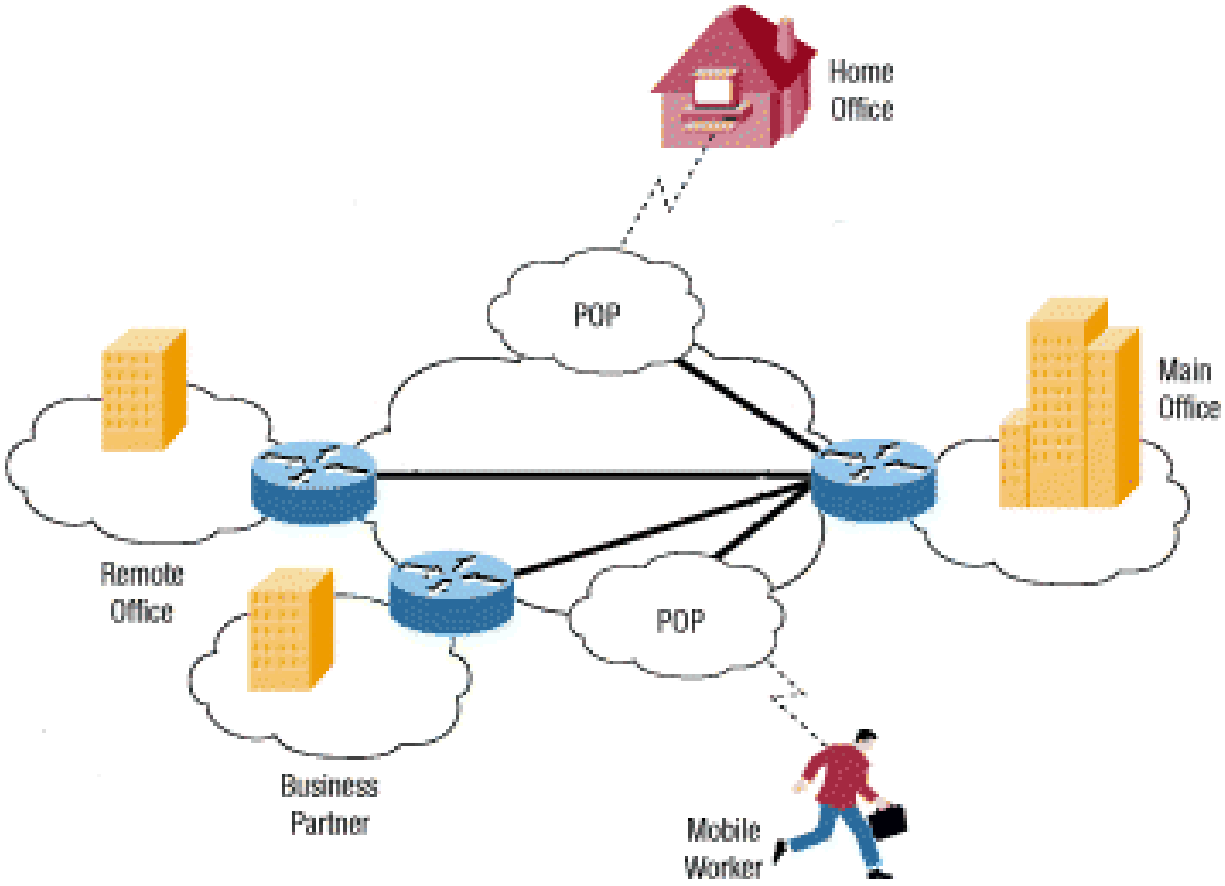
(b) Tunnel Mode

figure © unknown author online

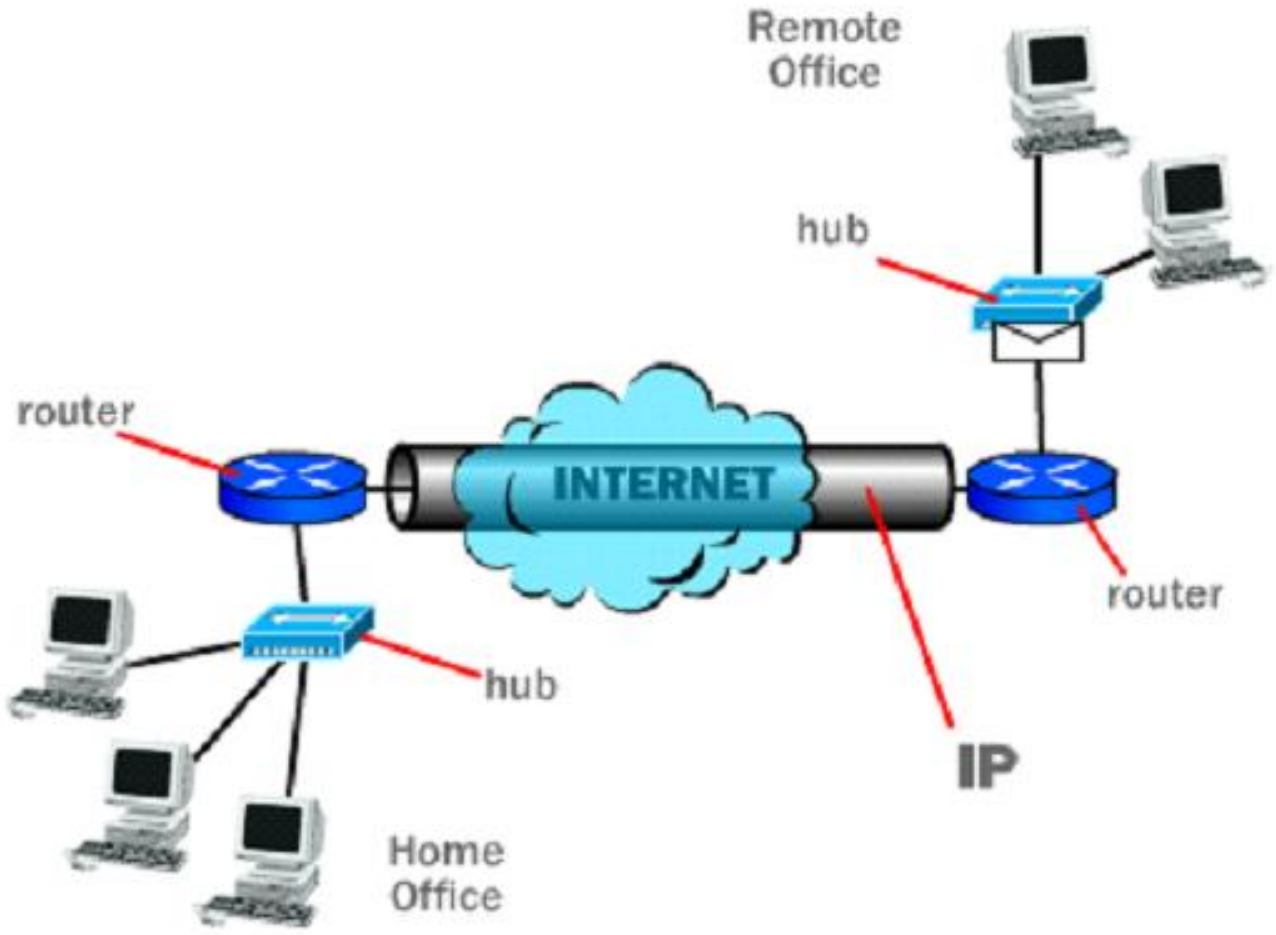
More ESP Miscellany

- 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

Typical VPN


Home
Office
POP
Main
Office
Remote
Office
POP
Business
Partner
Mobile
Worker

Site to Site VPN



Protocol types

- **Passenger -** original data being carried (IPX, NetBeui, IP)

- **Encapsulator -** “wrapper” for original data (GRE-Cisco, IPSec, L2F, PPTP, L2TP)

- **Carrier -** the network that the information is traveling through (IP)

Recap: What did we learn in IPSec

- IP provides little security
- IPSec: framework to overlay security
  - AH, ESP provide different guarantees
  - Key exchange
  - Can run in E2E or tunnel (nested too)
- Below App/transport
  - Offers security to all apps
  - Unlike SSL
- Critical to many "enterprise" deployments