INF3510 Information Security
University of Oslo
Spring 2011

Lecture 9

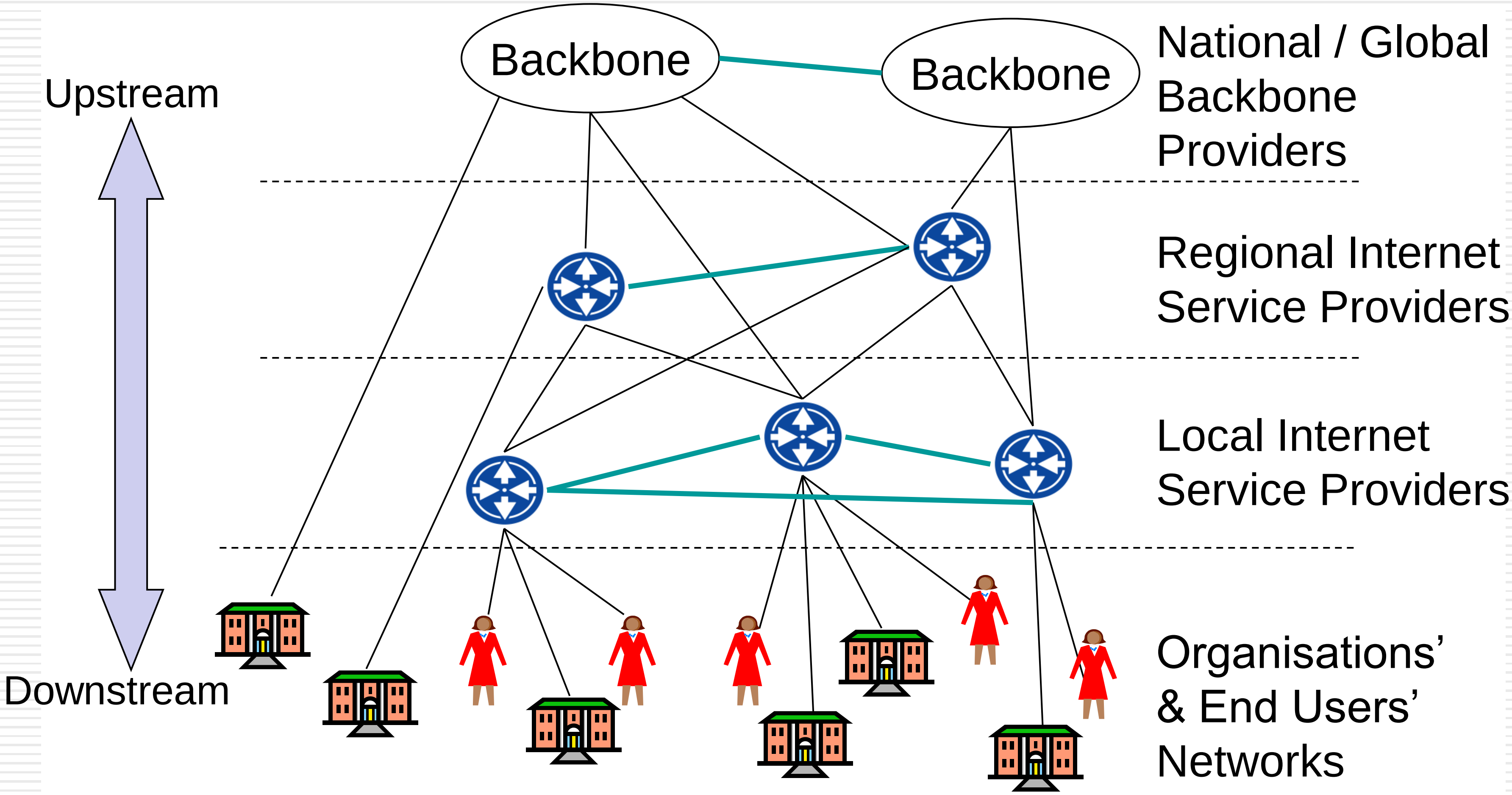
Communication Security

Audun Jøsang

Outline

- Network security concepts
  - Communication security
  - Perimeter security
- Protocol architecture and security services
- Example security protocols
  - HTTP Authentication
  - Transport Layer Security (TLS)
  - IP Layer Security (IPSec)

The Internet Hierarchy

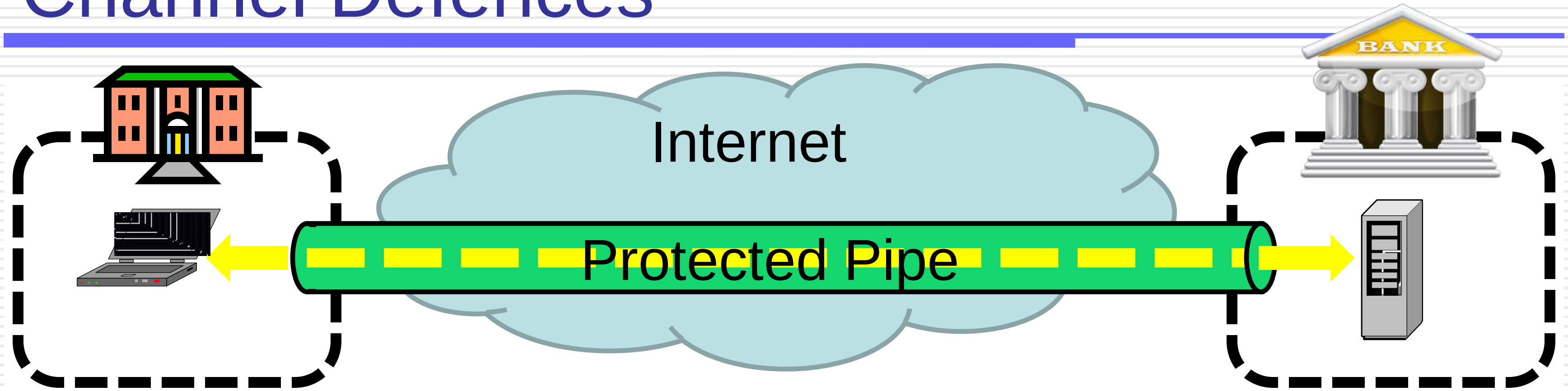

Upstream
Backbone
Backbone
National / Global Backbone Providers
Regional Internet Service Providers
Local Internet Service Providers
Downstream
Organisations' & End Users' Networks

Network Security Concepts

- Assumes that each organisation owns a network
  - Wants to protect own local network
  - Wants to protect communication with other networks

- **Network Security**: two main areas
  - **Communication Security:** measures to protect the data transmitted across networks between organisations and end users
  - **Perimeter Security:** measures to protect an organization's network from unauthorized access (theme for next lecture)

Communication Security Paradigm
Channel Defences



Security Services:

- Authentication
- Non-repudiation
- Integrity
- Confidentiality
- Availability

Provided by protocols:

- Rules for message exchange
- Cryptographically protected messages
- Key exchange/establishment

Communication Security Analogy
Transport Defences

- Transport security might only be meaningful if the endpoints are also secure
- "Using encryption on the Internet is the equivalent of arranging an armored car to deliver credit card information from someone living in a cardboard box to someone living on a park bench." (Gene Spafford)

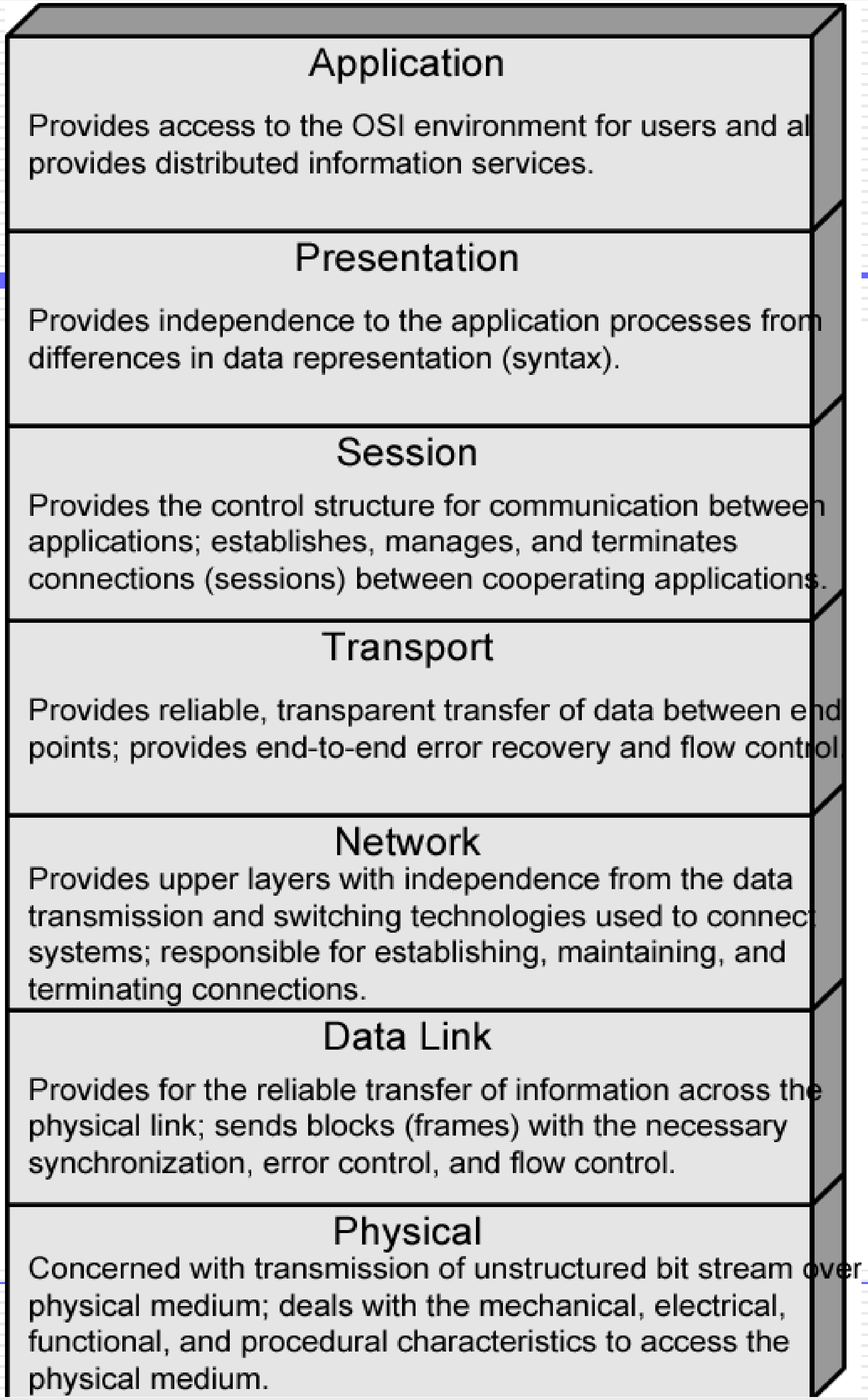
Communication Protocol Architecture

- Layered structure of hardware and software that supports the exchange of data between systems as well as a distributed application (e.g. email or web access)
- Each protocol consists of a set of rules for exchanging messages, i.e. “the protocol”.
- Two standards:
  - OSI Reference model
    - Never lived up to early promises
  - TCP/IP protocol suite
    - Most widely used

OSI – Open Systems Interconnection

- Developed by the International Organization for Standardization (ISO)
- A layer model of 7 layers
- Each layer performs a subset of the required communication functions
- Each layer relies on the next lower layer to perform more primitive functions
- Each layer provides services to the next higher layer
- Changes in one layer should not require changes in other layers

OSI Layers



Application

Provides access to the OSI environment for users and al provides distributed information services.

Presentation

Provides independence to the application processes from differences in data representation (syntax).

Session

Provides the control structure for communication between applications; establishes, manages, and terminates connections (sessions) between cooperating applications.

Transport

Provides reliable, transparent transfer of data between end points; provides end-to-end error recovery and flow control.

Network

Provides upper layers with independence from the data transmission and switching technologies used to connect systems; responsible for establishing, maintaining, and terminating connections.

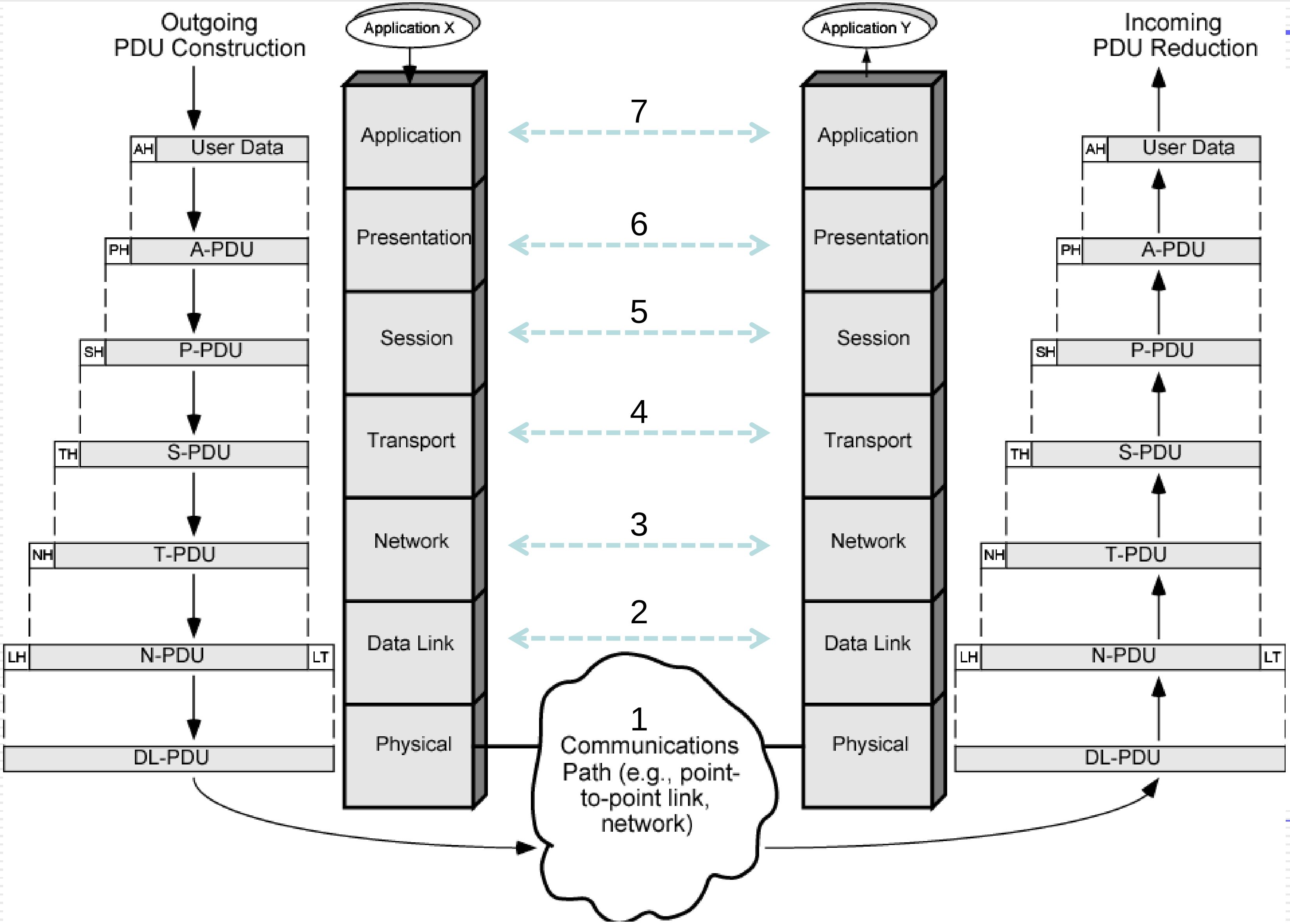
Data Link

Provides for the reliable transfer of information across the physical link; sends blocks (frames) with the necessary synchronization, error control, and flow control.

Physical

Concerned with transmission of unstructured bit stream over physical medium; deals with the mechanical, electrical, functional, and procedural characteristics to access the physical medium.

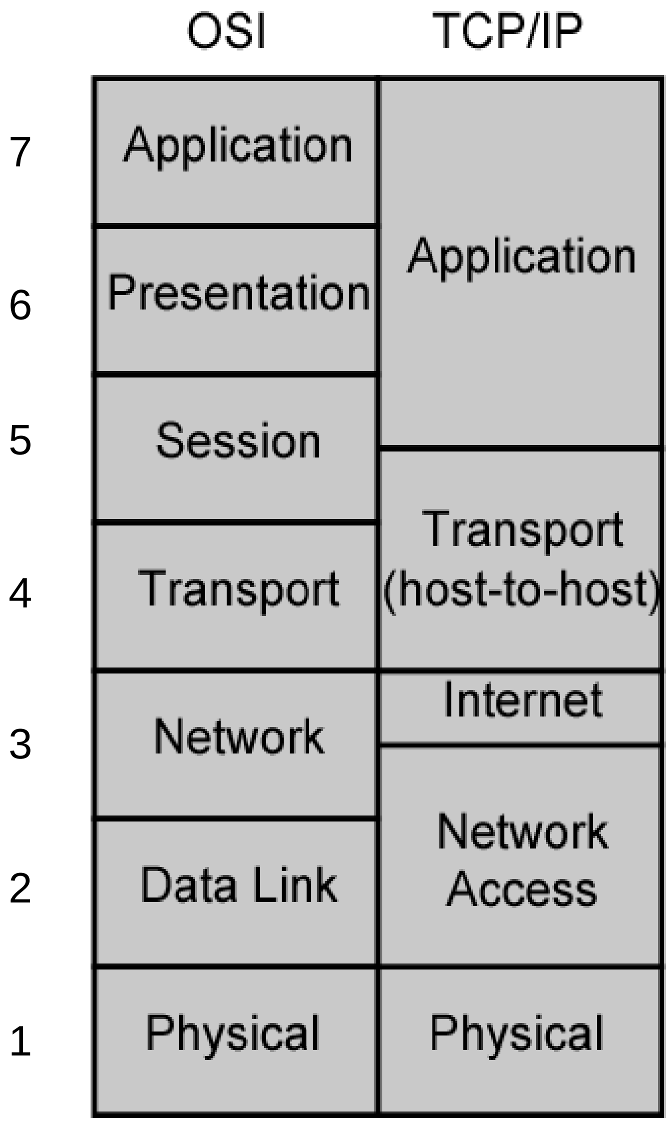
The OSI Environment



TCP/IP Protocol Architecture

- Developed by the US Defense Advanced Research Project Agency (DARPA) for its packet switched network (ARPANET)
- Used by the global Internet
- No official model but a working one.
  - Application layer
  - Host to host or transport layer
  - Internet layer
  - Network access layer
  - Physical layer

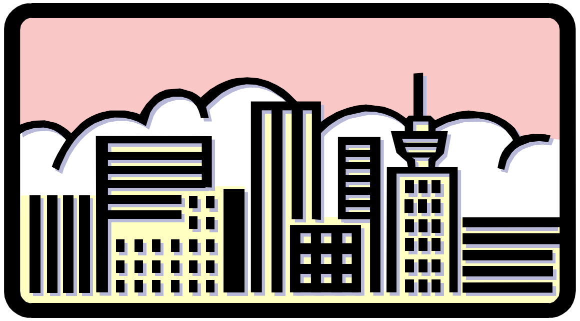
OSI v TCP/IP



	OSI	TCP/IP
7	Application	Application
6	Presentation	
5	Session	
4	Transport	Transport (host-to-host)
3	Network	Internet
2	Data Link	Network Access
1	Physical	Physical

OSI Security Architecture

- Originally specified as ISO 7498-2
- Republished as ITU-T X.800 “Security Architecture for OSI”
- defines a systematic way of defining and providing security requirements
- for us it provides a useful, if abstract, overview of concepts used in the study of security protocols



Security & Protocol Layers (X.800)

Security Service	Layer						
	1	2	3	4	5	6	7*
Peer entity authentication	·	·	Y	Y	·	·	Y
Data origen authentication	·	·	Y	Y	·	·	Y
Access control service	·	·	Y	Y	·	·	Y
Connection confidentiality	Y	Y	Y	Y	·	Y	Y
Connectionless confidentiality	·	Y	Y	Y	·	Y	Y
Selective field confidentiality	·	·	·	·	·	Y	Y
Traffic flow confidentiality	Y	·	Y	·	·	·	Y
Connection Integrity with recovery	·	·	·	Y	·	·	Y
Connection integrity without recovery	·	·	Y	Y	·	·	Y
Selective field connection integrity	·	·	·	·	·	·	Y
Connectionless integrity	·	·	Y	Y	·	·	Y
Selective field connectionless integrity	·	·	·	·	·	·	Y
Non-repudiation Origin	·	·	·	·	·	·	Y
Non-repudiation. Delivery	·	·	·	·	·	·	Y

Security Protocols

- A large variety of security protocols have been specified and implemented for different purposes
  - Authentication
  - Integrity
  - Confidentiality
  - Key establishment/exchange
  - E-Voting
  - Secret sharing
  - etc.
- Protocols are surprisingly difficult to get right!
  - Many vulnerabilities are discovered years later
  - … some are never discovered (maybe only by the attackers)

Protocols Overview

- This lecture discusses the operation of three network-related protocols that are in common use.
  - HTTP Authentication:
    Commonly used to authenticate users before allowing access to web content.
  - Transport Layer Security (TLS):
    Used extensively on the web and is often referred to in privacy policies as a means of providing confidential web connections.
  - IP Security (IPSec):
    Provides security services at the IP level and is used to provide Virtual Private Network (VPN) services.

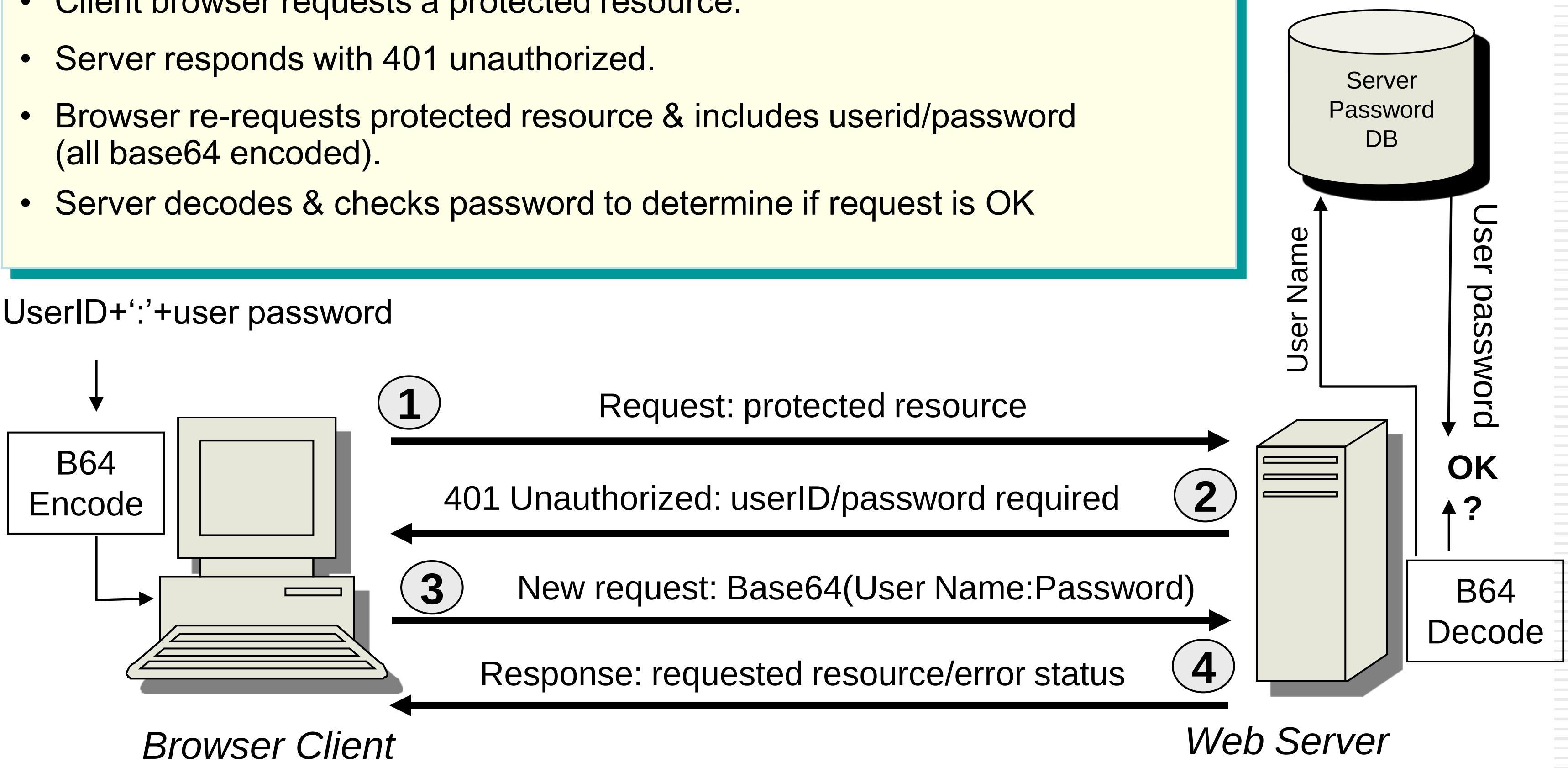
HTTP Authentication

HTTP Authentication: Overview

- Defined in RFC 2617.
- Describes both Basic and Digest authentication.
- HTTP is a request/response protocol.
  - 4xx class of status codes in responses are intended to inform the client of request errors.
  - 401 Unauthorized: The request requires user authentication. The response includes a WWW-Authenticate header field containing a challenge applicable to the requested resource. The client MAY repeat the request with a suitable Authorization header field.
- A realm name is used to indicate to users which username and password to use.

HTTP Authentication: Basic authentication

- Client browser requests a protected resource.
- Server responds with 401 unauthorized.
- Browser re-requests protected resource & includes userid/password (all base64 encoded).
- Server decodes & checks password to determine if request is OK


Server Password DB
User Name
User password
UserID+':'+user password
B64 Encode
1
Request: protected resource
2
401 Unauthorized: userID/password required
3
New request: Base64(User Name:Password)
4
Response: requested resource/error status
OK
?
B64 Decode
Browser Client
Web Server

HTTP Authentication: Basic authentication

- Based on the user supplying a user-ID and a password for each realm for which authentication is required.
- The server will service the request only if it can validate the user-ID and password for the protection space of the Request-URI.
- Upon receipt of an unauthorized request for a URI within the protection space, the origin server MAY respond with a challenge like the following:

  WWW-Authenticate: Basic realm=“SecretSpace”

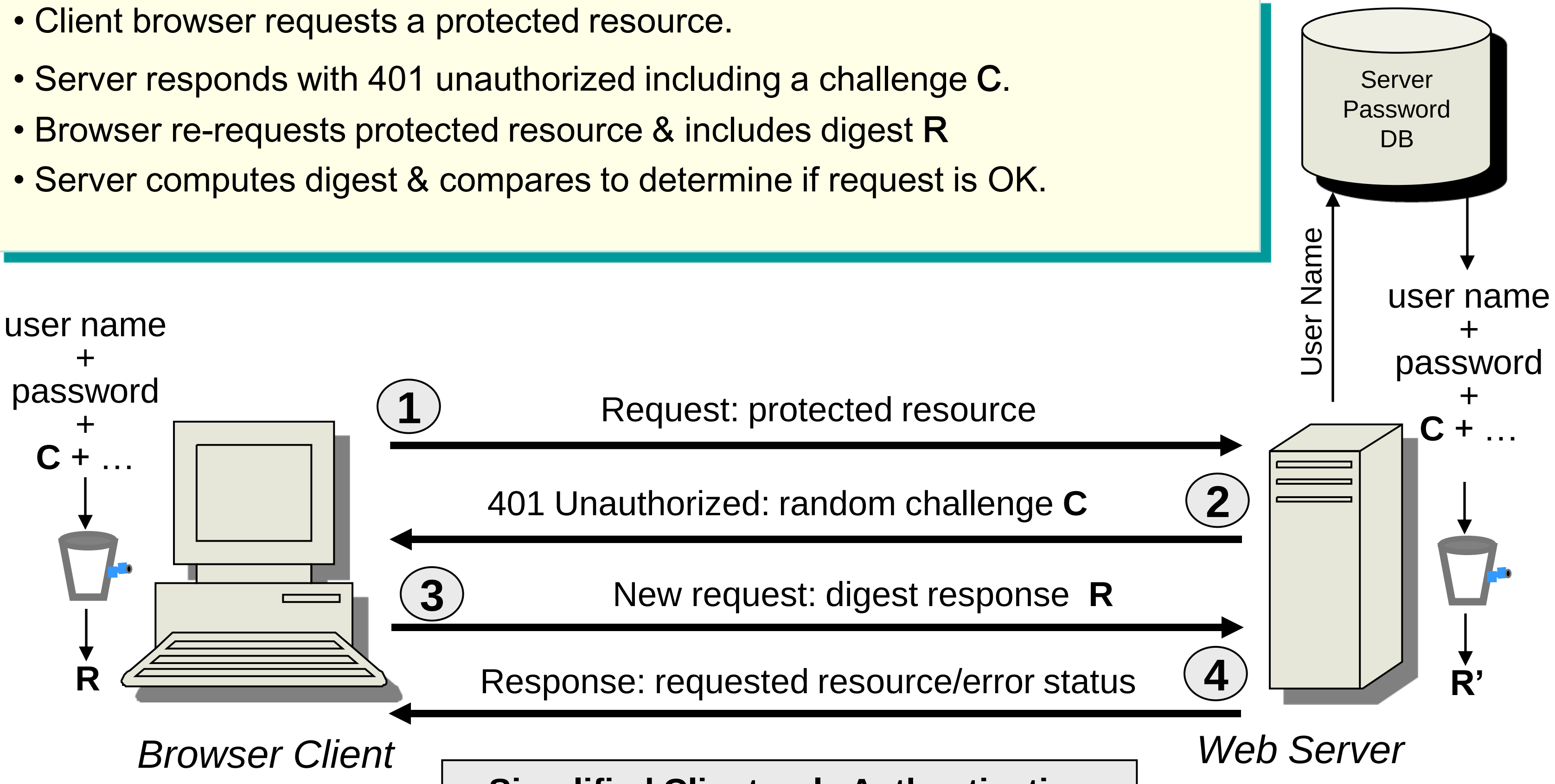
  where “SecretSpace" is the string assigned by the server to identify the protection space of the Request-URI.

HTTP Authentication: Basic authentication security

- Serious Problem: Results in essentially cleartext transmission of the user's password over the physical network.
  - Sniffers can capture the userid and password
  - ‘fake’ web servers can spoof a ‘real’ server and entice users to send their userids and passwords.
- Therefore, it should not be used (without enhancements) to protect sensitive or valuable information.

HTTP Authentication: Digest authentication

- Client browser requests a protected resource.
- Server responds with 401 unauthorized including a challenge **C**.
- Browser re-requests protected resource & includes digest **R**
- Server computes digest & compares to determine if request is OK.


Server Password DB
User Name
user name + password + C + …
user name + password + C + …
R
R'
1
Request: protected resource
2
401 Unauthorized: random challenge C
3
New request: digest response R
4
Response: requested resource/error status
Browser Client
Web Server

Simplified Client-only Authentication

HTTP Authentication: Digest authentication

- Digest scheme is based on a simple challenge-response paradigm.
  - The challenge is a nonce value (randomly chosen and used only once).
  - A valid response contains a hash of values such as the username, the password, the given nonce value, the HTTP method, and the requested URI.
  - The response is sent as the Request-Digest field.
  - The password is never sent in the clear.

HTTP Authentication: Security Considerations

- RFC 2617 lists several security considerations including:
  - Basic Authentication
  - Digest Authentication
  - Comparison of Digest with Basic Authentication
  - Replay Attacks
  - Weakness Created by Multiple Authentication Schemes
  - Online dictionary attacks
  - Man in the Middle
  - Spoofing by Counterfeit Servers
  - Storing passwords

HTTP Authentication: Online dictionary attacks

- If the attacker can eavesdrop, the captured nonce/response pairs can be tested against a list of common words (dictionary).
  - The dictionary is usually much smaller than the total number of possible passwords.
  - The response for each password in the dictionary is determined and compared to the captured response value.
- The server can mitigate this attack by not allowing users to select weak passwords.

HTTP Authentication: Man-in-the-middle

- Both Basic and Digest authentication are vulnerable to man-in-the-middle (MITM) attacks, for example, from a hostile or compromised proxy.
  - Has the same problems as eavesdropping.
  - But also offers some additional opportunities.
- For Digest, the MITM could change the challenge to one that requests only Basic authentication.
  - The offer of a ‘free’ proxy services might tempt a gullible user.
  - Some browsers will warn users if the password is sent in the clear.

HTTP Authentication: Storing passwords

- Both schemes rely on the user/client providing a password and the server ‘checking’ this password.
  - Basic: Server decodes to recover the password and compares to the entry in its password database.
  - Digest: Server computes a hash value and compares this to the received value. The ‘password’ file contains the hash of the username, realm and password.
  - The username and password must be prearranged in some fashion not addressed by RFC2617.
- If the password file is compromised, an attacker can gain access. The password file stored on the server must be secured.

HTTP Authentication: Security Considerations Summary

- Basic authentication alone is not a secure method of user authentication.
- Digest authentication is weak compared to public-key based mechanisms but its better than the much weaker Basic authentication.
- Digest Authentication has very limited integrity services.
  - “Many needs for secure HTTP transactions cannot be met by Digest Authentication. For those needs TLS or SHTTP are more appropriate protocols. In particular Digest authentication cannot be used for any transaction requiring confidentiality protection.” RFC 2617

Transport Layer Security

TLS/SSL

SSL/TLS: Overview

- 1994: Netscape Communications developed the network authentication protocol Secure Sockets Layer (SSL) 2.0.
- 1995: Microsoft countered with a protocol known as Private Communications Technology (PCT) 1.0 which claimed to improve on SSL 2.0.
- 1995: Netscape release their own improvements - SSL 3.0 - which widely used.
- 1996: SSL 3.0 was submitted to the IETF as an Internet draft, and an IETF working group was formed to develop a recommendation.

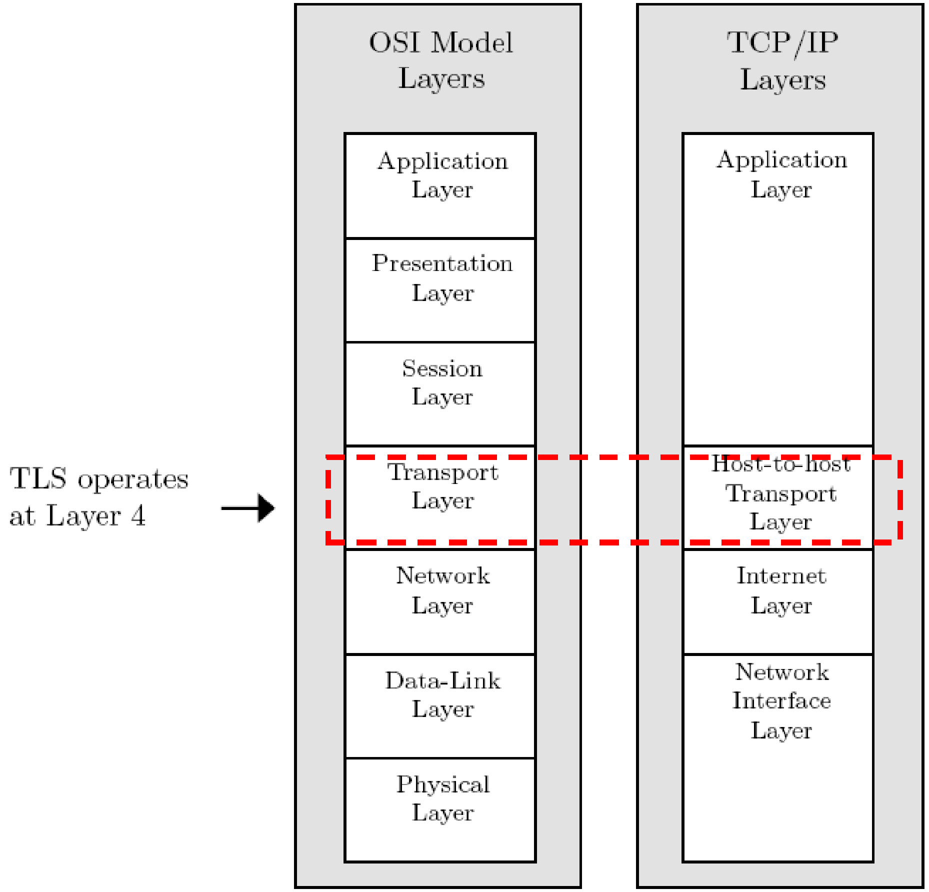
SSL/TLS: Overview

- In January 1999, RFC 2246 was issued by the IETF, documenting the result: the Transport Layer Security (TLS) protocol 1.0, which is virtually indistinguishable from SSL 3.0 but includes some PCT 1.0.
- Microsoft use the term SCHANNEL (secure channel) to name the Security Support Provider (SSP) in Windows that implements all four of the authentication protocols discussed previously—SSL 2.0, PCT 1.0, SSL 3.0, and TLS 1.0.

TLS: Overview

- TLS is a cryptographic services protocol based upon PKI and commonly used on the Internet.
  - Most often used to allow browsers to establish secure sessions with web servers.
- Port 443 is reserved for HTTP over TLS and HTTPS is the name of the URL scheme used with this port.
  - http://www.develop.com implies the use of standard HTTP using port 80.
  - https://www.develop.com implies the use of HTTP over TLS using port 443.

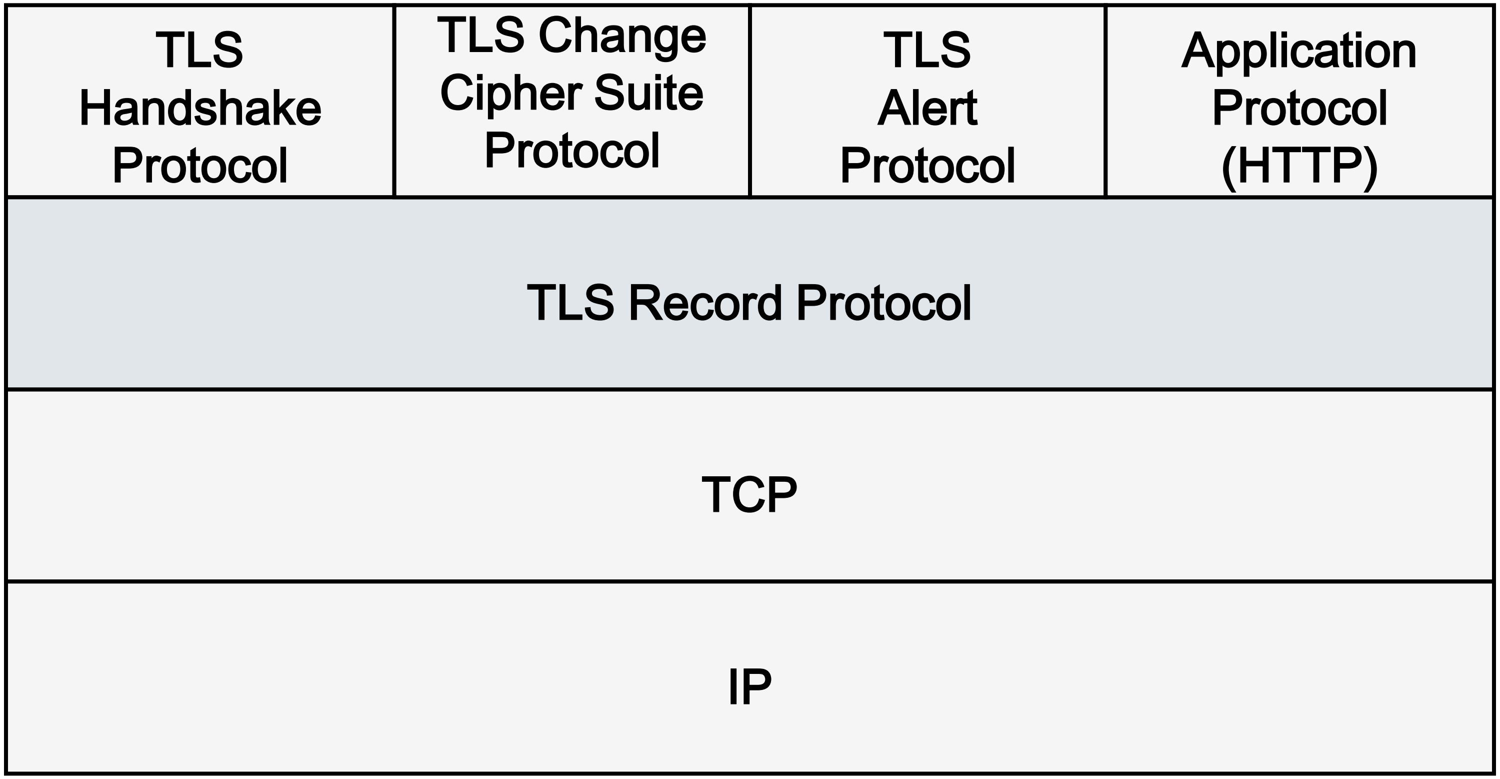
TLS:
Layer 4 Security



TLS: Architecture Overview

- Designed to provide secure reliable end-to-end services over TCP.
- Consists of 3 higher level protocols:
  - TLS Handshake Protocol
  - TLS Alert Protocol
  - TLS Change Cipher Spec Protocol
- The TLS Record Protocol provides basic services to various higher level protocols.

TLS:
Protocol Stack



TLS Handshake Protocol

TLS Change Cipher Suite Protocol

TLS Alert Protocol

Application Protocol (HTTP)

TLS Record Protocol

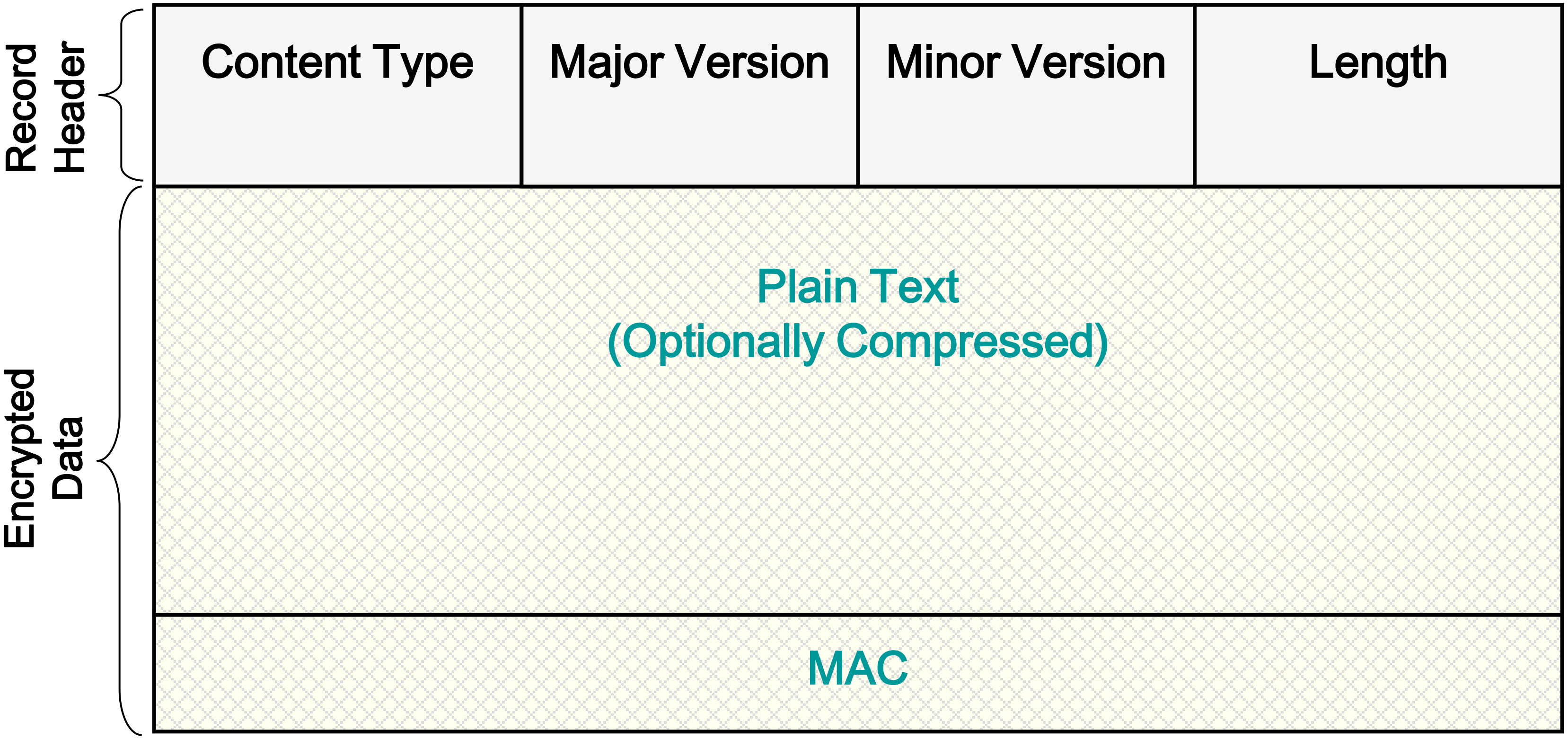
TCP

IP

TLS:
Record Protocol Overview

- Provides two services for SSL connections.
  - Message Confidentiality:
    - Ensure that the message contents cannot be read in transit.
    - The Handshake Protocol establishes a symmetric key used to encrypt SSL payloads.
  - Message Integrity:
    - Ensure that the receiver can detect if a message is modified in transmission.
    - The Handshake Protocol establishes a shared secret key used to construct a MAC.

TLS:
Record Format



Record Header

Content Type

Major Version

Minor Version

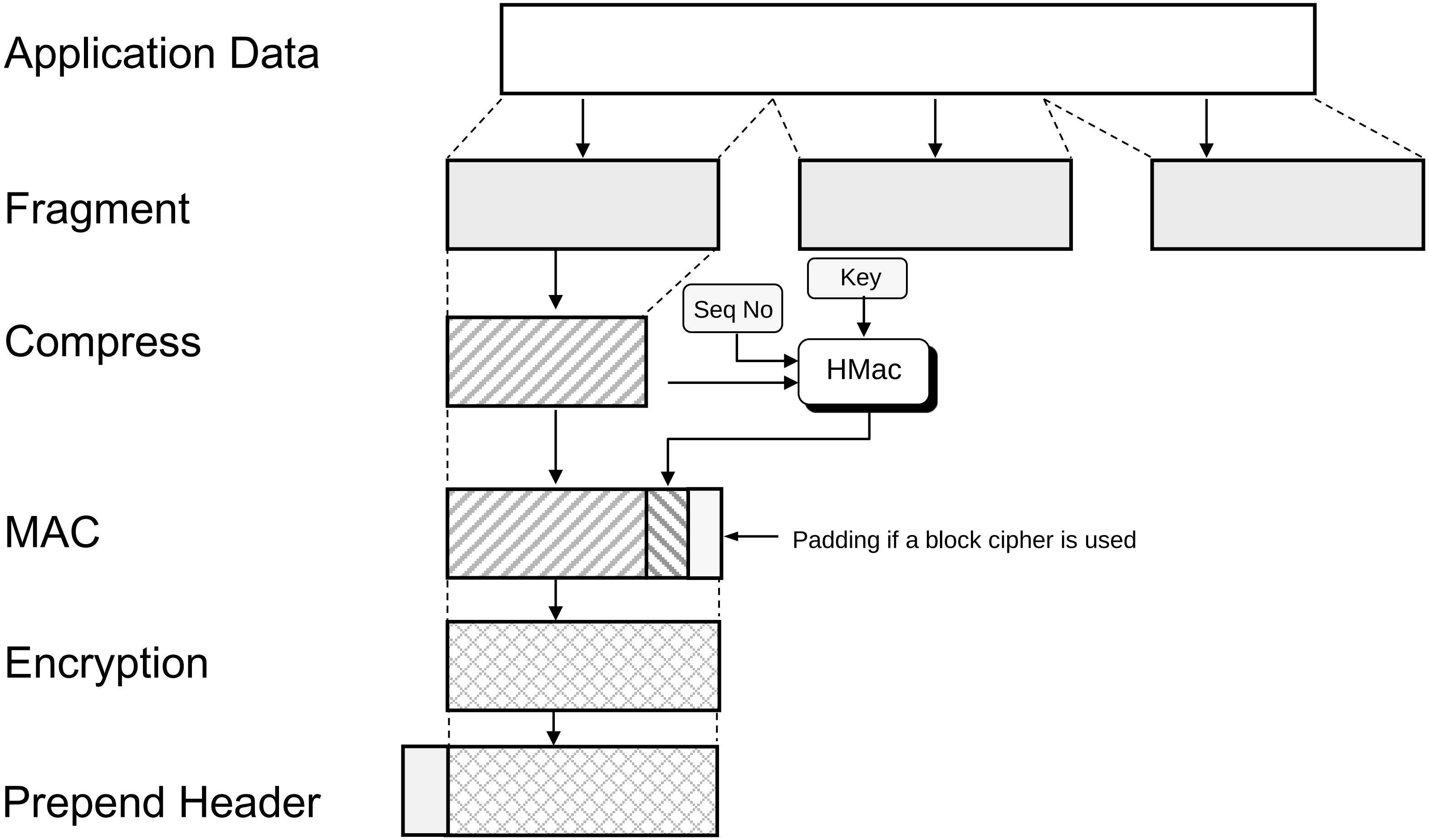
Length

Encrypted Data

Plain Text
(Optionally Compressed)

MAC

TLS: Record Protocol Operation



TLS:
Record Protocol Operation

- Fragmentation:
  - Each application layer message is fragmented into blocks of 214 bytes or less.
- Compression:
  - Optionally applied.
  - SSL v3 & TLS – default compression algorithm is null
- Add MAC:
  - Calculate a MAC over the compressed data using a MAC secret from the connection state.
  - The algorithm used is based on the HMAC as defined in RFC 2104.

TLS: Record Protocol Operation

- Encrypt:
  - The compressed data plus MAC are encrypted using a symmetric cipher.
  - Permitted ciphers include AES, IDEA,DES, 3DES, RC4
  - For block ciphers, padding is applied after the MAC to make a multiple of the cipher’s block size.

TLS:
Record Protocol Operation

- Prepend TLS Record Header containing:
    - Content Type
    - Protocol Version:

Major Version	Minor Version	Version Type
3	0	SSLv3
3	1	TLS 1.0
3	2	TLS 1.1
3	3	TLS 1.2

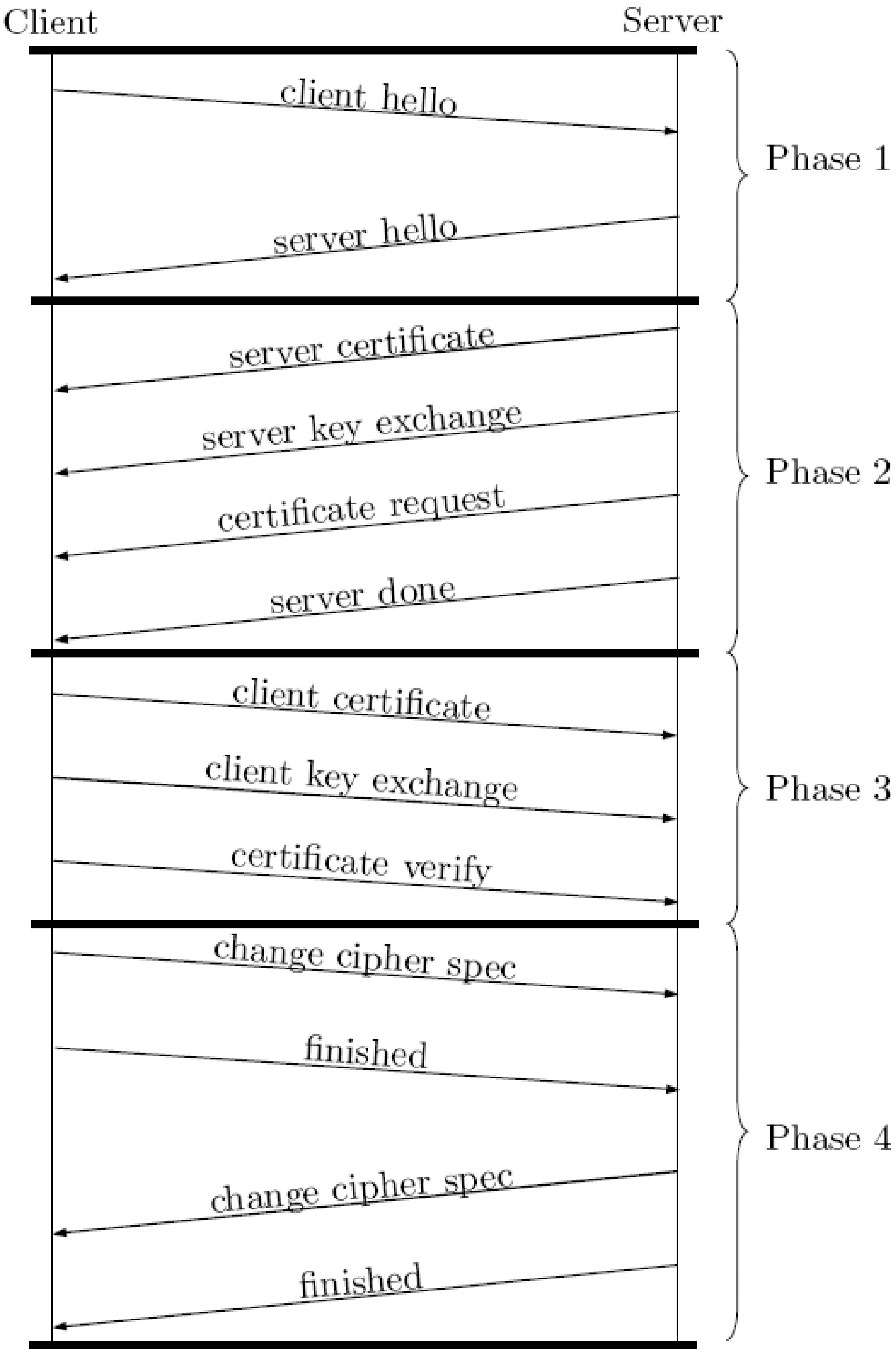
    - Length: length in octets of the data
  - Defined content types are:
    - change_cipher_spec
    - alert
    - handshake
    - application_data

TLS: Handshake Protocol

- The handshake protocol
  - Negotiates the encryption to be used
  - Establishes a shared session key
  - Authenticates the server
  - Authenticates the client (optional)
  - Completes the session establishment
- After the handshake application data is transmitted securely
- Several variations of the handshake exist
  - RSA variants
  - Diffie-Hellman variants

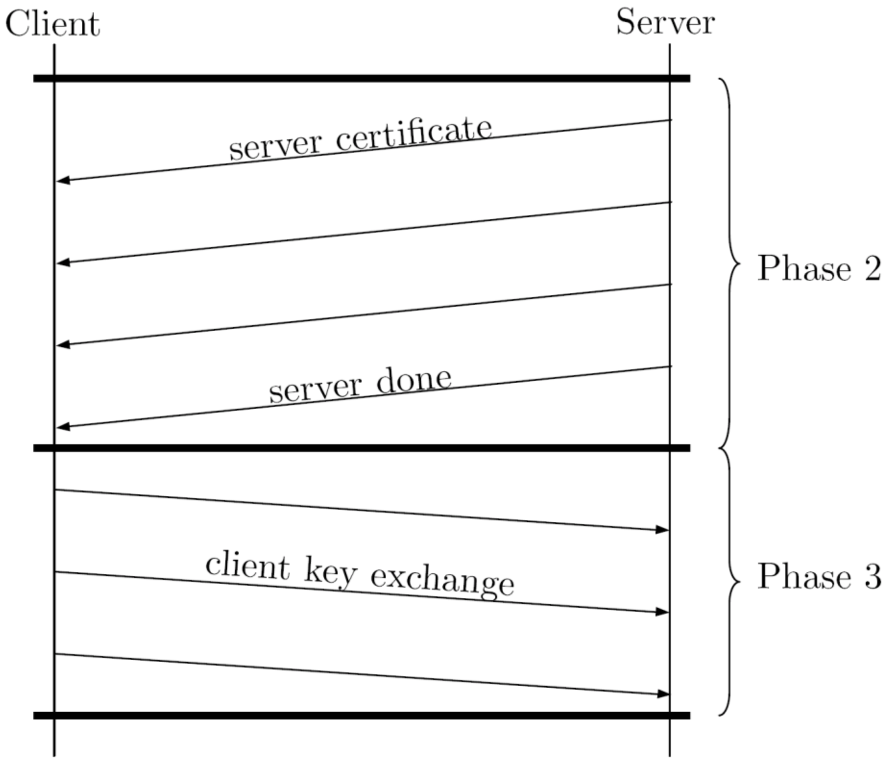
TLS: Handshake Four phases

- Phase 1: Initiates the logical connection and establishes its security capabilities
- Phases 2 and 3: Performs key exchange. The messages and message content used in this phase depends on the handshake variant negotiated in phase 1.
- Phase 4: Completes the setting up of a secure connection.



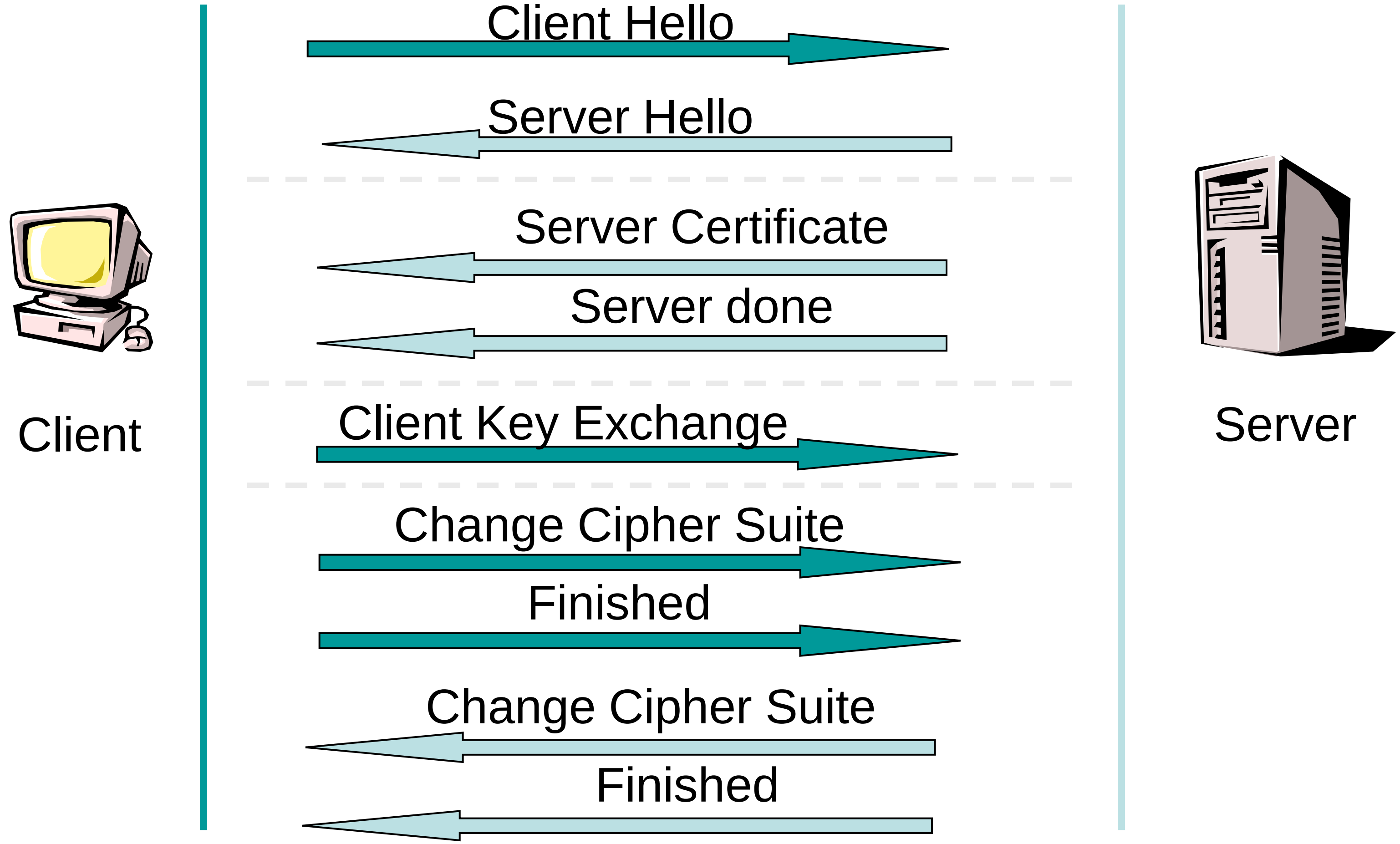
TLS: Simplified RSA-based Handshake

- The simplest variation provides server-only authentication in the presence of a server public key suitable for RSA encryption.
- On completion of Phase 1, assume that RSA-based Key Exchange has been selected.



Diagram

TLS:
Simplified RSA-based Handshake



TLS: Simplified RSA-based Handshake

- Client hello
  - Advertises available cipher suites (e.g. RSA, RC4/40, MD5)
  - Checks for previous session ID
- Server hello
  - Returns the selected cipher suite
  - Server adapts to client capabilities
- Server Certificate
  - X.509 digital certificate sent to client.
  - Client verifies the certificate including that the certificate signer is in its acceptable Certificate Authority (CA) list. Now the client has the server’s certified public key.
- Server done
  - To indicate that the server has finished sending messages.

TLS:
Simplified RSA-based Handshake

- Client Key Exchange
  - Client selects a random ‘pre-master secret’
  - Client encrypts the ‘pre-master secret’ using the server’s public key
  - Client sends the encrypted ‘pre-master secret’ to the server.
  - Server decrypts using its private key to recover the ‘pre-master secret’.
  - Both parties now compute the master secret using the pre-master secret and other exchanged values.
- Completion
  Both client and server exchange finished and change_cipher_spec messages to complete the authentication phase.

SSL and TLS Limitations

- Higher layers should not be overly reliant on SSL or TLS always negotiating the strongest possible connection between two peers
- There are a number of ways a ‘man in the middle’ attacker can attempt to make two entities drop down to the least secure method they support.
- For example, an attacker could block access to the port a secure service runs on, or attempt to get the peers to negotiate an unauthenticated connection.

SSL and TLS Limitations

- Applications should never transmit information over a channel less secure than they require.
- SSL and TLS are only as secure as the cryptographic algorithms determined in the handshake protocol.
- Both require a secure web browser and a secure operating system to be ‘secure’
  - Do these things actually exist?

IP Layer Security

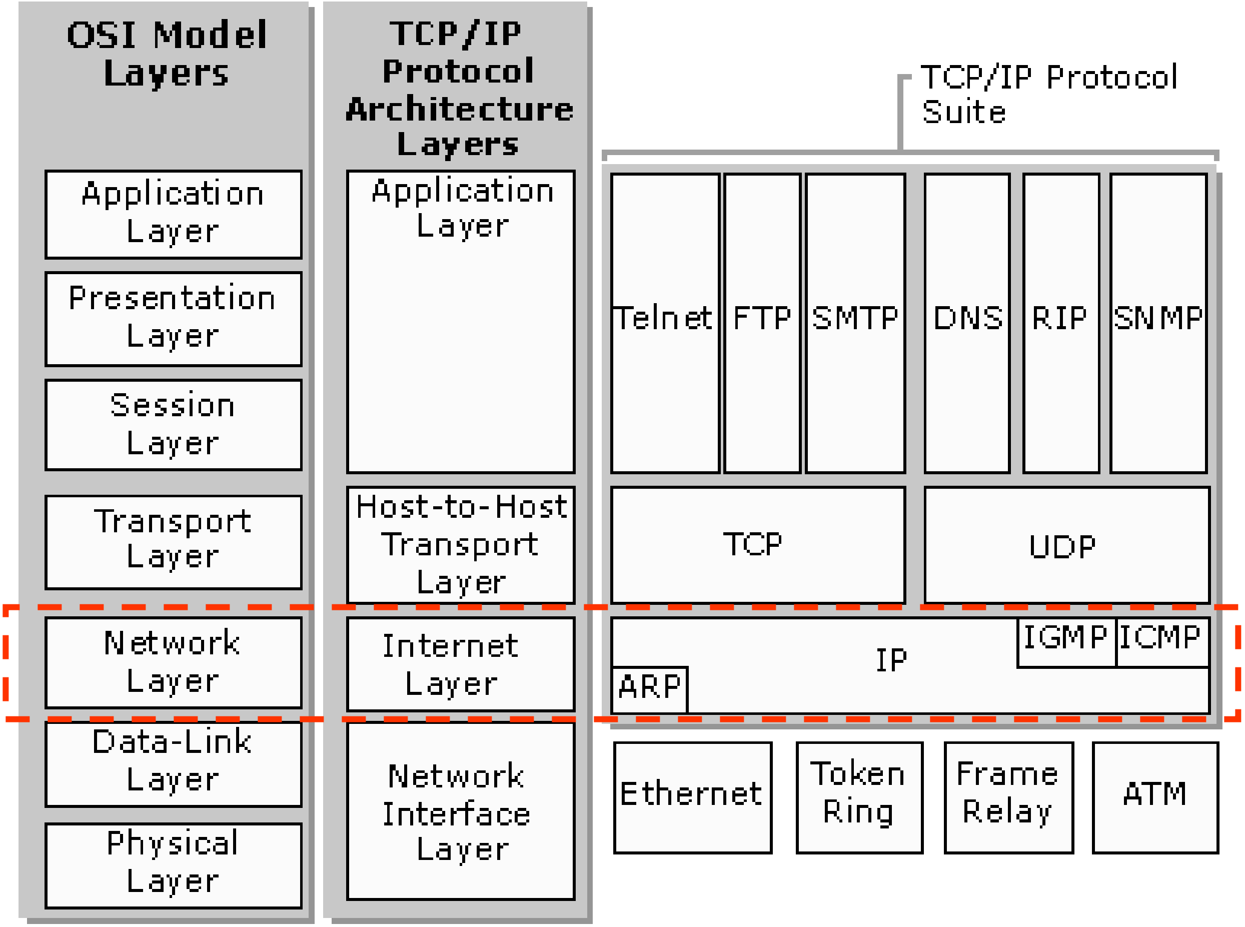
IPSec & Virtual Private Networks

IPSec: Introduction

- Internet Protocol security (IPSec) is standard for secure communications over Internet Protocol (IP) networks, through the use of cryptographic security services.
- Uses encryption, authentication and key management algorithms
- Based on an end-to-end security model at the IP level
- Internet Engineering Task Force (IETF) Working Group
- URL: http://www.ietf.org/html.charters/ipsec-charter.html
- Provides a security architecture for both IP V4 and IP V6
  - Mandatory for IPv6
  - Optional for IPv4

Layer 3 Security

IP Sec Operation →



IPSec:
Security Services

- Message Confidentiality.
  - Protects against unauthorized data disclosure.
  - Accomplished by the use of encryption mechanisms.
- Traffic Analysis Protection.
  - A person monitoring network traffic cannot know which parties are communicating, how often, or how much data is being sent.
  - Provided by concealing IP datagram details such as source and destination address.
- Message Integrity.
  - IPsec can determine if data has been changed (intentionally or unintentionally) during transit.
  - Integrity of data can be assured by using a MAC.

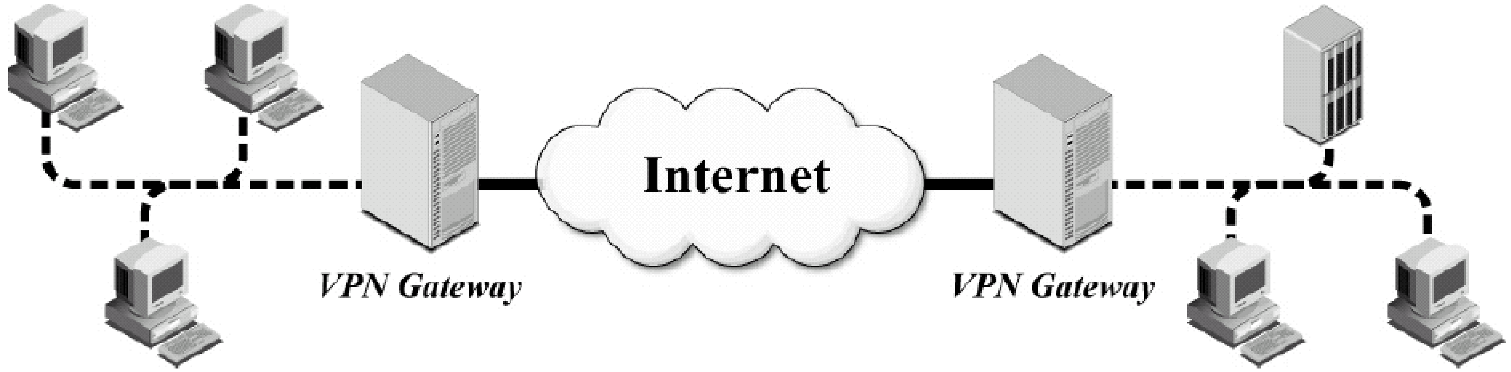
IPSec:
Security Services

- Message Replay Protection.
  - The same data is not delivered multiple times, and data is not delivered grossly out of order.
  - However, IPsec does not ensure that data is delivered in the exact order in which it is sent.
- Peer Authentication.
  - Each IPsec endpoint confirms the identity of the other IPsec endpoint with which it wishes to communicate.
  - Ensures that network traffic is being sent from the expected host.
- Network Access Control.
  - Filtering can ensure users only have access to certain network resources and can only use certain types of network traffic.

IPSec: Common Architectures

- Gateway-to-Gateway Architecture
- Host-to-Gateway Architecture
- Host-to-Host Architecture

IPSec: Gateway-to-Gateway Architecture

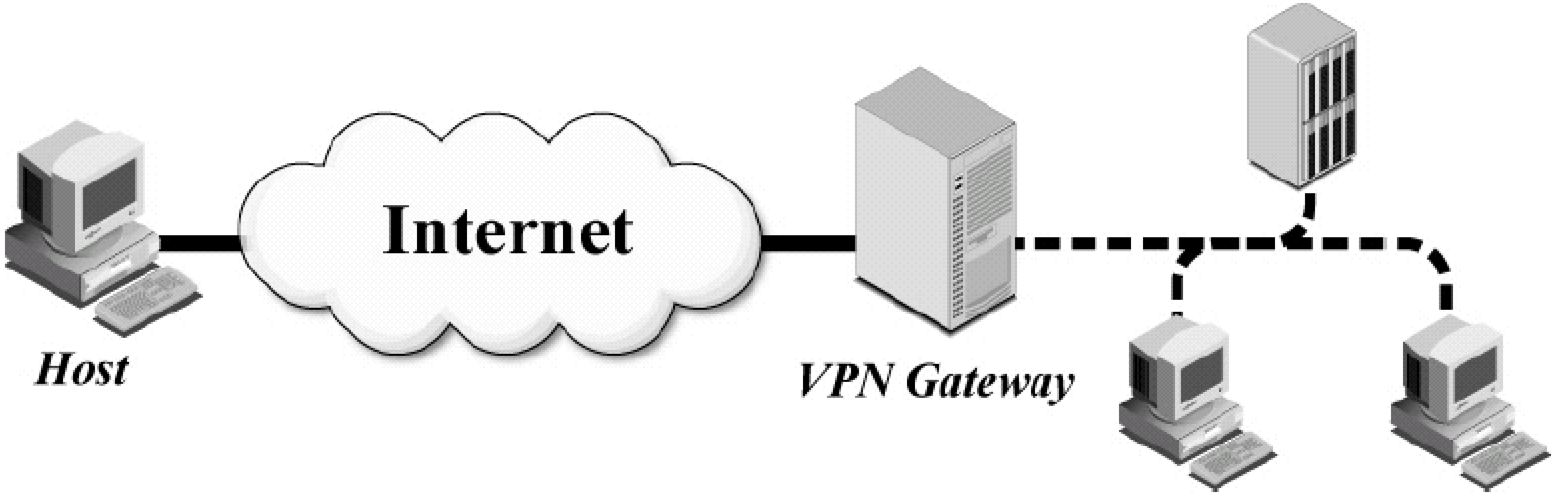


Source: NIST Special Publication 800-77

IPSec: Gateway-to-Gateway Architecture

- Provides secure network communications between two networks.
- Establish a VPN connection between the two gateways.
- Network traffic is routed through the IPsec connection, protecting it appropriately.
- Only protects data between the two gateways.
- most often used when connecting two secured networks, such as linking a branch office to headquarters over the Internet.
- Gateway-to-gateway VPNs often replace more costly private wide area network (WAN) circuits.

IPSec: Host-to-Gateway Architecture

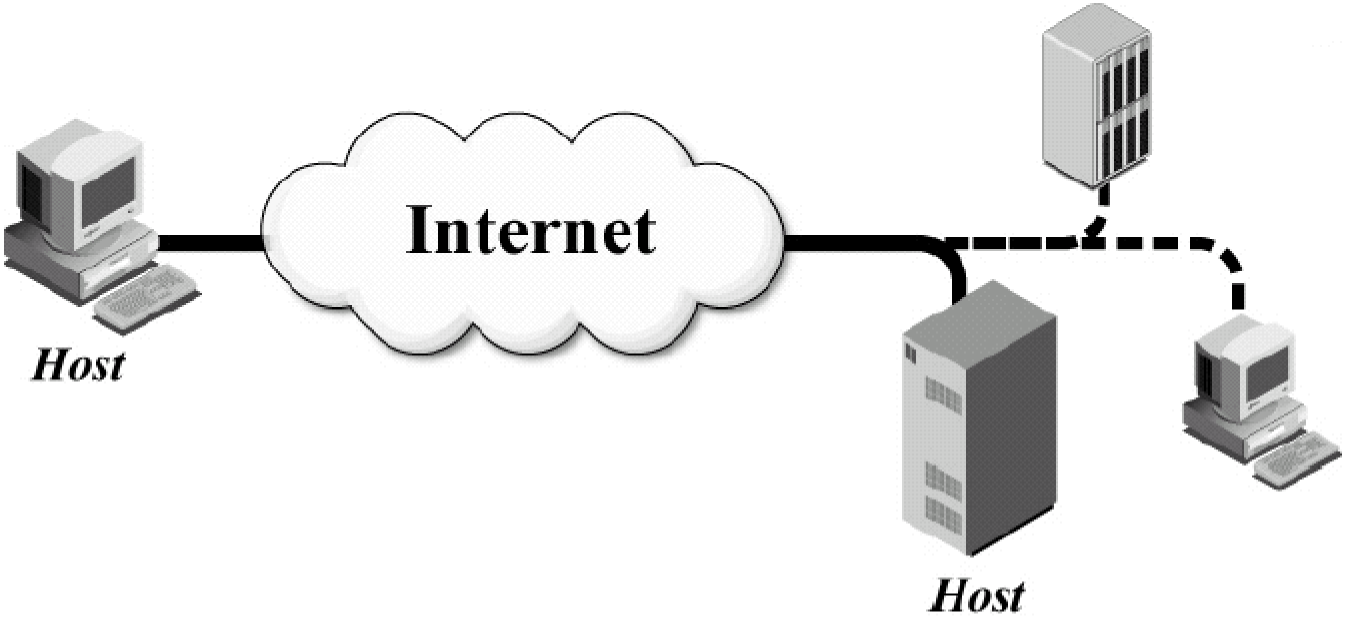


Source: NIST Special Publication 800-77

IPSec: Host-to-Gateway Architecture

- Commonly used to provide secure remote access.
- The organization deploys a VPN gateway onto their network; each remote access user then establishes a VPN connection between the local computer (host) and the VPN gateway.
- As with the gateway-to-gateway model, the VPN gateway may be a dedicated device or part of another network device.
- Most often used when connecting hosts on unsecured networks to resources on secured networks, such as linking travelling employees around the world to headquarters over the Internet.

IPSec:
Host-to-Host Architecture



Source: NIST Special Publication 800-77

IPSec: Host-to-Host Architecture

- Typically used for special purpose needs, such as system administrators performing remote management of a single server.
- Only model that provides end-to-end protection for data throughout its transit.
- Resource-intensive to implement and maintain in terms of user and host management.
- All user systems and servers that will participate in VPNs need to have VPN software installed and/or configured.
- Key establishment is often accomplished through a manual process.

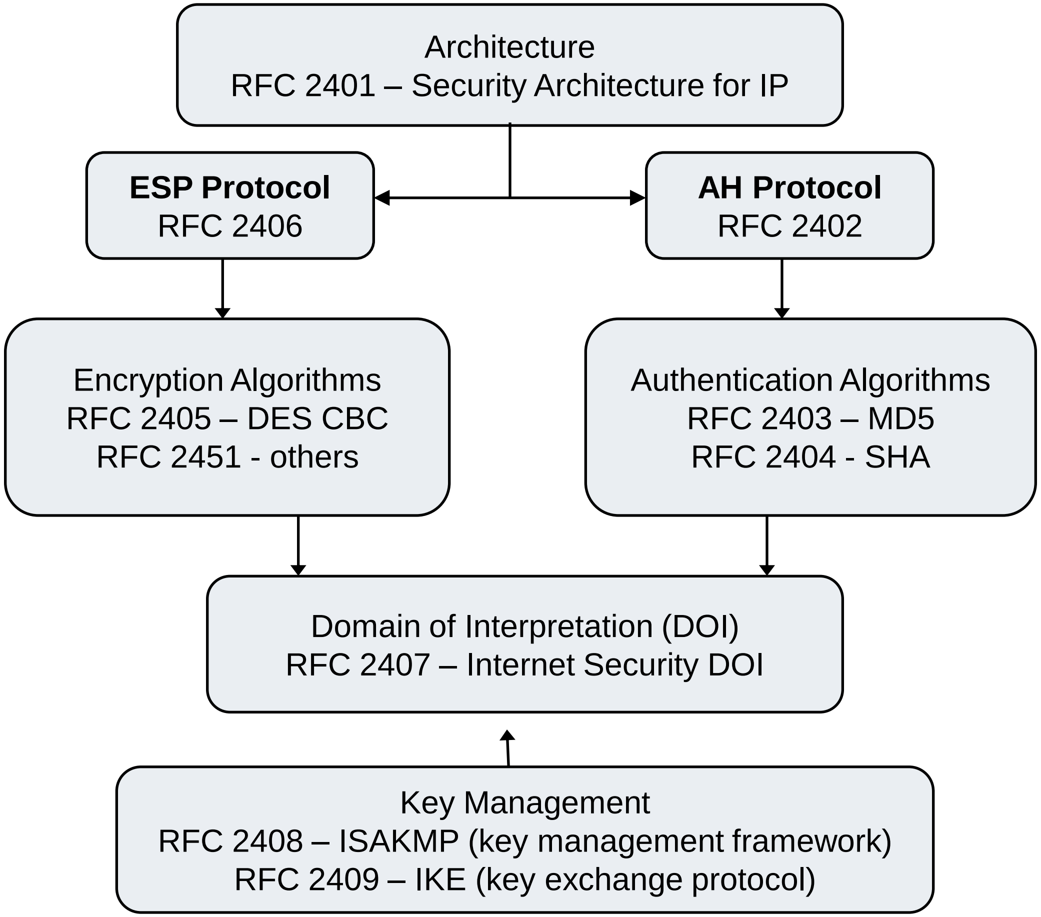
IPSec: Benefits

- If applied at a firewall/router, strong security applies to all traffic crossing this boundary.
  - Internal workstations need not be reconfigured.
- Is transparent to applications.
  - Operates at layer 3 so applications are not aware of its operation.
- Can be transparent to end users.
  - System administrator configures IPSec;
    the end user is not involved.
- Can provide security for individual users.
  - Can be configured on specific systems.

IPSec:
Architectural Overview

- IPSec is a framework for security that operates at the Network Layer.
- It extends the IP packet header (using additional protocol numbers, not options).
- Provides the ability to encrypt any higher layer protocol, including arbitrary TCP and UDP sessions.
- Requires operating system support, not application support.

IPSec:
Document Roadmap



Architecture
RFC 2401 – Security Architecture for IP

ESP Protocol
RFC 2406

AH Protocol
RFC 2402

Encryption Algorithms
RFC 2405 – DES CBC
RFC 2451 - others

Authentication Algorithms
RFC 2403 – MD5
RFC 2404 - SHA

Domain of Interpretation (DOI)
RFC 2407 – Internet Security DOI

Key Management
RFC 2408 – ISAKMP (key management framework)
RFC 2409 – IKE (key exchange protocol)

IPSec: Protocols Types

- Encapsulating Security Payload (ESP)
  - Confidentiality, authentication, integrity and replay protection
- Authentication Header (AH)
  - Authentication, integrity and replay protection. However there is no confidentiality
- Internet Key Exchange (IKE)
  - negotiate, create, and manage security associations

Setting up an IPSec connection

- IPSec protocols need to perform key exchange over an insecure channel.
- Requires complex protocols and processor intensive operations.
- This is largely automated after initial manual configuration by administrator prior to connection setup.
- (See ISAKMP, IKE, Oakley, Skeme and SAs)

Security Associations

- A security association (SA) contains info needed by an IPSec endpoint to support one end of an IPSec connection.
- Can include cryptographic keys and algorithms, key lifetimes, security parameter index (SPI), and security protocol identifier (ESP or AH).
- The SPI is included in the IPSec header to associate a packet with the appropriate SA.
- The SA tells the endpoint how to process inbound IPSec packets or how to generate outbound packets.

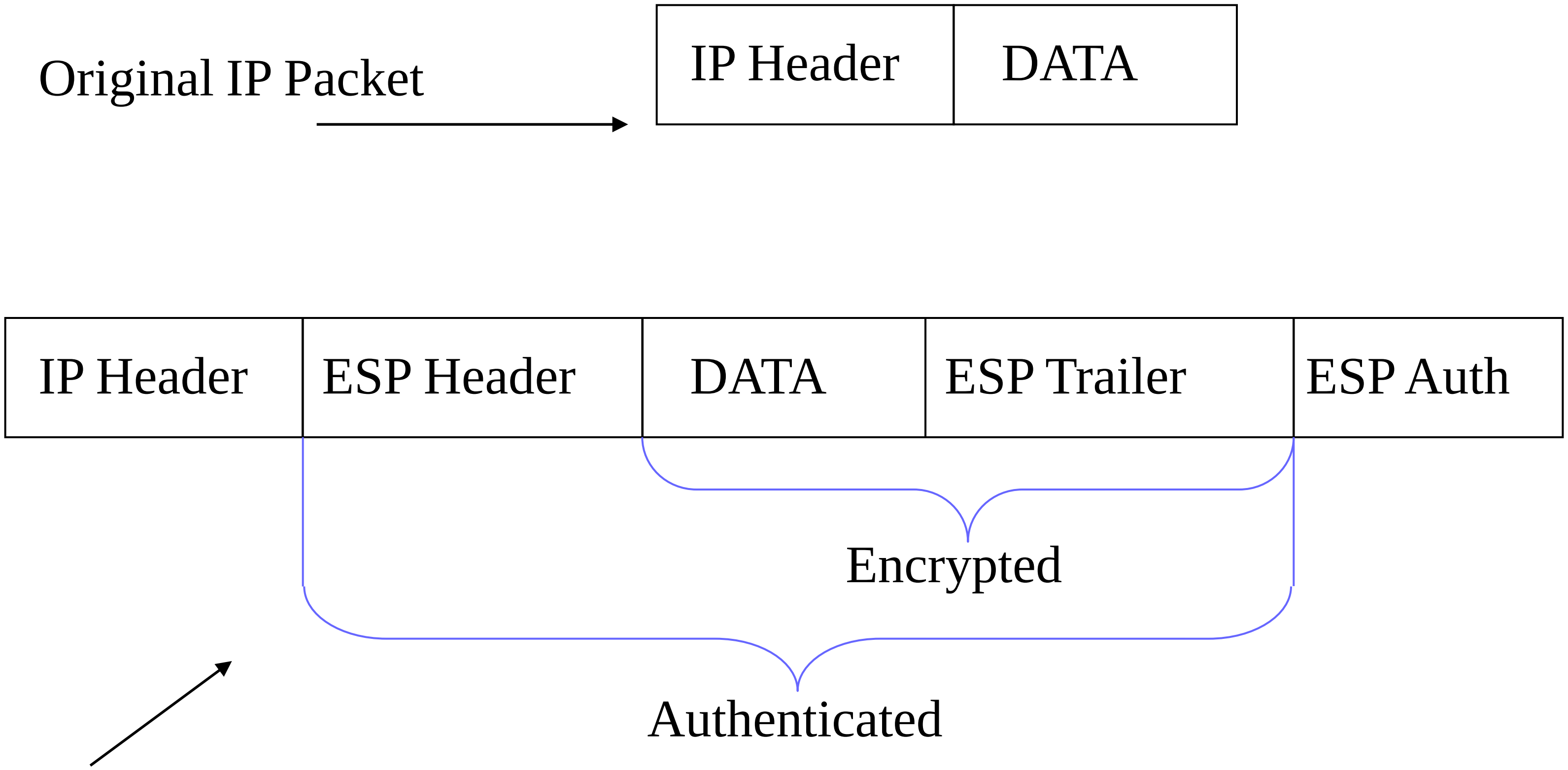
Security Associations

- Security Associations are simplex
  - need one for each direction of connection
- An SA is applied to AH or ESP, if both AH and ESP protection is provided there must be separate SA's for each.
- SA's are stored in a security association database (SAD).

IPSec:
Modes of operation

- Each protocol (ESP or AH) can operate in transport or tunnel mode.
- Transport mode:
  - Operates primarily on the payload (data) of the original packet.
  - Generally only used in host-to-host architectures.
- Tunnel mode:
  - Original packet encapsulated into a new one, payload is original packet.
  - Typical use is gateway-to-gateway architecture.

Transport Mode ESP

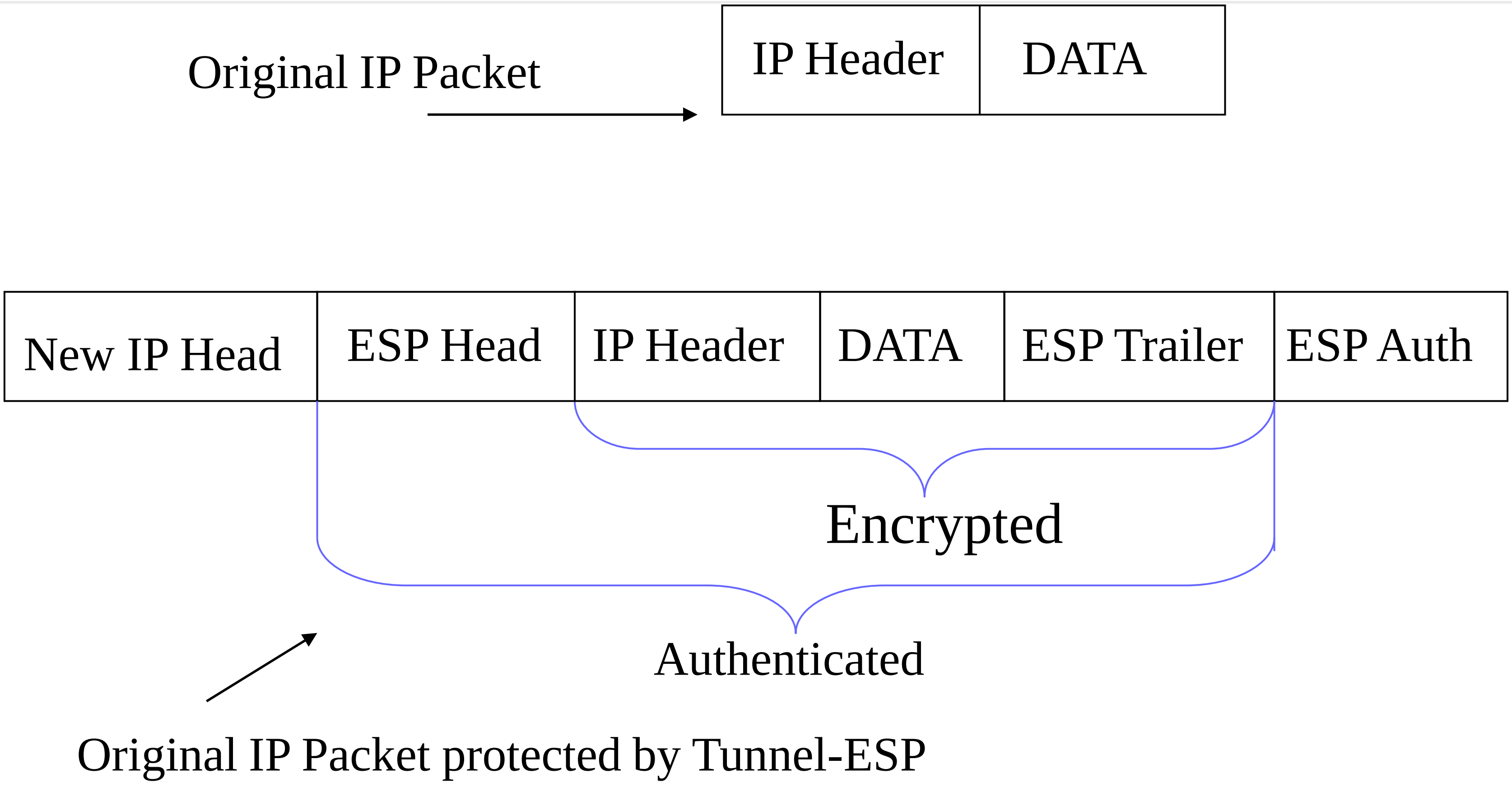


Original IP Packet protected by Transport-ESP

IPSec - ESP in Transport Mode: Outbound Packet Processing

- The data after the original IP header is padded by adding an ESP trailer and the result is then encrypted using the symmetric cipher and key in the SA.
- An ESP header is prepended.
- If an SA uses the authentication service, an ESP MAC is calculated over the data prepared so far and appended.
- The original IP header is prepended.
- However, some fields in the original IP header must be changed. For example,
  - Protocol field changes from TCP to ESP.
  - Total Length field must be changed to reflect the addition of the AH header.
  - Checksums must be recalculated.

Tunnel Mode ESP



IPSec - ESP in Tunnel Mode: Outbound Packet Processing

- The entire original packet is padded by adding an ESP trailer and the result is then encrypted using the symmetric cipher and key agreed in the SA.
- An ESP header is prepended.
- If an SA uses the authentication service, an ESP MAC is calculated over the data prepared so far and appended.
- A new ‘outer’ IP header is prepended.
  - The ‘inner’ IP header of the original IP packet carries the ultimate source and destination addresses.
  - The ‘outer’ IP header may contain distinct IP addresses such as addresses of security gateways.
  - The ‘outer’ IP header Protocol field is set to ESP.

Endnote

- Security protocols rely on a set of assumptions
  - Trust in involved parties (i.e. that they do what is expected of them)
  - Shared secrets (e.g. that keys are securely set up)
  - Strength of cryptographic functions
- Bad implementation of a theoretically secure protocol makes the system vulnerable
- The human factor must be considered whenever humans are involved in the protocol execution
  - Poor security usability causes security vulnerabilities