الجامعة الافتراضية السورية

أمن الشبكات
Network and Infrastructure Security

د. محمد العصورة

Dr. Mohammed Assora

دكتوراه في امن شبكات الحواسب

PhD. In Computer Network Security

Objectives of Lecture

Investigate how IPSec, SSL/TLS, and SSH provide security at the transport, network and application layers.

- Study major applications of these protocols in e-commerce and remote administration.

Contents

IPSec

SSL/TLS

SSH

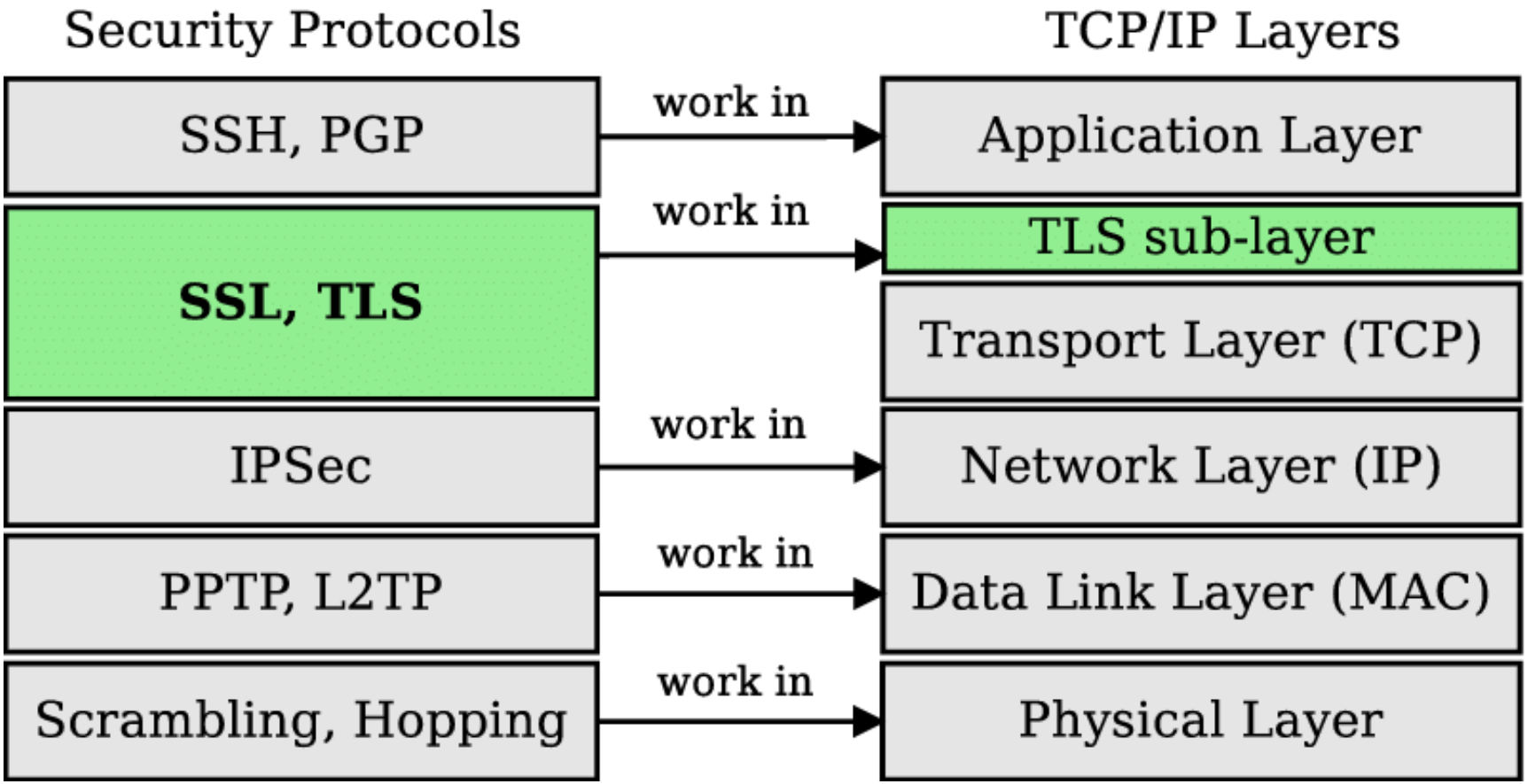
Web Security protocols

Network security (or web security) protocols are used to protect computer data and communication in transit

A number of approaches to providing network security are possible.

The various approaches that have been considered are similar in the services they provide, but they differ with respect to their scope of applicability and their relative location within the TCP/IP protocol stack.

Relative location of Security protocols in TCP/IP stack



IPSec

- Internet Standard since 1998
- References
  - RFC 4301, “Security Architecture for the Internet Protocol”
  - RFC 4302, “IP Authentication Header”
  - RFC 4303, “IP Encapsulating Security Payload”
  - RFC 7296, “Internet Key Exchange (IKEv2) Protocol”

IPSec Application

- IPSec protocols designed for both
  - IPv4 (optional support)
  - IPv6 (mandatory support)

  as extension headers
- Possible applications:
  - Virtual Private Networks (VPN) over the Internet
    - A company can build a secure network, built over the public Internet, with private access
  - Secure remote access over the Internet
    - An end user may gain access to a company network

IPSec Benefits

- Benefits of IPSec
  - transparent to applications
    - TCP/UDP API unchanged
  - transparent to users
    - no need to train users on security issues
  - IPSec implemented in a firewall or router
    - security for all traffic crossing the perimeter
    - no need for change software; conversion is done by the firewall/router

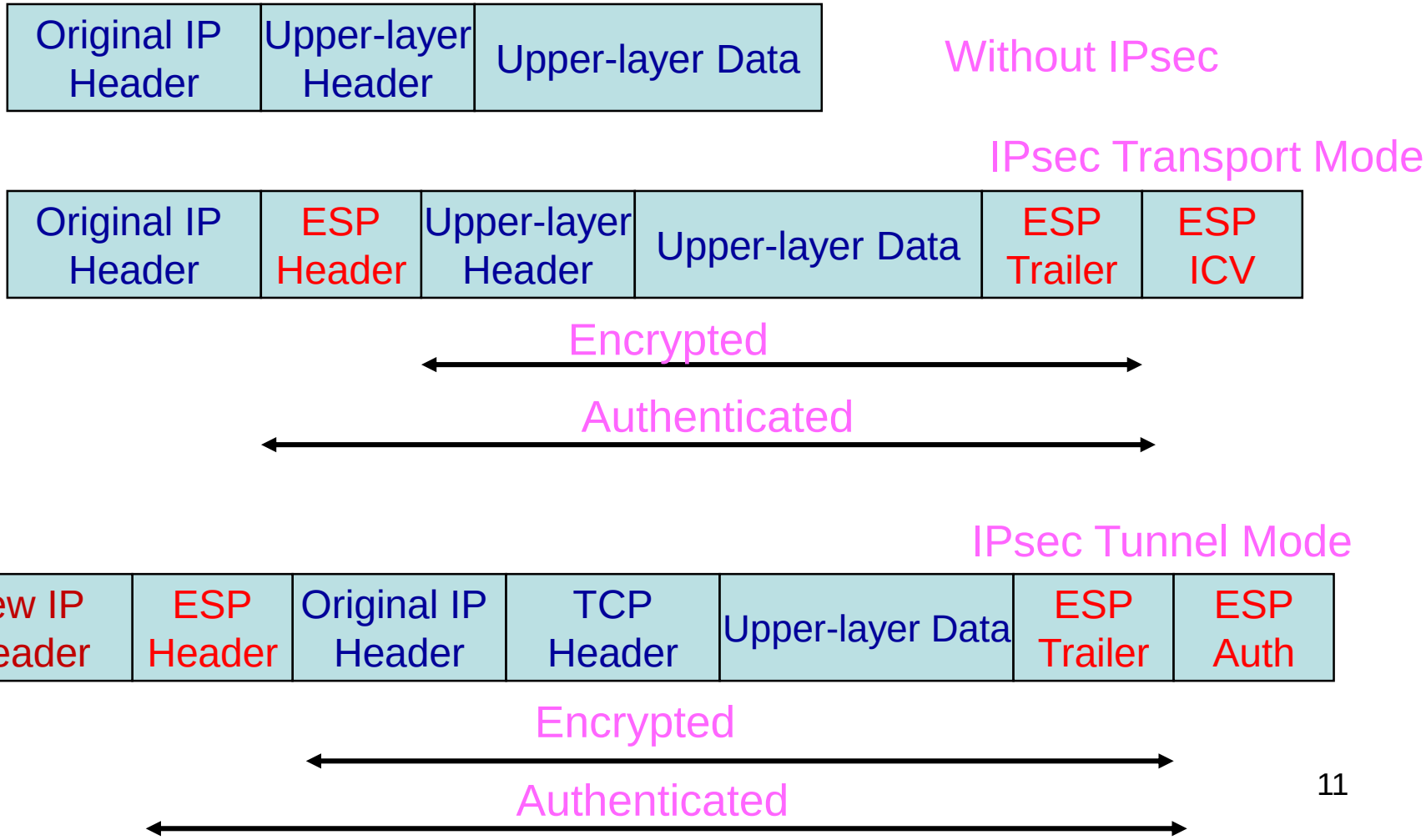
IPSec Protocols

- IPSec Protocols:
  - Authentication Header (AH) for authentication
  - Encapsulating Security Payload (ESP) for encryption
    - Note: ESP can also have authentication
  - Internet Security and Key Management Protocol (ISAKMP/Oaxley) for key management

IPSec Modes

- Both AH and ESP supports two mode of use
  - Transport mode
  - Tunnel mode

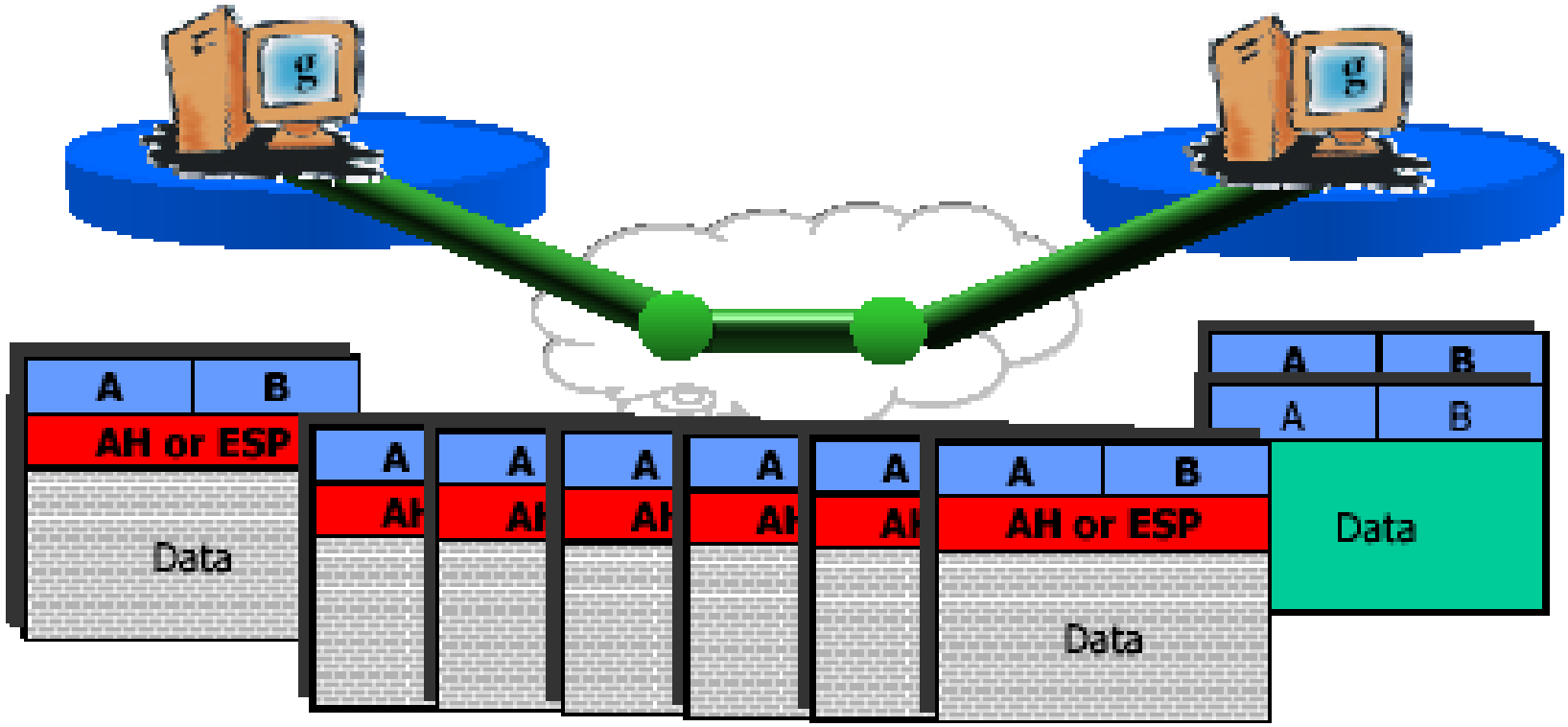
IPsec authentication and encryption



IPSec Transport Mode

- Transport Mode
  - provides protection
    - to the upper-layer protocols
    - in other words, to the payload of IP packets
  - normally used for end-to-end communication
- ESP in Transport Mode
  - encrypts and optionally authenticates the IP payload
  - IP header not protected

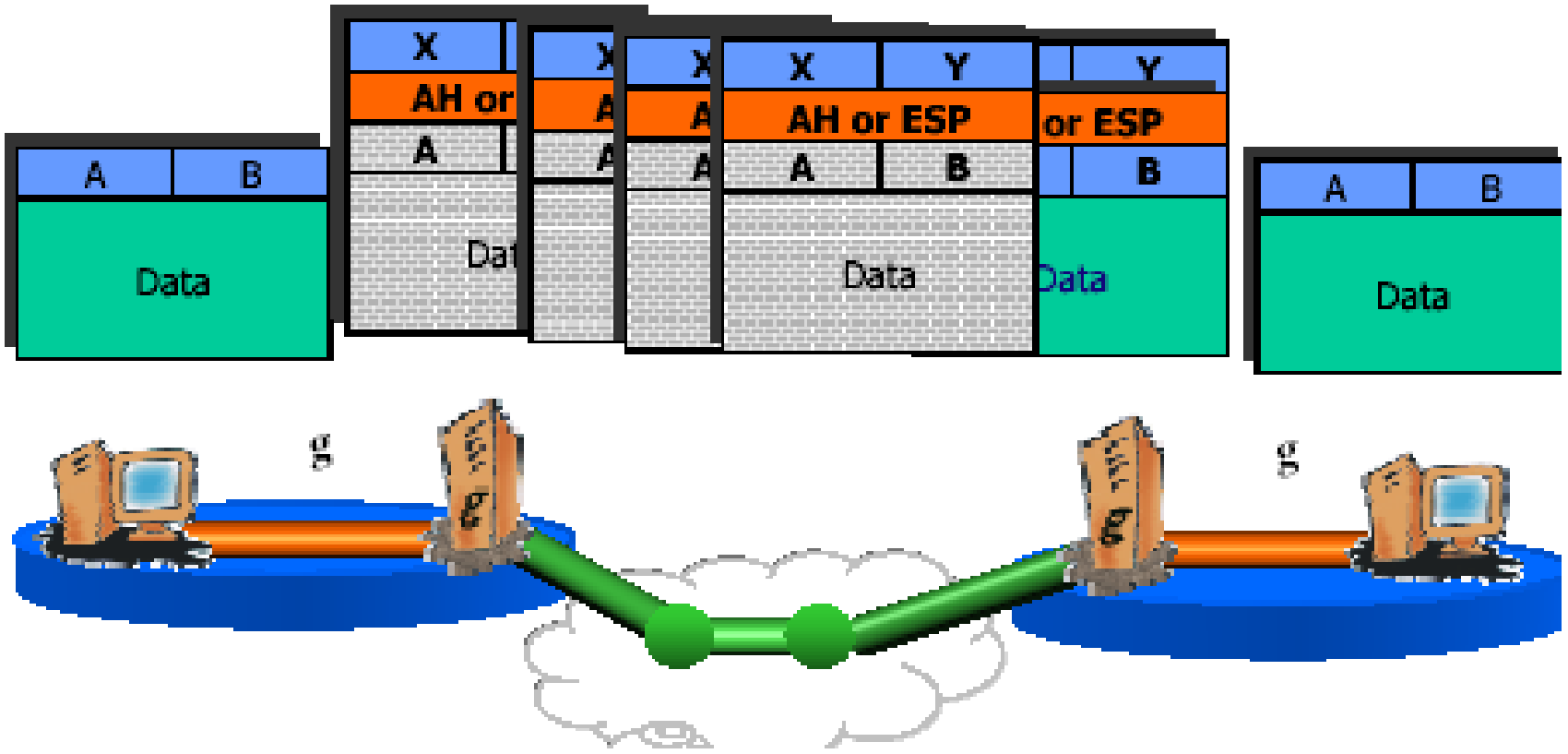
IPSec Transport Mode



IPSec Tunnel Mode

- Tunnel mode
  - provides protection to the entire IP packet
  - normally used in “gateway-to-gateway” communication
- How does it work?
  - AH/ESP headers are added to the IP packet
  - the entire packet is treated as the payload of a new outer IP packet with a new outer IP header
- Packets travel through a “tunnel”
  - No routers along the way are able to examine the original packet

IPSec Tunnel Mode



Tunneling Example

- Tunneling: example
  - Host A on network NA generates an IP packet addressed to host B on network NB
  - The packet is routed to the firewall of network NA
  - The firewall encapsulates the packet in an outer IP header
  - The new packet is routed to the firewall of network NB
  - The firewall extracts, decrypts and authenticates the original packet
  - The original packet is routed and delivered to B

IPSec Key Management

- Key Management in IPSec
  - Determination and distribution of secret keys
  - Four keys for each pair of communication endpoints:
    - transmit and receive with AH
    - transmit and receive with ESP
- Two supports for key management:
  - Manual
    - Performed by system administrators
  - Automated
    - Based on Diffie-Hellman secret key exchange protocols

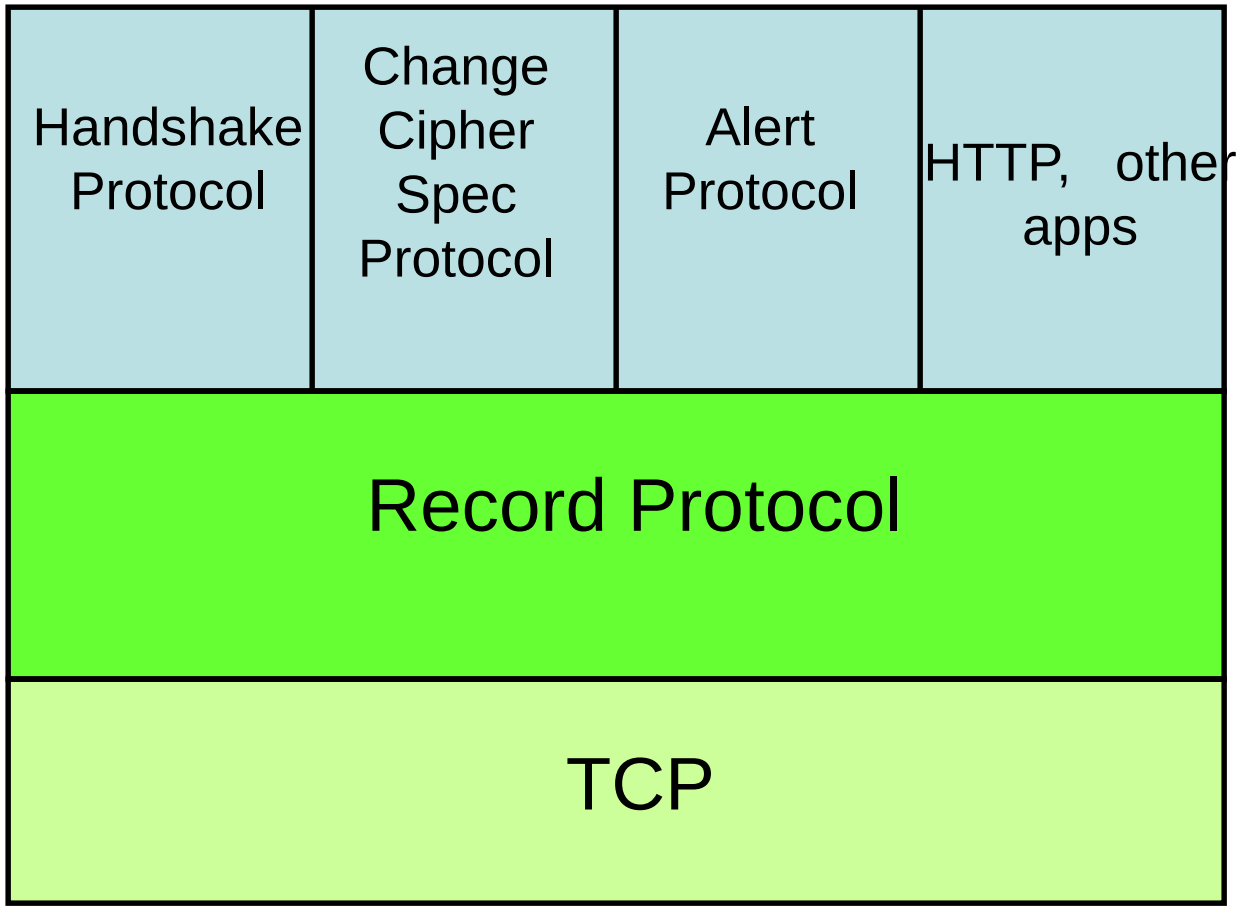
SSL/TLS Overview

- SSL = Secure Sockets Layer.
- TLS = Transport Layer Security.
  - Defined in RFC 4346.
  - Open-source implementation at http://www.openssl.org/.
- SSL/TLS provides security ‘at TCP layer’.
  - Uses TCP to provide reliable, end-to-end transport.
  - Applications need some modification.
  - In fact, usually a thin layer between TCP and HTTP.

SSL/TLS Basic Features

- SSL/TLS widely used in Web browsers and servers to support ‘secure e-commerce’ over HTTP.
  - Built into Microsoft IE, Netscape, Mozilla, Apache, IIS,…
  - The (in)famous browser lock.
- TLS architecture provides two layers:
  - TLS Record Protocol
    - Provides secure, reliable channel to upper layer.
  - Upper layer carrying:
    - TLS Handshake Protocol, Change Cipher Spec. Protocol, Alert Protocol, HTTP, any other application protocols.

TLS Protocol stack



TLS Record Protocol

- Provides secure, reliable channel to upper layer.
- Carries application data and TLS ‘management’ data.
- TLS Record Protocol provides:
  - Data origin authentication and integrity: MAC using algorithm similar to HMAC. Based on MD-5 or SHA-1 hash algorithms. MAC protects 64 bit sequence number for anti-replay.
  - Confidentiality. Bulk encryption using symmetric algorithm. IDEA, DES, 3DES, AES,…

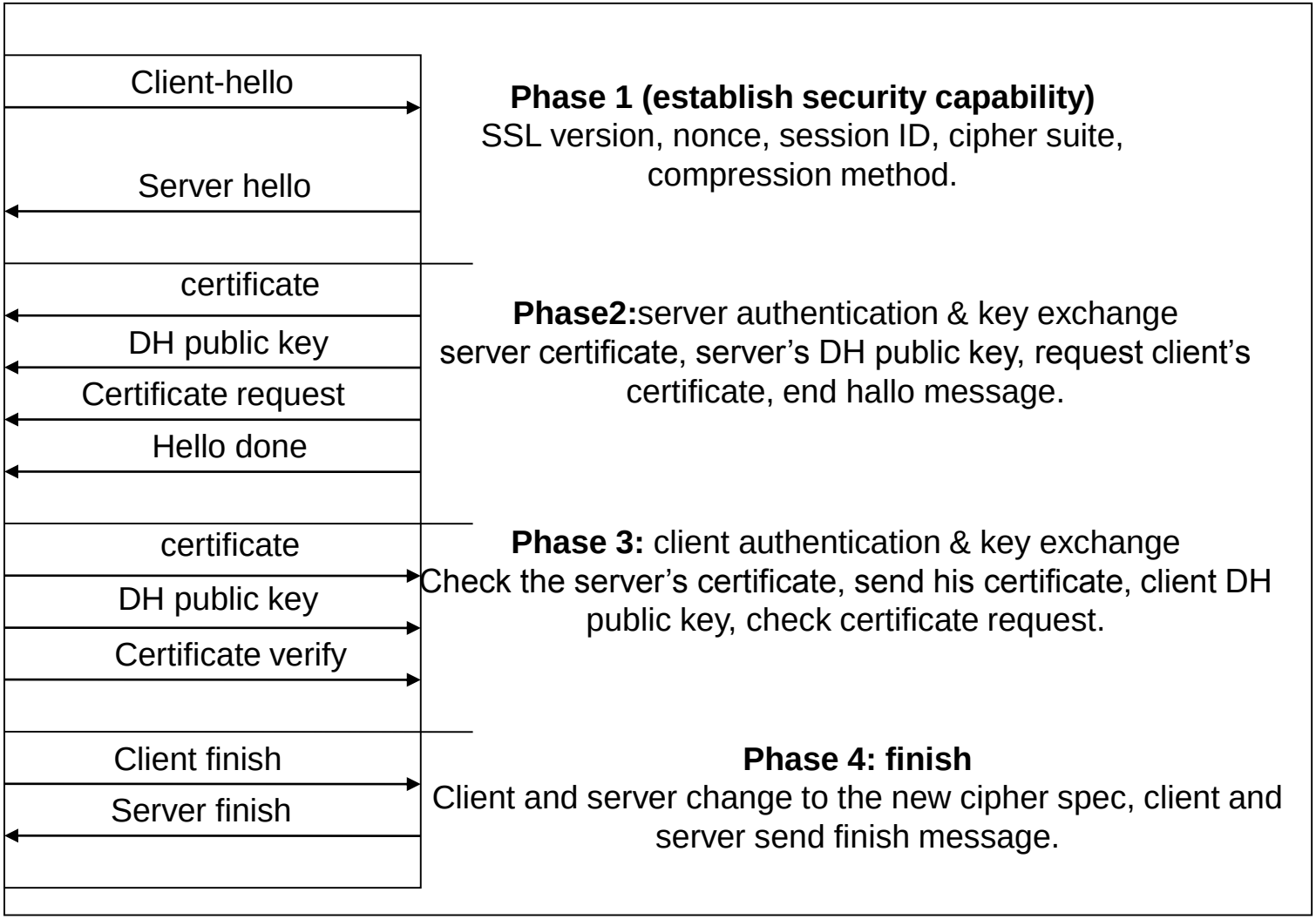
- Data from application/upper layer TLS protocol partitioned into fragments (max size 2^{14} bytes).
- MAC first then pad (if needed), finally encrypt.
- Prepend header.
  - Content type, version, length of fragment.
- Submit to TCP.

TLS Handshake Protocol

- Like IPSec, TLS needs symmetric keys:
  - MAC and encryption at Record Layer.
  - Different keys in each direction.
- These keys are established as part of the TLS Handshake Protocol.
- As with IKE in IPSec, the TLS Handshake Protocol is a complex protocol with many options…

TLS Handshake Protocol Security Goals

- Entity authentication of participating parties.
  - Participants are called ‘client’ and ‘server’.
  - Server nearly always authenticated, client more rarely.
  - Appropriate for most e-commerce applications.
- Establishment of a fresh, shared secret.
  - Shared secret used to derive further keys.
  - For confidentiality and authentication in TLS Record Protocol.
- Secure ciphersuite negotiation.
  - Encryption and hash algorithms
  - Authentication and key establishment methods.


Client-hello
Server hello
certificate
DH public key
Certificate request
Hello done
certificate
DH public key
Certificate verify
Client finish
Server finish
Phase 1 (establish security capability)
SSL version, nonce, session ID, cipher suite, compression method.
Phase2:server authentication & key exchange
server certificate, server's DH public key, request client's certificate, end hallo message.
Phase 3: client authentication & key exchange
Check the server's certificate, send his certificate, client DH public key, check certificate request.
Phase 4: finish
Client and server change to the new cipher spec, client and server send finish message.

SSL/TLS Applications

- Secure e-commerce using SSL/TLS.
  - Client authentication not needed until client decides to buy something.
  - TLS provides secure channel for sending credit card information, personal details, etc.
  - Client authenticated using credit card information, merchant bears (most of) risk.
- Very successful (amazon.com, on-line supermarkets, airlines,…)

SSL/TLS Applications

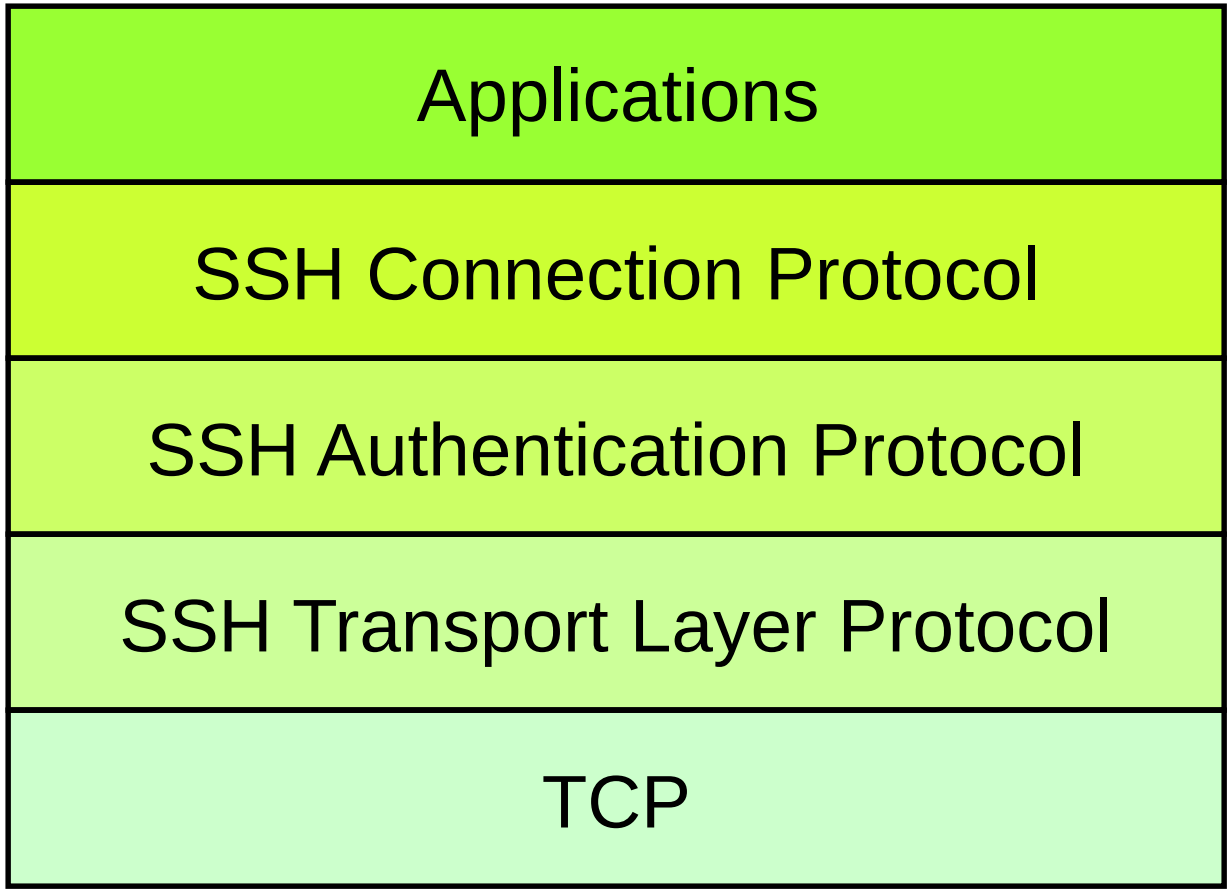
- Secure e-commerce: some issues.
  - No guarantees about what happens to client data (including credit card details) after session: may be stored on insecure server.
  - Does client understand meaning of certificate expiry and other security warnings?
  - Does client software *actually* check complete certificate chain?
  - Does the name in certificate match the URL of e-commerce site? Does the user check this?
  - Is the site the one the client thinks it is?
  - Is the client software proposing appropriate ciphersuites?

SSH Overview

- SSH = Secure Shell.
  - Initially designed to replace insecure rsh, telnet utilities.
  - Secure remote administration (mostly of Unix systems).
  - Extended to support secure file transfer and e-mail.
  - Latterly, provide a general secure channel for network applications.

- SSH provides security at Application layer.
  - Only covers traffic explicitly protected.
  - Applications need modification, but port-forwarding eases some of this.
  - Built on top of TCP, reliable transport layer protocol.

SSH-2 Architecture



Comparing IPSec, SSL/TLS, SSH

- All three have initial (authenticated) key establishment then key derivation.
- All protect ciphersuite negotiation.
- All three use keys established to build a ‘secure channel’.
- Operate at different network layers.
- can all be used to build VPNs.

Comparing IPsec, SSL/TLS, SSH

Security of all three undermined by:

- Implementation weaknesses.
- Weak server platform security.
  - Worms, malicious code, rootkits,…
- Weak user platform security.
  - Keystroke loggers, malware,…
- Limited deployment of certificates and infrastructure to support them.
  - Especially client certificates.
- Lack of user awareness and education.
  - Users click-through on certificate warnings.
  - Users fail to check URLs.
  - Users send sensitive account details to bogus websites (“phishing”) in response to official-looking e-mail.