
THE UNIVERSITY OF BRITISH COLUMBIA

# Network Security

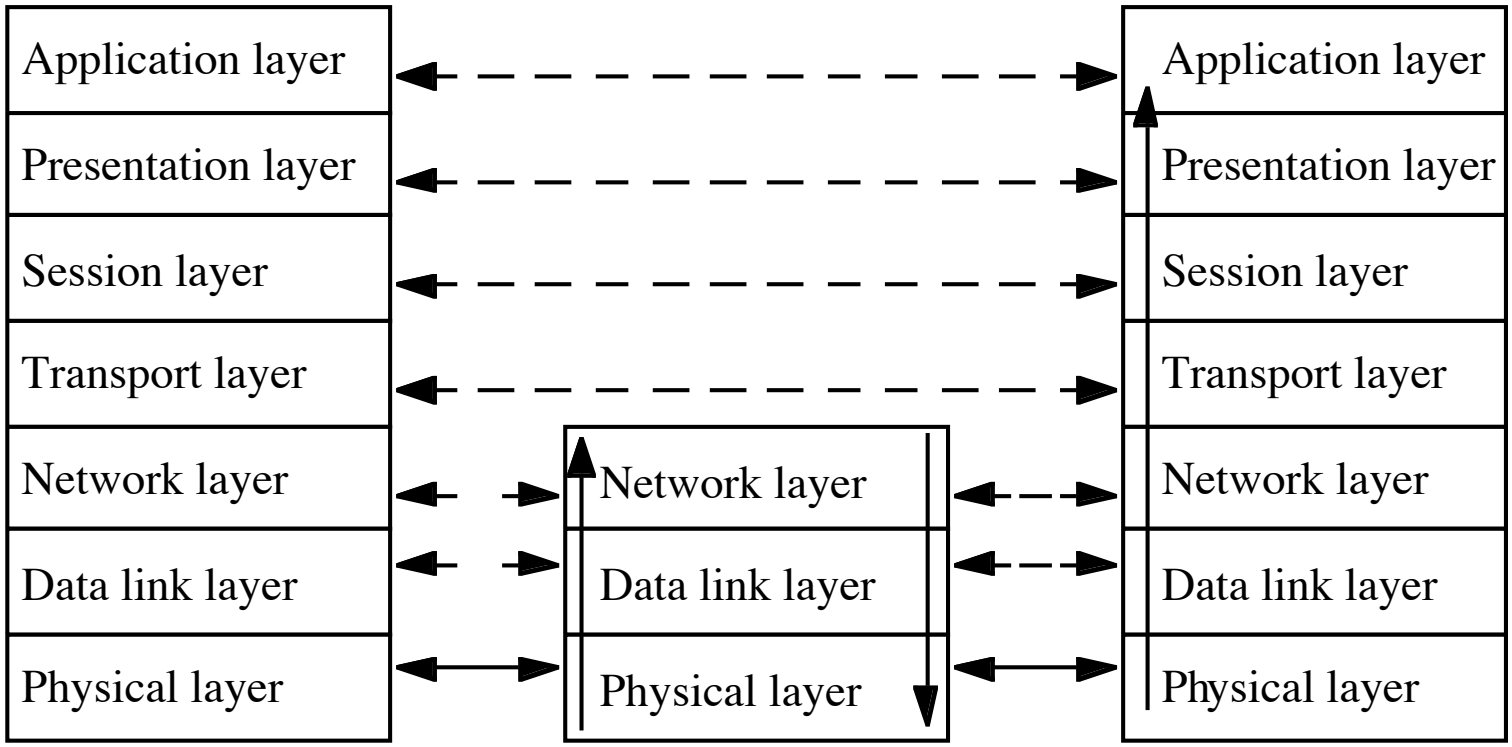
EECE 412

Session 8

# Outline

- Link & end-to-end protocols
- SSL
- IPSec

# Networks


Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data link layer
Physical layer
Network layer
Data link layer
Physical layer
Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data link layer
Physical layer

# Link and End-to-End Protocols

Link Protocol

End-to-End (or E2E) Protocol

# Examples

- Telnet protocol
  - Messages between client, server enciphered, and encipherment, 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

# Link vs. End-to-end protection

## 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
- Attacker can read source, destination

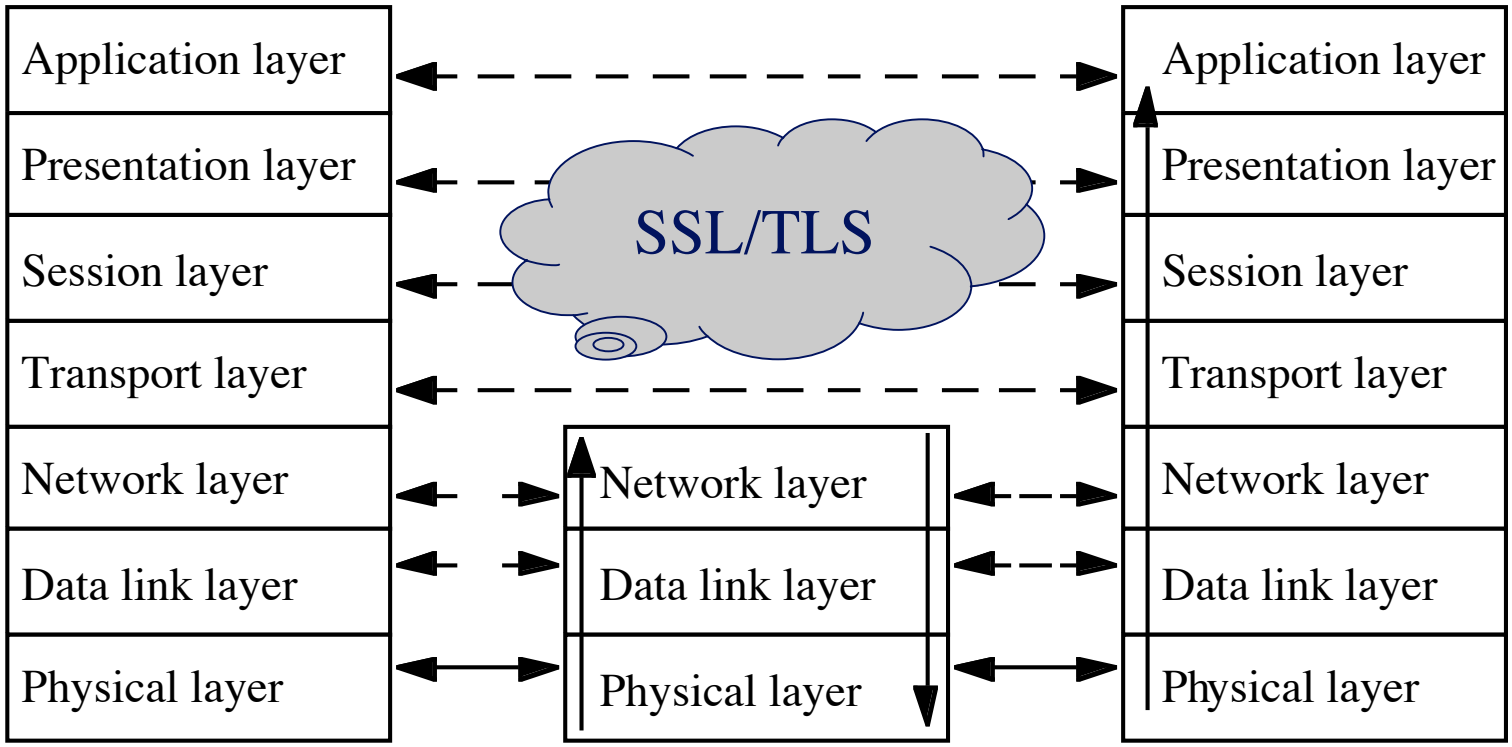
# Example Protocols

- Privacy-Enhanced Electronic Mail (PEM)
  - Applications layer protocol
- Secure Socket Layer (SSL)
  - Transport layer protocol
- IP Security (IPSec)
  - Network layer protocol

THE UNIVERSITY OF BRITISH COLUMBIA

# Secure Socket Layer (SSL)
# a.k.a.
# Transport Layer Security (TLS)

# Networks


Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data link layer
Physical layer
SSL/TLS
Network layer
Data link layer
Physical layer
Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data link layer
Physical layer

# SSL Session

Association between two peers

- Two peers may have several sessions
- Information for each association:
  - Unique session identifier
  - Peer's X.509v3 certificate, if needed
  - Compression method
  - Cipher spec for cipher and MAC
  - "Master secret" shared with peer
    - 48 bits

# SSL Connection

Describes how data exchanged with peer in a session

- Several connections per session
- Information for each connection
  - Random data
  - Write keys (used to encipher data)
  - Write MAC key (used to compute MAC)
  - Initialization vectors for ciphers, if needed
  - Sequence numbers

# Supporting Crypto

- All parts of SSL use them
- Initial phase: public key system exchanges keys
  - Messages enciphered using classical ciphers, and MACed
  - Only certain combinations allowed
    - Depends on algorithm for key exchange cipher
  - Key exchange (a.k.a., interchange) algorithms:
    - RSA
    - Diffie-Hellman
    - Fortezza

# RSA: Cipher, MAC Algorithms

| ***Interchange cipher*** | ***Classical cipher*** | ***MAC Algorithm*** |
|---|---|---|
| RSA, key ≤ 512 bits | *none* | MD5, SHA |
| | RC4, 40-bit key | MD5 |
| | RC2, 40-bit key, CBC mode | MD5 |
| | DES, 40-bit key, CBC mode | SHA |
| RSA | *None* | MD5, SHA |
| | RC4, 128-bit key | MD5, SHA |
| | IDEA, CBC mode | SHA |
| | DES, CBC mode | SHA |
| | DES, EDE mode, CBC mode | SHA |

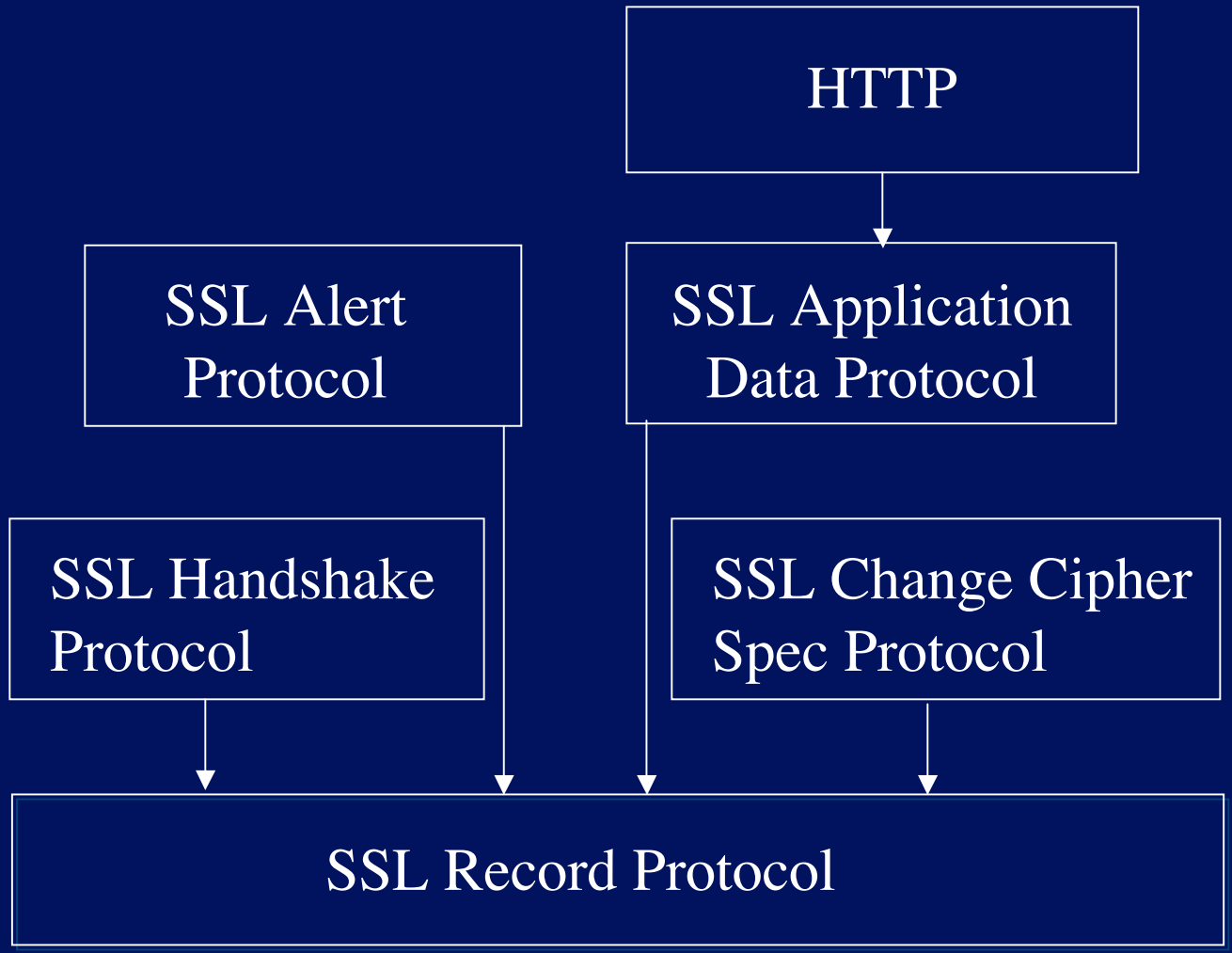
# D-H: Cipher, MAC Algorithms

| *Interchange cipher* | *Classical cipher* | *MAC Algorithm* |
|---|---|---|
| Diffie-Hellman, DSS Certificate | DES, 40-bit key, CBC mode | SHA |
| | DES, CBC mode | SHA |
| | DES, EDE mode, CBC mode | SHA |
| Diffie-Hellman, key ≤ 512 bits RSA Certificate | DES, 40-bit key, CBC mode | SHA |
| | DES, CBC mode | SHA |
| | DES, EDE mode, CBC mode | SHA |

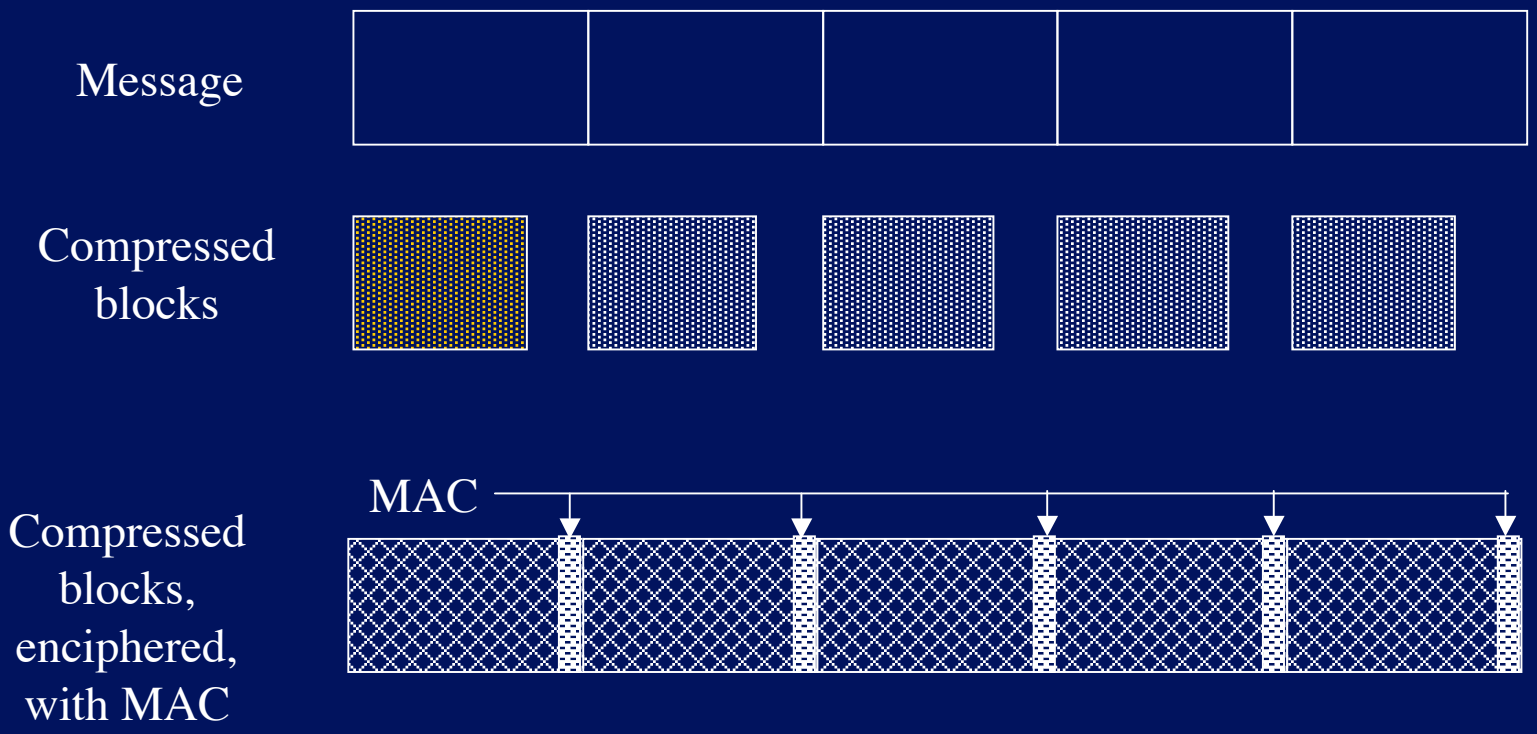
# Fortezza: Cipher, MAC Algorithms

| ***Interchange cipher*** | ***Classical cipher*** | ***MAC Algorithm*** |
|---|---|---|
| Fortezza key exchange | *none* | SHA |
| | RC4, 128-bit key | MD5 |
| | Fortezza, CBC mode | SHA |

# SSL Protocols



# SSL Record Layer

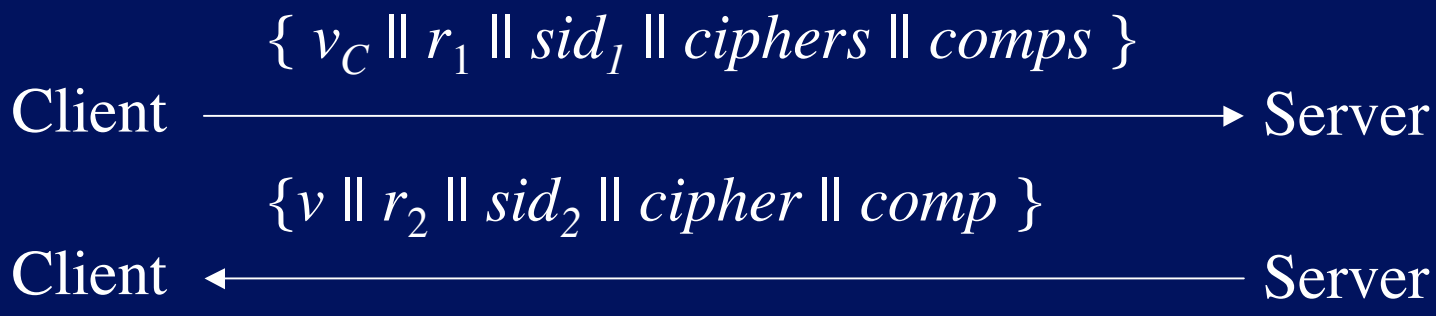


# Overview of Handshake Rounds

1. Create SSL connection between client, server
2. Server authenticates itself
3. Client validates server, begins key exchange
4. Acknowledgments all around

# Handshake Round 1

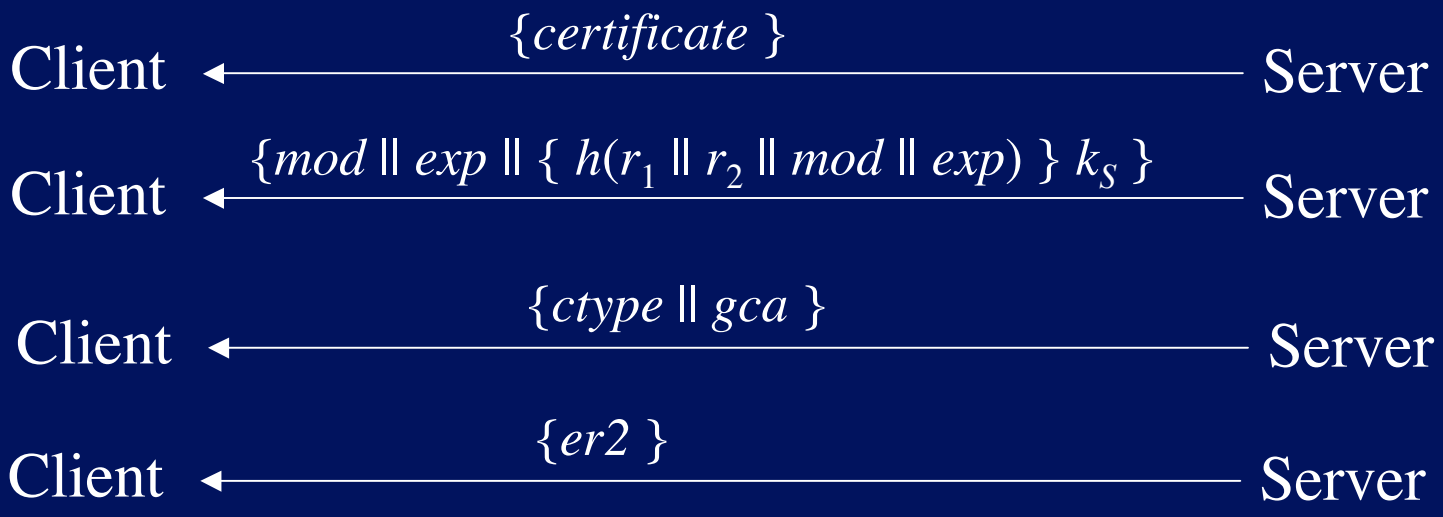
Purpose: Create SSL connection between client, server




| | |
|---|---|
| $v_C$ | Client's version of SSL |
| $v$ | Highest version of SSL that Client, Server both understand |
| $r_1$, $r_2$ | nonces (timestamp and 28 random bytes) |
| $sid_1$ | Current session id (0 if new session) |
| $sid_2$ | Current session id (if s1 = 0, new session id) |
| *ciphers* | Ciphers that client understands |
| *comps* | Compression algorithms that client understand |
| *cipher* | Cipher to be used |
| *comp* | Compression algorithm to be used |

# Handshake Round 2

## Purpose: Server authenticates itself




Note: if Server not to authenticate itself, only last message sent; third step omitted if Server does not need Client certificate

| | |
|---|---|
| *mod* | *public key modulus* |
| *exp* | *public key exponent* |
| $k_S$ | Server's private key |
| *ctype* | Certificate type requested (by cryptosystem) |
| *gca* | "Good" certification authorities |
| *er2* | End round 2 message |

# Handshake Round 3

## Purpose: Client validates server, begins key exchange


Both Client, Server compute master secret *master*:

$master = \text{MD5}(pre \,||\, \text{SHA}(\text{'A'} \,||\, pre \,||\, r_1 \,||\, r_2) \,||$
$\text{MD5}(pre \,||\, \text{SHA}(\text{'BB'} \,||\, pre \,||\, r_1 \,||\, r_2) \,||$
$\text{MD5}(pre \,||\, \text{SHA}(\text{'CCC'} \,||\, pre \,||\, r_1 \,||\, r_2)$


| | |
|---|---|
| *msgs* | Concatenation of previous messages sent/received this handshake |
| *opad*, *ipad* | As above |

# Handshake Round 4

Client sends "change cipher spec" message using that protocol

Client ⟶ Server

{ *h*(*master* || *opad* || *h*(*msgs* || 0x434C4E54 || *master* || *ipad* )) }

Client ⟶ Server

Server sends "change cipher spec" message using that protocol

Client ⟵ Server

{ *h*(*master* || *opad* || *h*(*msgs* || *master* | *ipad*)) }

Client ⟵ Server

*msgs* Concatenation of messages sent/received this handshake in *previous* rounds (does notinclude these messages)

*opad*, *ipad*, *master* As above

# SSL Change Cipher Spec Protocol

- Send single byte
- In handshake, new parameters considered "pending" until this byte received

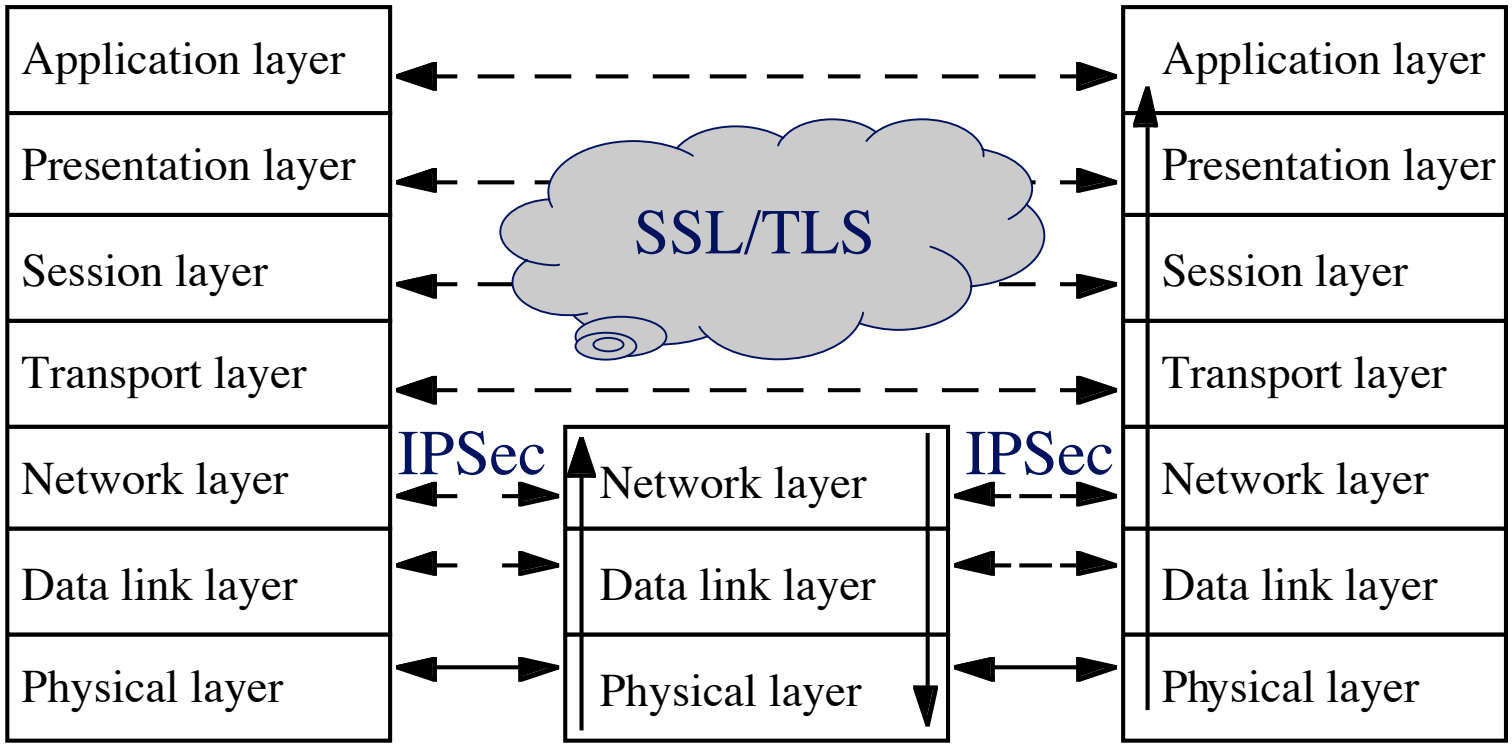
# SSL Alert Protocol

- Closure alert
  - Sender will send no more messages
- Error alerts
  - Warning
    - connection remains open
  - Fatal error
    - connection torn down as soon as sent or received


THE UNIVERSITY OF BRITISH COLUMBIA

# IPSec

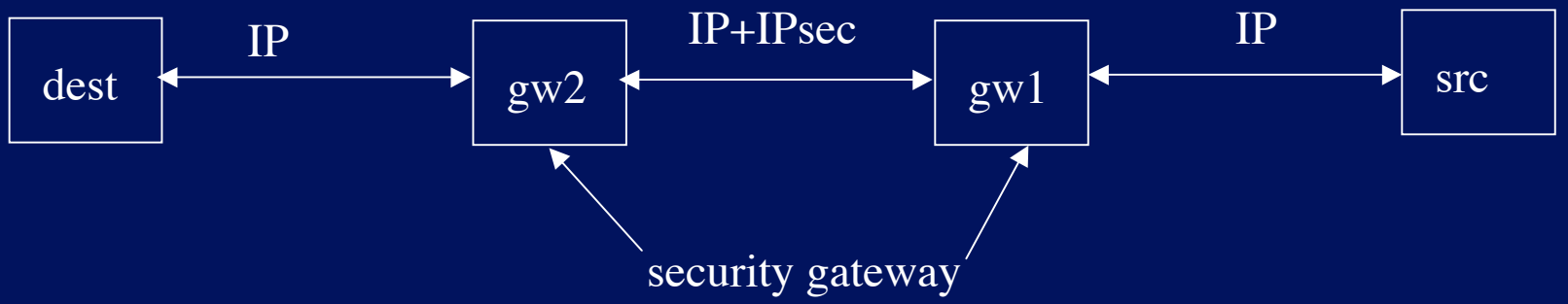
# Networks


Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data link layer
Physical layer
SSL/TLS
IPSec
IPSec
Network layer
Data link layer
Physical layer
Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data link layer
Physical layer

# IPsec

Provides

- message confidentiality
- message integrity
- authentication of endpoints
- replay detection



# Transport Mode


- Encapsulate IP packet data
- Use IP to send IPsec-wrapped data packet
- Note: IP header not protected

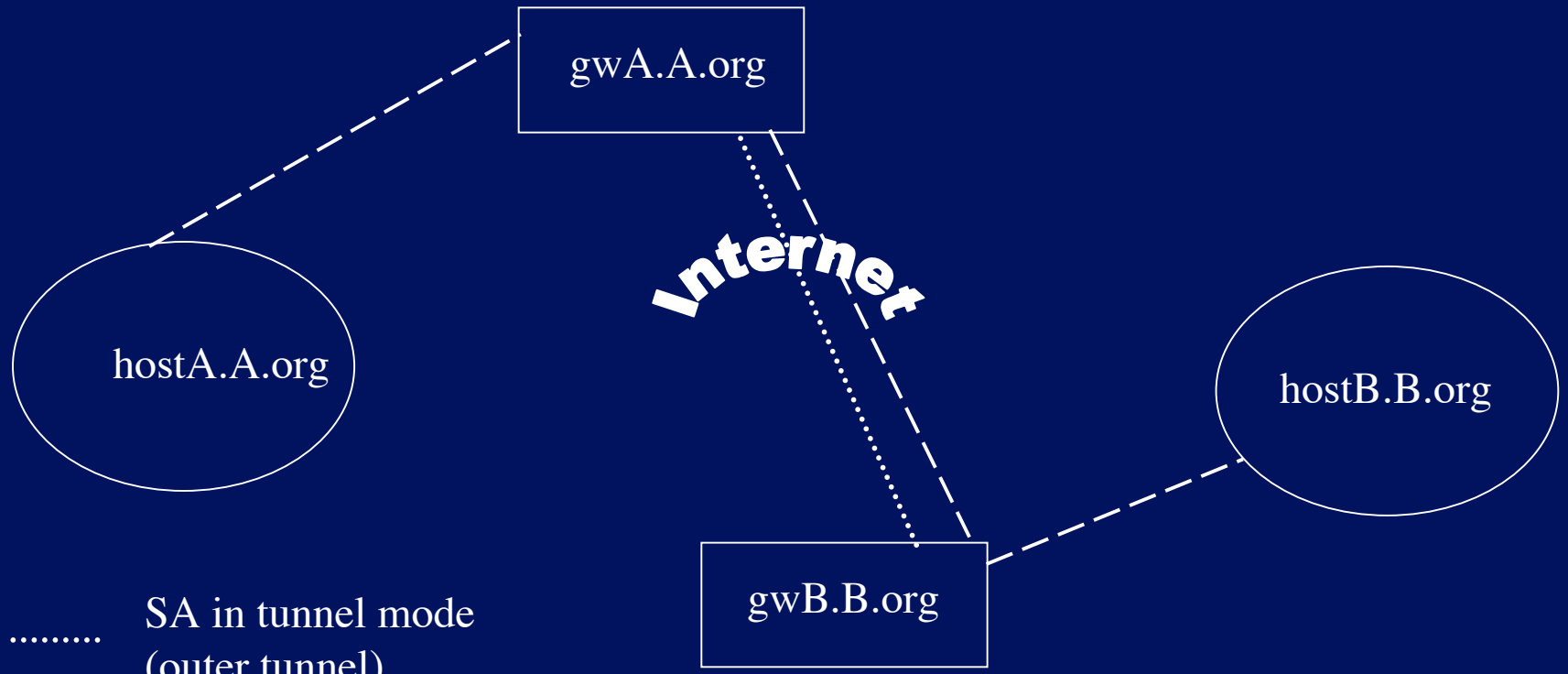
# Tunnel Mode


- Encapsulate whole IP packet (header *and* data)
- Use IP to send IPsec-wrapped packet
- Note: IP header protected

# IPsec Protocols

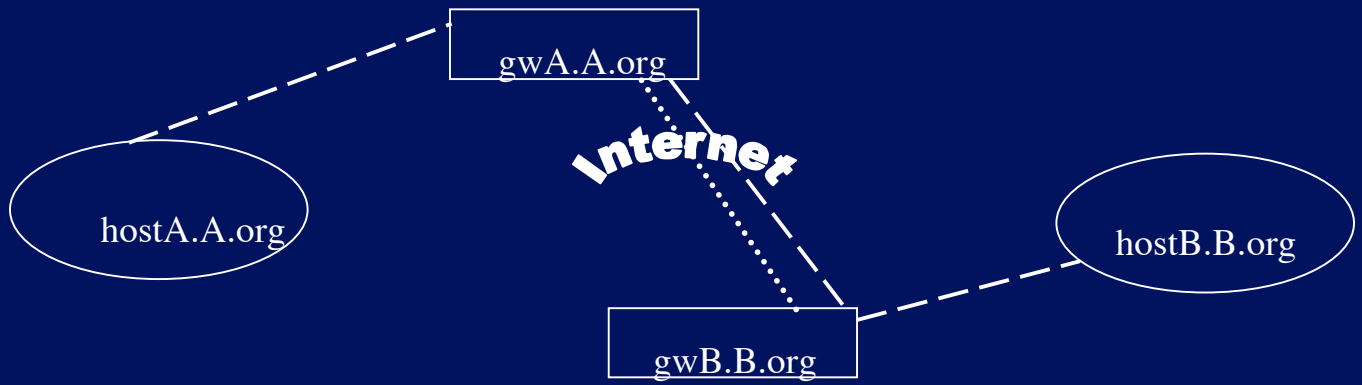
- Authentication Header (AH) protocol
  - Message integrity
  - Origin authentication
  - Anti-replay
- Encapsulating Security Payload (ESP) protocol
  - Confidentiality
  - Others provided by AH

# Example: Systems



# Example: Packets

| 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 |
|---|---|---|---|---|---|---|---|



# IPsec Architecture

Security Policy Database (SPD)

- rules to handle messages
  - discard, add security services, forward unchanged
- associated with network interface
- determines appropriate rule(s) based on IP packet attributes
  - source, destination, transport protocol

# Example

- Goals
  - Discard SMTP packets from host 192.168.2.9
  - Forward SMTP packets from 192.168.19.7 without change
  - "Secure" all other SMTP packets to 10.1.2.3-103
- 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
```

# IPsec Architecture

## Security Association (SA)

- Association between peers for security services
  - Identified by
    1. dest address
    2. security protocol (AH or ESP)
    3. unique 32-bit number (security parameter index, SPI)
- Unidirectional
- SA database (SAD)


THE UNIVERSITY OF BRITISH COLUMBIA

# Which to Use: PEM, SSL, IPsec?