CSE413 – Security of Information Systems 2020

PhD Furkan Gözükara, Toros University

https://github.com/FurkanGozukara/Security-of-Information-Systems-CSE413-2020

Lecture 10

Communications Security

Composed from Nils Gruschka, University of Oslo, Information Security 2018 Lectures

Outline

- Network security concepts
  - Communication security
  - Perimeter security
- Protocol architecture and security services
- Example security protocols
  - Transport Layer Security (TLS)
  - IP Layer Security (IPSec)
- VPN – Virtual Private Network

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:** Protection of data transmitted across networks between organisations and end users
  - – Topic for this lecture

- **Perimeter Security:** Protection of an organization's network from unauthorized access
  - – Topic for next lecture
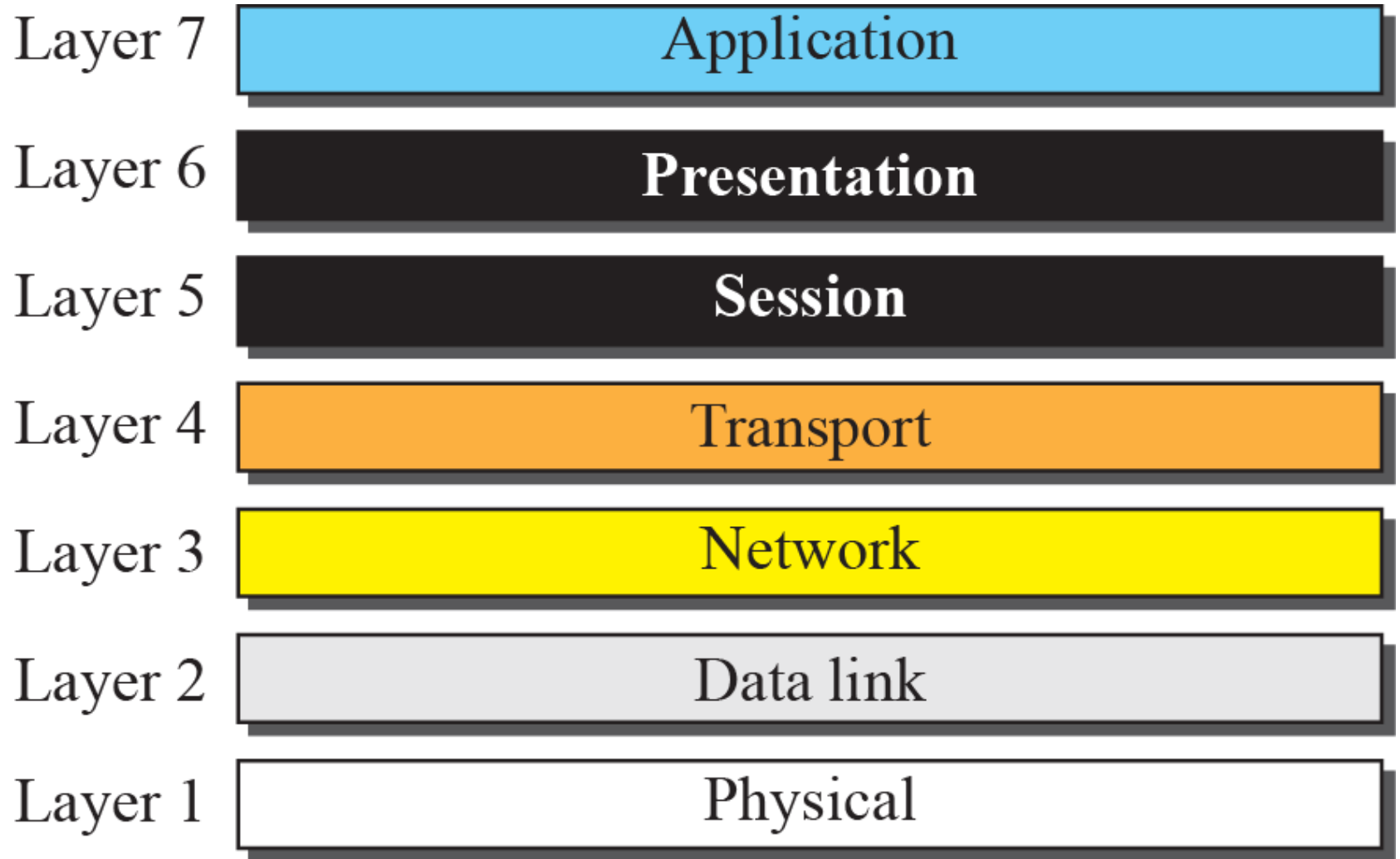
Communication Architecture
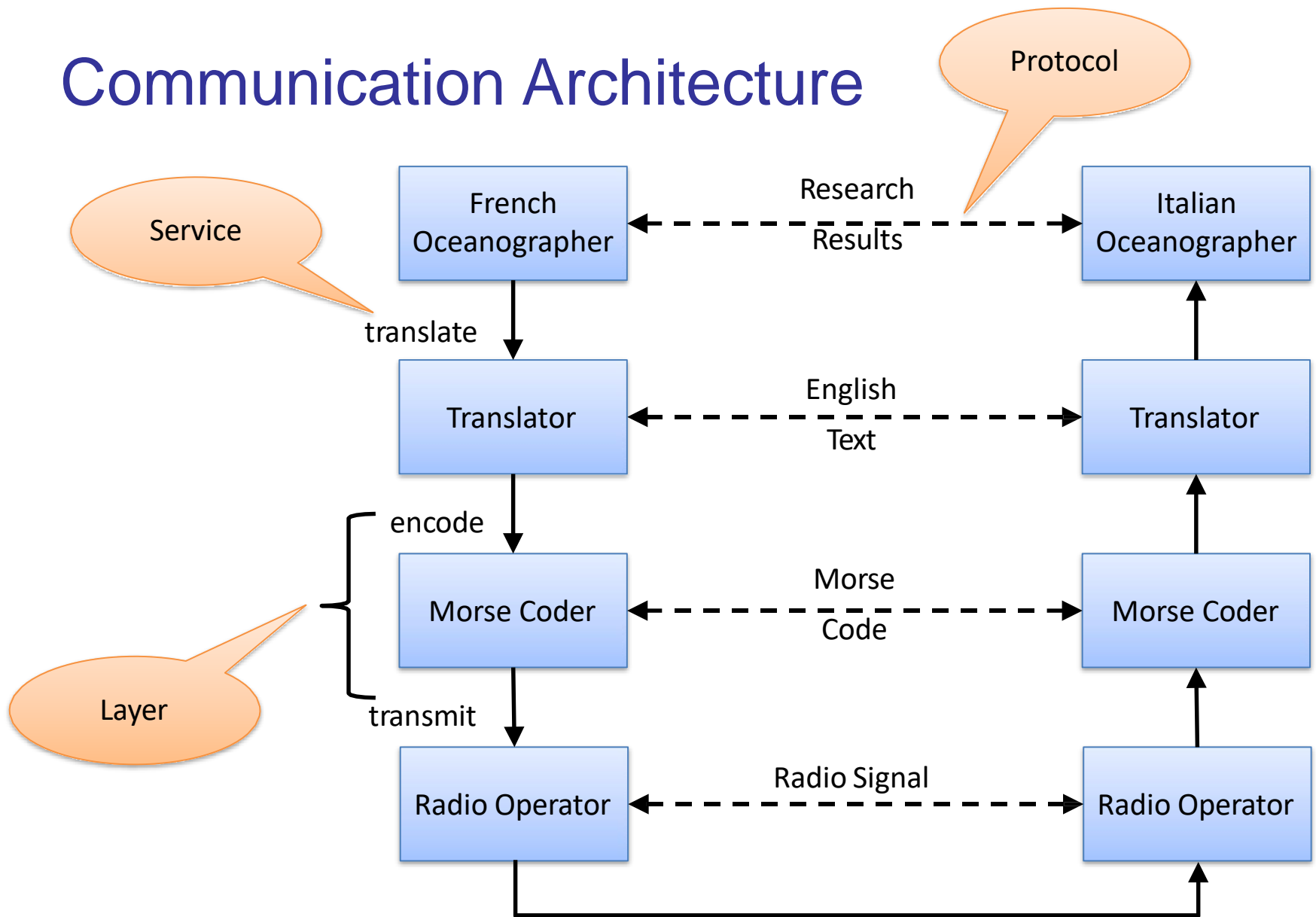
Communication Protocol Architecture

- Layered structure of hardware and software that supports the exchange of data between systems

- 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
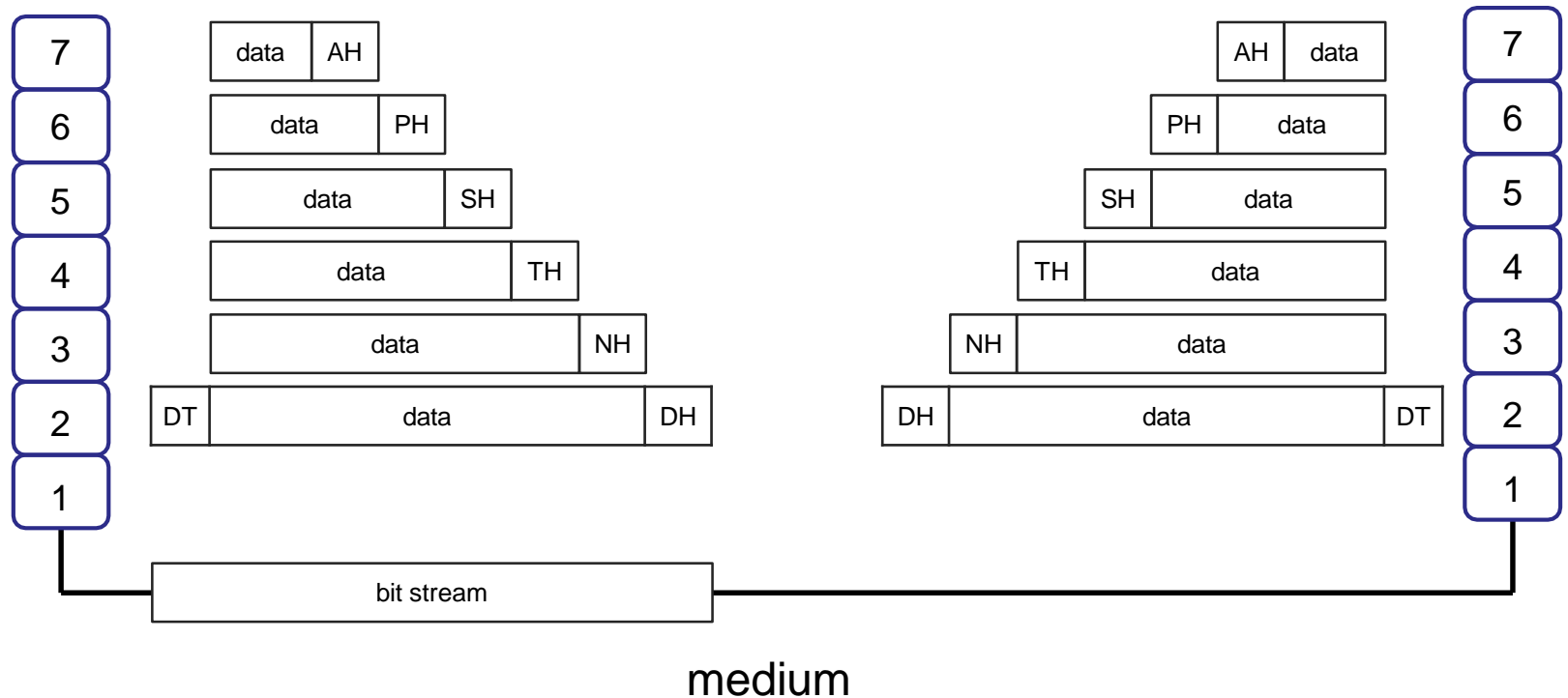 - TCP/IP protocol suite
    - Most widely used

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

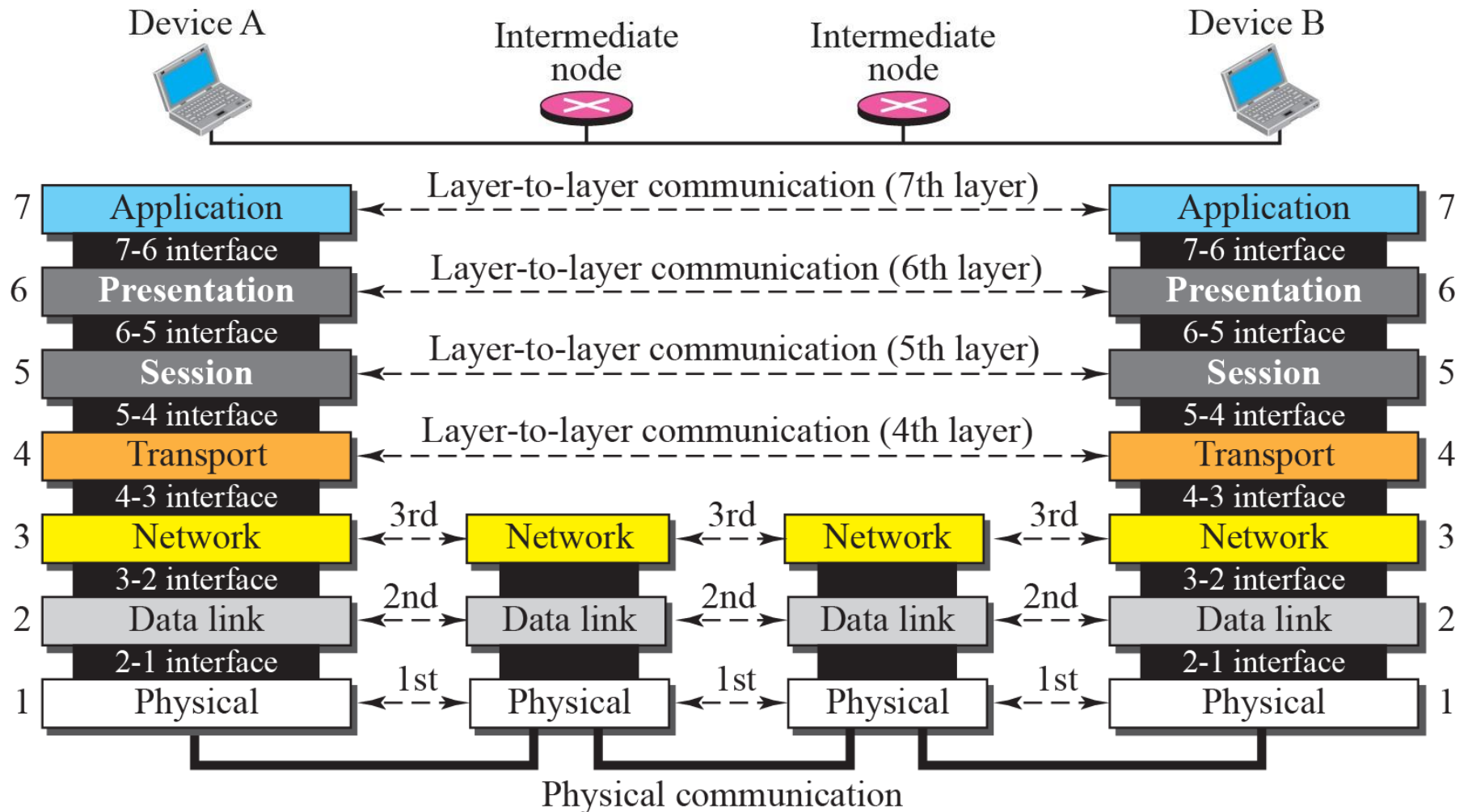# The OSI Protocol Stack

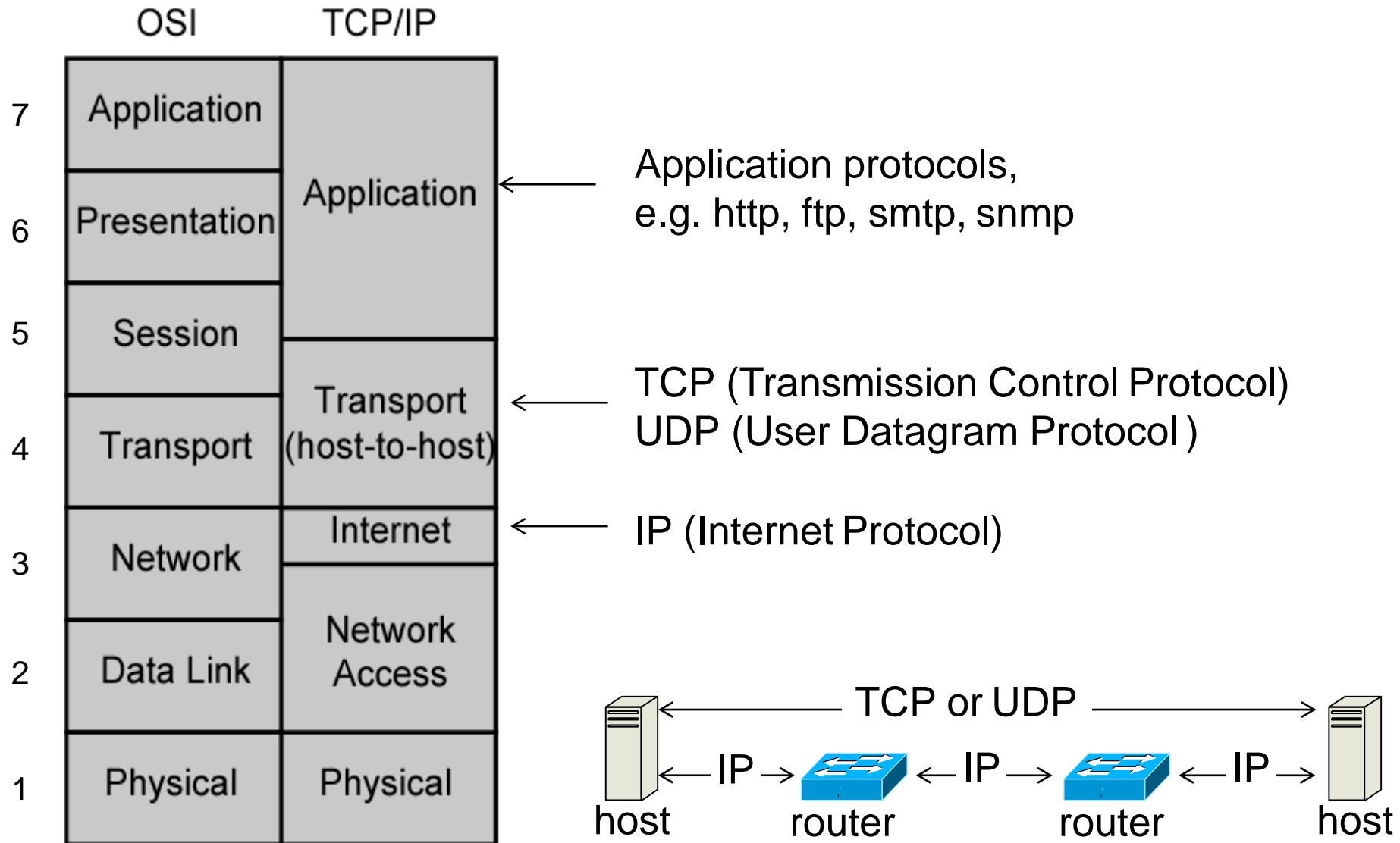| Layer 7	Application
Layer 6	**Presentation**
Layer 5	**Session**
Layer 4	Transport
Layer 3	Network
Layer 2	Data link
Layer 1	Physical

Communication across OSI

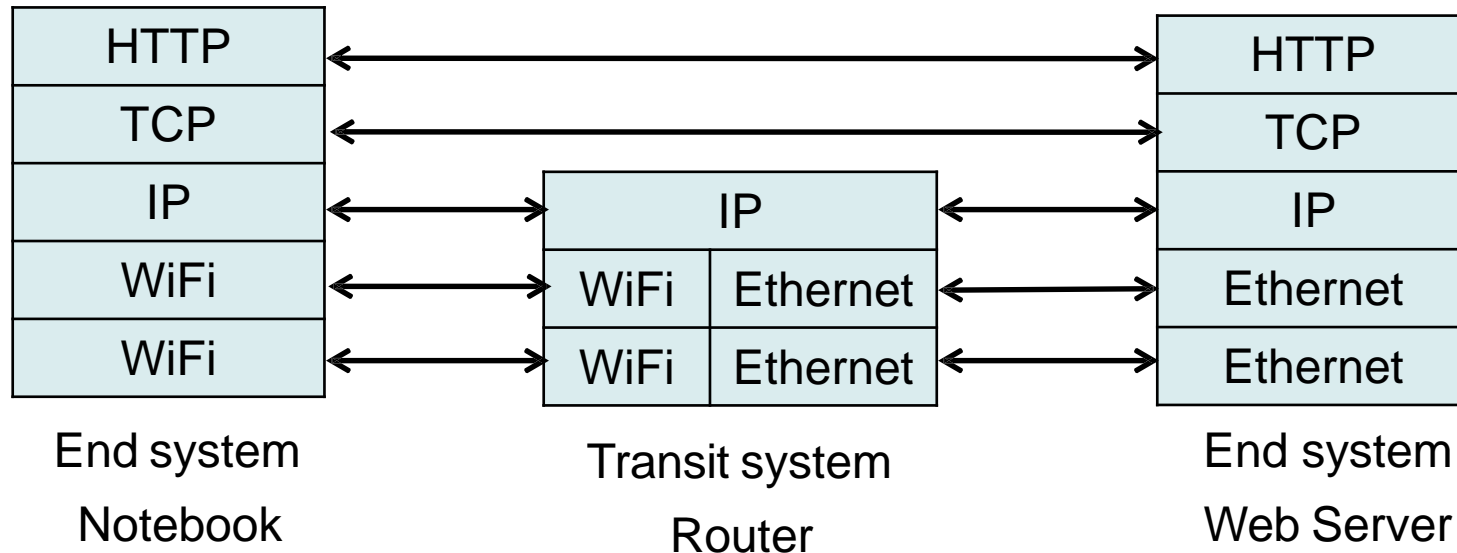# Communication across OSI

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 it's a working one.
  - Application layer
  - Host to host or transport layer
  - Internet layer
  - Network access layer
  - Physical layer

OSI model vs. TCP/IP model (The Internet)

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

Application protocols,
e.g. http, ftp, smtp, snmp

TCP (Transmission Control Protocol)
UDP (User Datagram Protocol)

IP (Internet Protocol)

TCP or UDP

host → IP → router ← IP → router ← IP → host

TCP/IP Model

- Example: Access over WiFi router

HTTP	← →	HTTP
TCP	← →	TCP
IP	← → IP ← →	IP
WiFi	← → WiFi | Ethernet ← →	Ethernet
WiFi	← → WiFi | Ethernet ← →	Ethernet

End system   Transit system   End system

Notebook   Router   Web Server

Communication Security Analogy

Physical transport security

Internet

Protected Pipe

Digital communication security
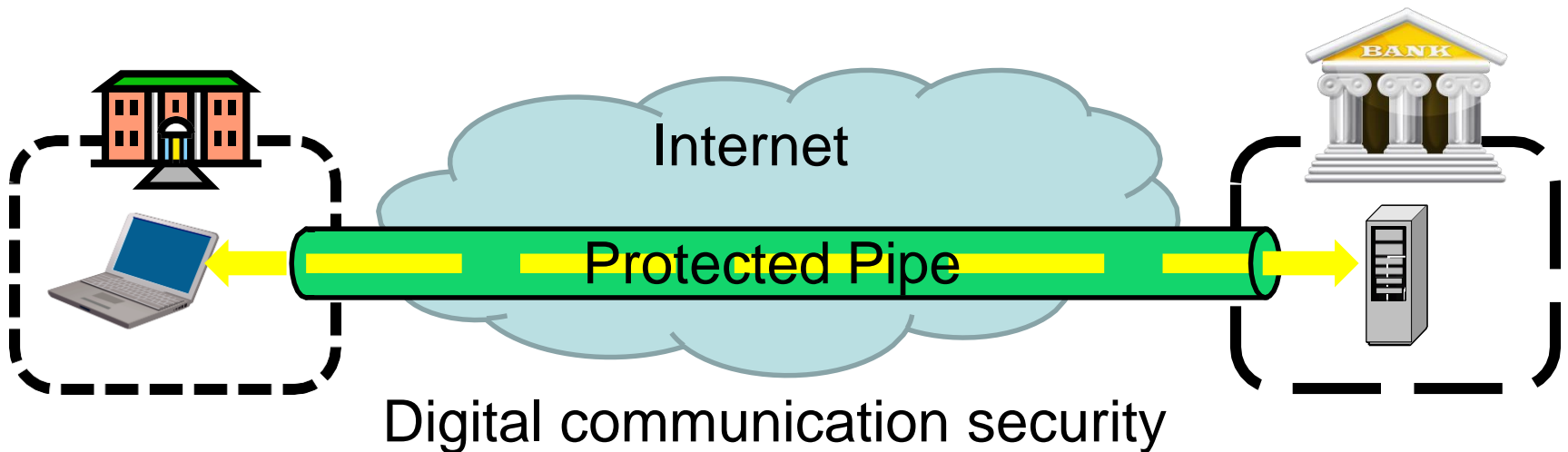
Security Protocols

- Many different 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 (e.g. for TLS: DROWN, POODLE, ROBOT, Logjam, FREAK, BEAST, …)
  - … some are never discovered (or maybe only by the attackers)

Security Protocols Overview

- This lecture discusses the operation of two network-related protocols that are in common use.
  - 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.

Transport Layer Security

TLS/SSL

SSL/TLS: History

- 1994: Netscape Communications developed the network authentication protocol Secure Sockets Layer, SSLv2.
  - Badly broken
- 1995: Netscape release their own improvements SSLv3.
  - Widely used for many years.
- 1996: SSLv3 was submitted to the IETF as an Internet draft, and an IETF working group was formed to develop a recommendation.
- In January 1999, RFC 2246 was issued by the IETF, Transport Layer Security Protocol: TLS 1.0
  - Similar to, but incompatible with SSLv3
  - Currently TLS 1.2 (2008) (allows backwards compatibility with SSL)
  - Draft TLS 1.3 (2016) (totally bans SSL)
  - Firefox browser enabled TLS 1.3 by default in February 2017[1]

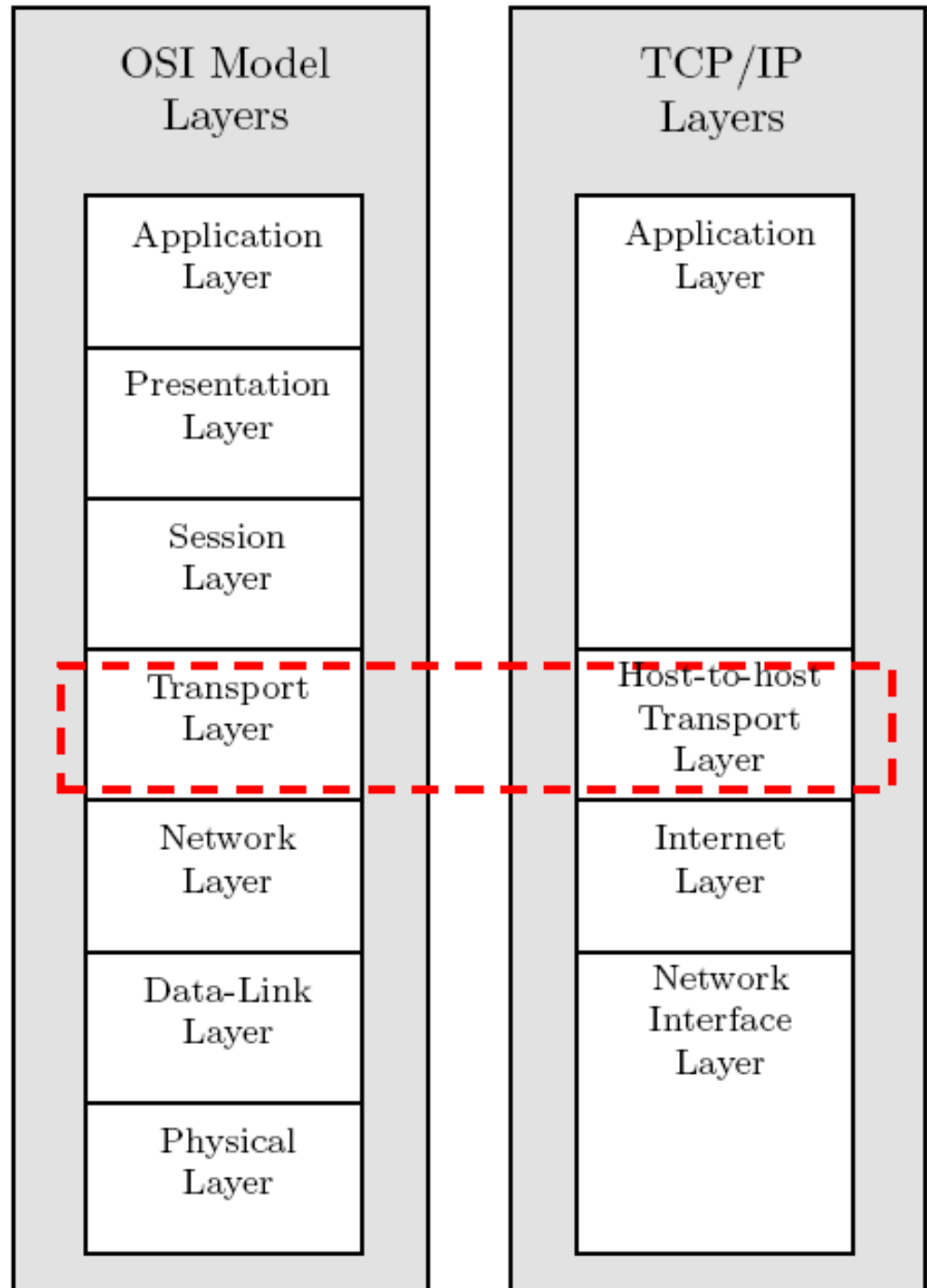# SSL/TLS Protocol versions
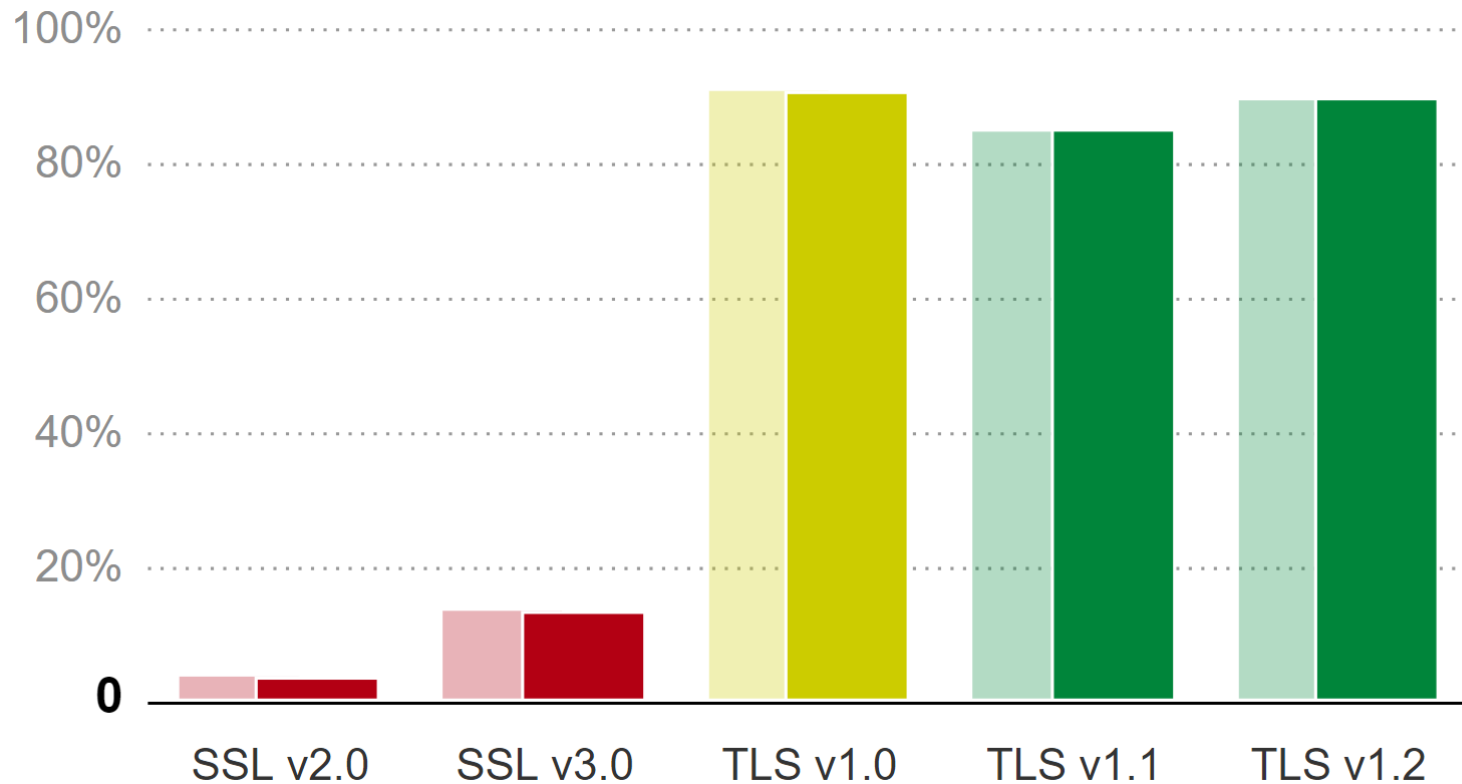
TLS: Overview

- TLS is a cryptographic services protocol based on the Browser PKI, and is commonly used on the Internet.
  - Each server has a server certificate and private key installed
  - Allows browsers to establish secure sessions with web servers.
- Port 443 is reserved for HTTP over TLS/SSL and the protocol https is used with this port.
  - http://www.xxx.com implies using standard HTTP using port 80.
  - https://www.xxx.com implies HTTP over TLS/SSL with port 443.
- Other applications:
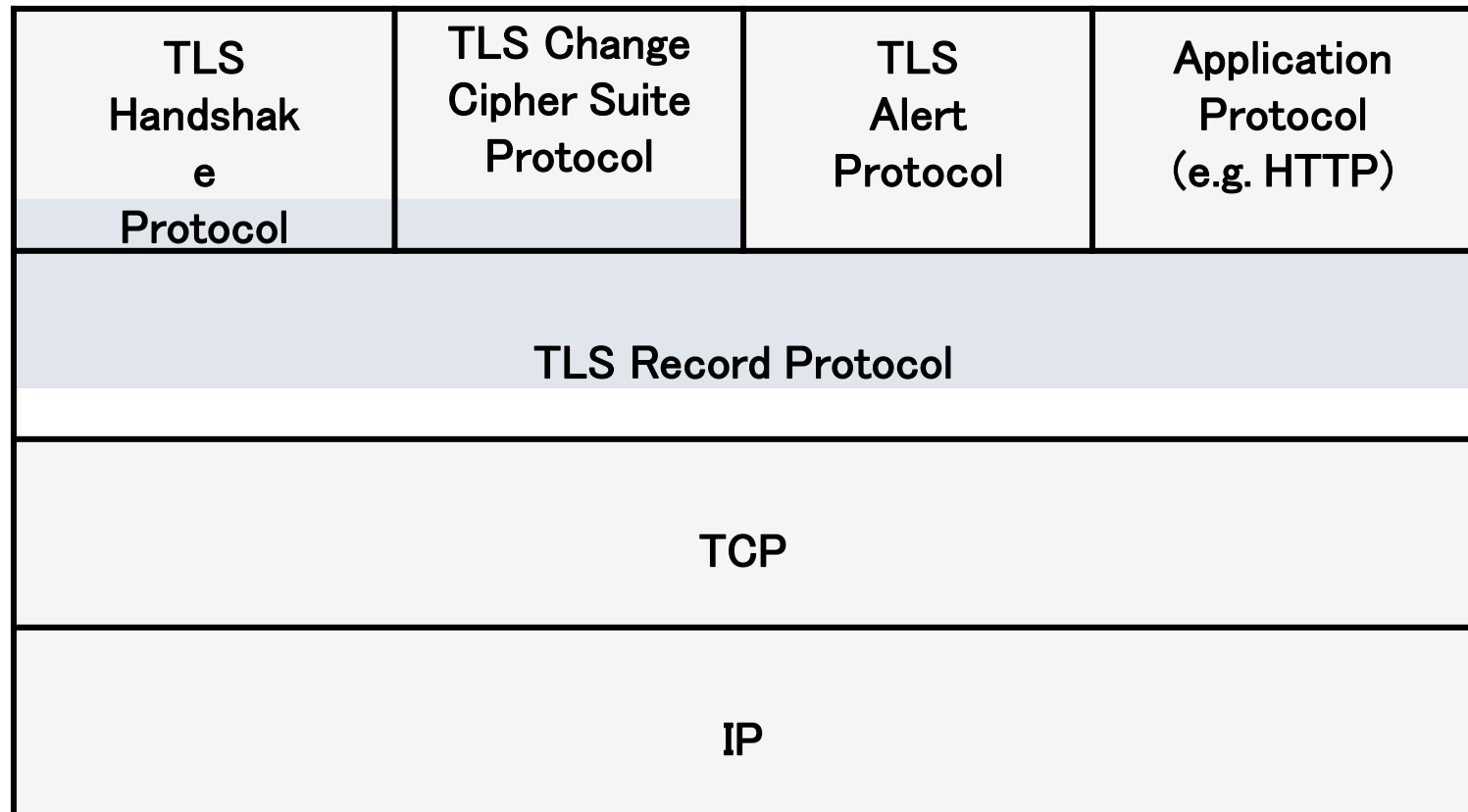 - IMAP over TLS: port 993
  - POP3 over TLS: port 995

TLS:
Layer 4 Security

OSI Model Layers / TCP/IP Layers

TLS operates at Layer 4 →

OSI Model Layers:
- Application Layer
- Presentation Layer
- Session Layer
- Transport Layer
- Network Layer
- Data-Link Layer
- Physical Layer

TCP/IP Layers:
- Application Layer
- Host-to-host Transport Layer
- Internet Layer
- Network Interface Layer

TLS:
Protocol Stack

TLS Handshake Protocol	TLS Change Cipher Suite Protocol	TLS Alert Protocol	Application Protocol (e.g. HTTP)
TLS Record Protocol			
TCP			
IP			

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 the practical encryption and integrity services to various application protocols.
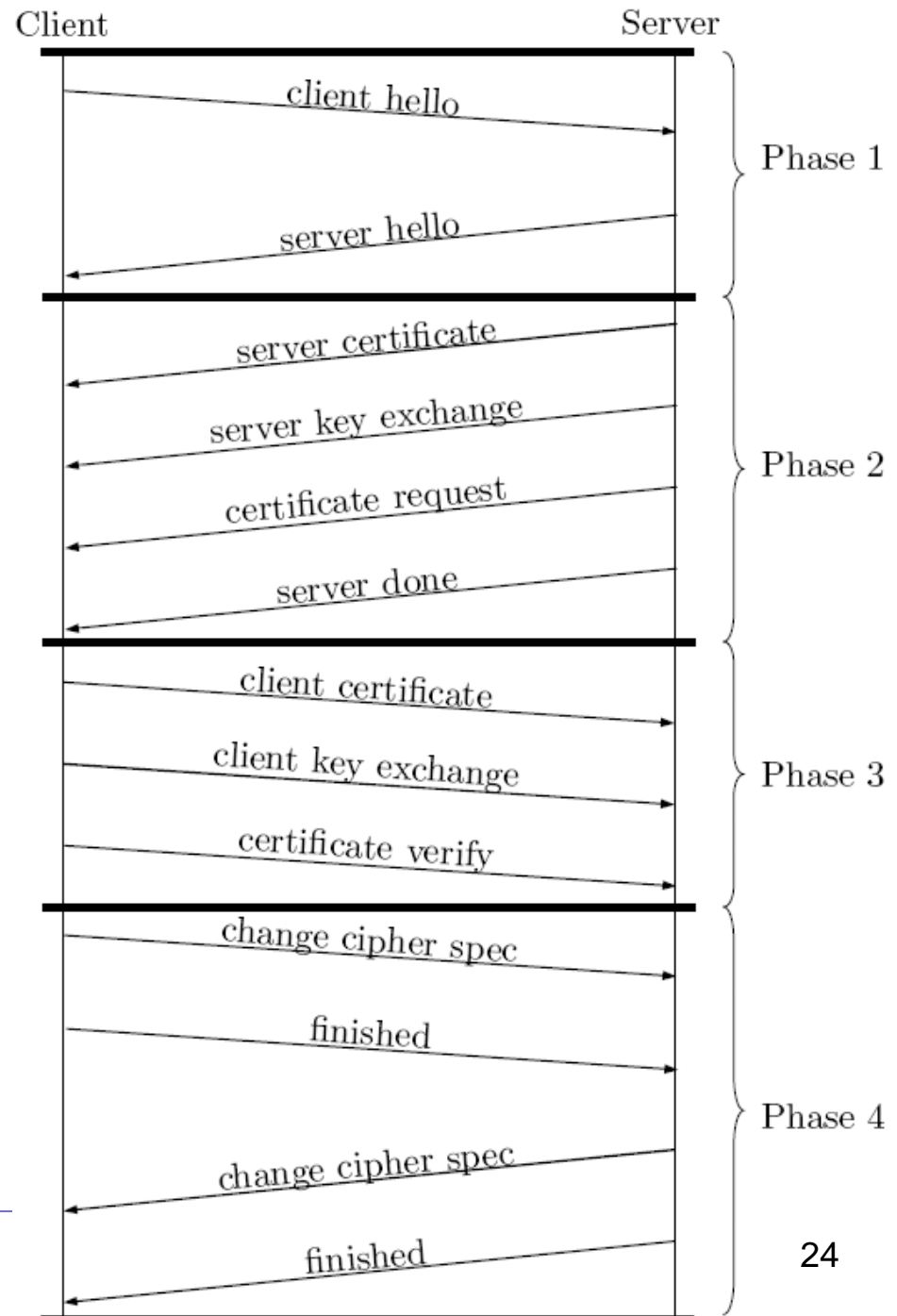
TLS: Handshake Protocol

- 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.

Client   Server

client hello
server hello
— Phase 1

server certificate
server key exchange
certificate request
server done
— Phase 2

client certificate
client key exchange
certificate verify
— Phase 3

change cipher spec
finished
change cipher spec
finished
— Phase 4

24

TLS: Simplified RSA-based Handshake

Client

Server

Supported crypto algorithms and protocol versions

Client Hello →

Server Hello ←

Common protocol, Common algorithm, Server certificate

Secret material encrypted with server pub. key

Client Key Exchange →

Client and Server generate session key from secret material

Go to crypto with common algorithm and session key

Change Cipher Suite →

Change Cipher Suite ←

Go to crypto with common algorithm and session key

Continues with TLS Record protocol encrypted with session key

TLS: Elements of Handshake

- Client hello
    - Advertises available algorithms (e.g. RSA, AES, SHA256)
    - Different types of algorithms bundled into "Cipher Suites"
    - Format:
      TLS_*key-exchange-algorithm*_WITH_*data-protection-algorithm*
    - Example: TLS_RSA_WITH_AES_256_CBC_SHA256
        - RSA for key exchange
        - AES with CBC mode for encryption
        - SHA256 as hash function for authentication and integrity protection

- Server hello
    - Returns the selected cipher suite
    - Server adapts to client capabilities

TLS: Elements of Handshake

- **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.

- **Client Certificate**
  - Optionally, the client can send its X.509 certificate to server, in order to provide mutual authentication

- **Server/Client Key Exchange**
  - The client and server can a establish session key using asymmetric encryption or DH key exchange (details below)

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 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:
  - Calculates a MAC over the compressed data using a MAC secret from the connection state.

- ## Encrypt:
  - Compressed data plus MAC are encrypted with 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: Key Exchange

- Two possibilities for exchange of secret key material (premaster secret, PS):
  - RSA encryption
  - DH exchange
- RSA encryption:
  - Client generates PS + encrypts PS with server public key (RSA)
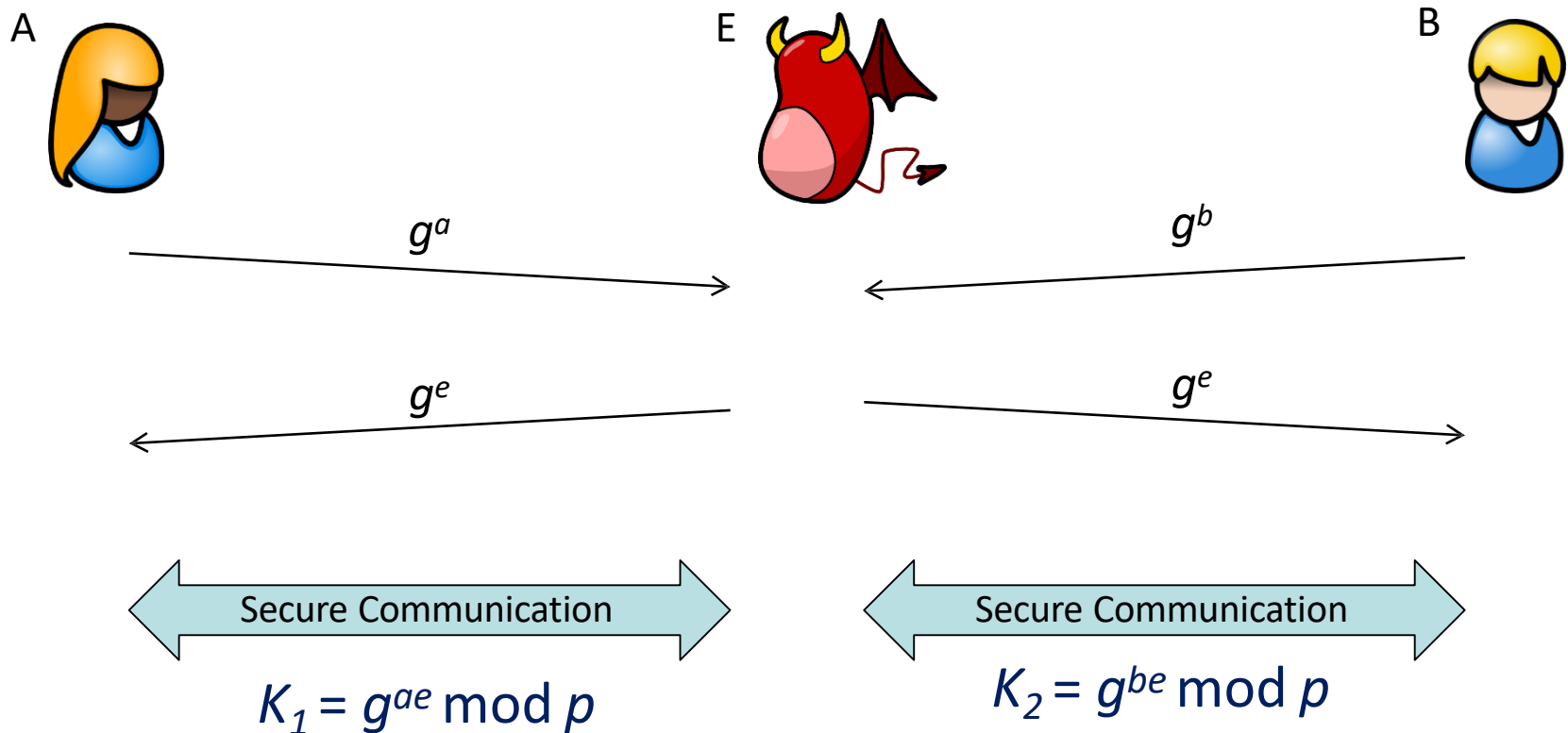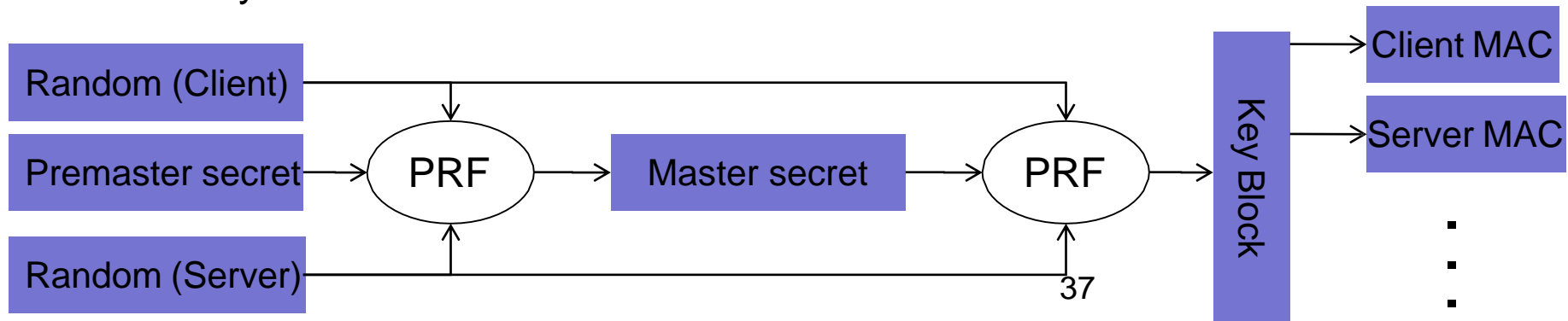 - Server decrypts PS with server private key (RSA)

Illustration of DH Key Exchange

Image source: Wikipedia
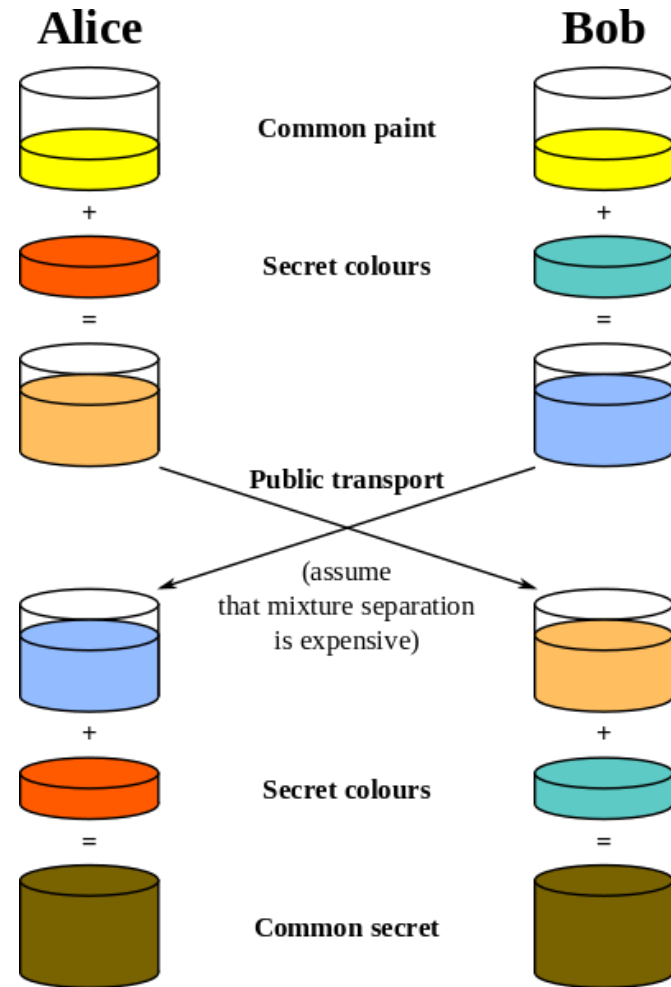
Diffie Hellman Key exchange

- Process:
  - Alice and Bob agree on (public parameters):
    - Large prime number p (all calculation are performed „mod p")
    - Generator g (i.e. g is primitive root mod p)
  - Alice chooses random number a ($1 < a < p$ - 1) and sends g^a to Bob
  - Bod chooses random number b ($1 < b < p$ - 1) and send g^b to Alice
  - Common secret: $K = (g^a)^b \bmod p = (g^b)^a \bmod p = g^{ab} \bmod p$

- Security:
  - K can not be calculated from g^a or g^b

Weakness of DH Key Exchange

A

E

B

g^a

g^b

g^e

g^e

Secure Communication

Secure Communication

$K_1 = g^{ae} \bmod p$

$K_2 = g^{be} \bmod p$

TLS: Key Exchange

- Two possibilities for exchange of secret key material (premaster secret, PS):
  - RSA encryption
  - DH exchange

- RSA encryption:
  - Client generates PS + encrypts PS with server public key (RSA)
  - Server decrypts PS with server private key (RSA)

- DH exchange:
  - Client and server perform Diffie-Hellman-Exchange (DH)
  - Server signs his DH value with his private key (RSA)
  - Client validates signature with server public key (RSA)

TLS Key Exchange

- Problem with RSA key exchange?
- Lets assume adversary records complete TLS session
- If later private key of server is known
  - Premaster secret can be decrypted
  - Session key can be calculated
  - Complete payload can be decrypted
- With DH exchange:
  - later knowledge of private key is useless
  - Payload remains protected
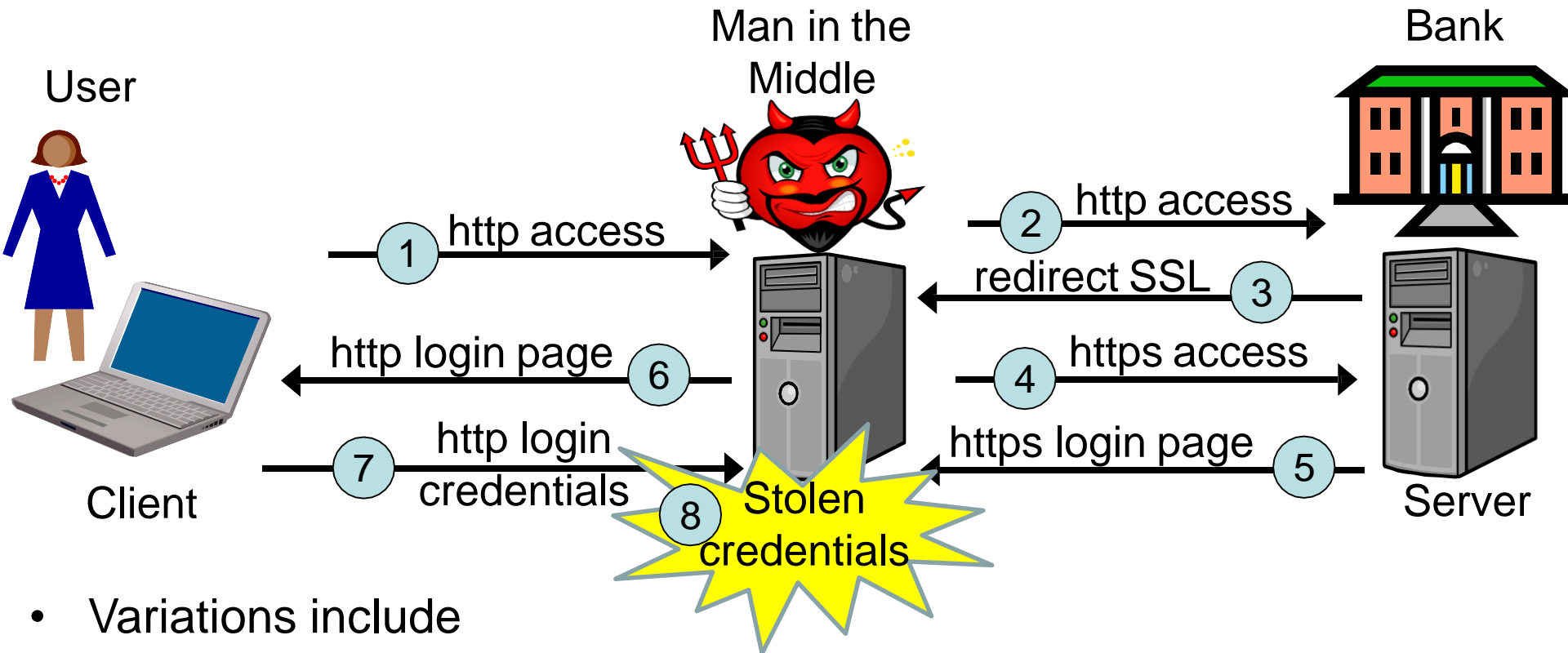 - "perfect forward secrecy"

TLS: Symmetric key derivation

- Using two random numbers (from client and server) + premaster secret

- Key material calculation (general)
  - Uses "Key Expansion"
  - Internally using a pseudo random function (based on hash function)
  - Can produce arbitrary length key material

- Master secret calculation
  - Input: Premaster Secret, random number client, random number server
  - Output: Master Secret (48 byte)

- Encryption/MAC key calculation
  - Input: Master Secret, random number client, random number server
  - Output: Key block, is partitioned into required keys

Random (Client) → PRF → Master secret → PRF → Key Block → Client MAC
Premaster secret → PRF
Random (Server) → PRF → Server MAC

37

SSL/TLS Challenges

- Higher layers should not be overly reliant on SSL/TLS.

- Many vulnerabilities exist for SSL/TLS.
  - People are easily tricked
  - Changing between http and https causes vulnerability to SSL stripping attacks
  - SSL/TLS only as secure as the cryptographic algorithms used in handshake protocol: hashing, symmetric and asymmetric crypto.

- Relies on Browser PKI which has many security issues
  - Fake server certificates difficult to detect
  - Fake root server certificates can be embedded in platform, see e.g. Lenovo Komodia advare scam

SSL Stripping Attack

User

Man in the Middle

Bank

1 http access

2 http access

redirect SSL 3

http login page 6

4 https access

http login credentials 7

https login page 5

8 Stolen credentials

Client

Server

- Variations include
  - MitM server can connect to client over https in msg (6) with server certificate that has similar domain name as real server.
  - Attacker can leave the connection after stealing credentials, then the client connects directly to real server with https
  - Attacker just downgrades the https connnection to a vulnerable SSL/TLS version or a broken cipher suite

HSTS – HTTP Strict Transport Security
Preventing SSL Stripping

- A secure server can instruct browsers to only use https

- When requesting website that uses HSTS, the browser automatically forces connect with https.

- Users are not able to override policy

- Two ways of specifying HSTS websites
  - List of HSTS websites can be preloaded into browsers
  - HSTS policy initially specified over a https connection
    - ➢ HSTS policy can be changed over a https connection

- Disadvantages
  - HSTS websites can not use both http and https
  - Difficult for a website to stop using https
  - Can cause denial of service, e.g. no fallback to http in case of expired server certificate

Preventing SSL Stripping with HSTS
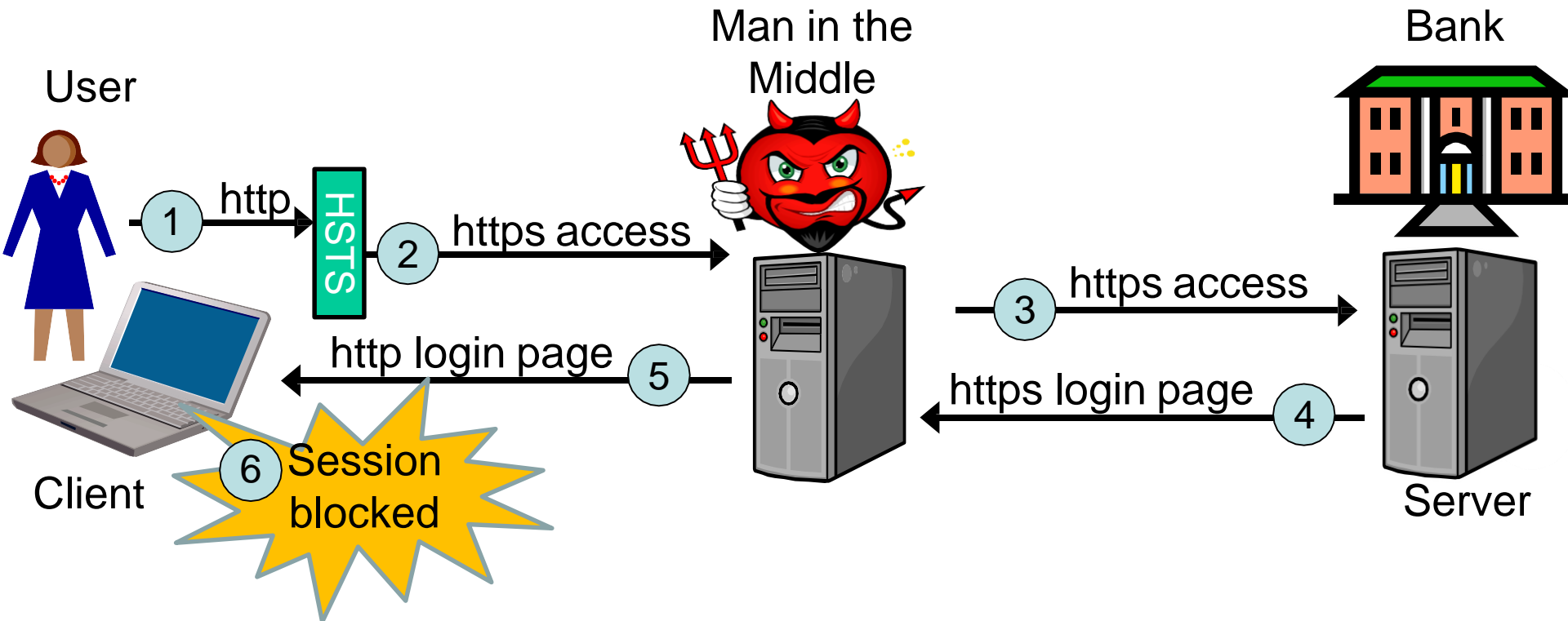
- Limitation of HSTS:
  - Requires first visit to secure website to set HSTS policy in browser
- Can be solved by browser having preloaded list of HSTS websites
- Browsers would be vulnerable if attacker could delete HSTS cache

Phishing and failed authentication
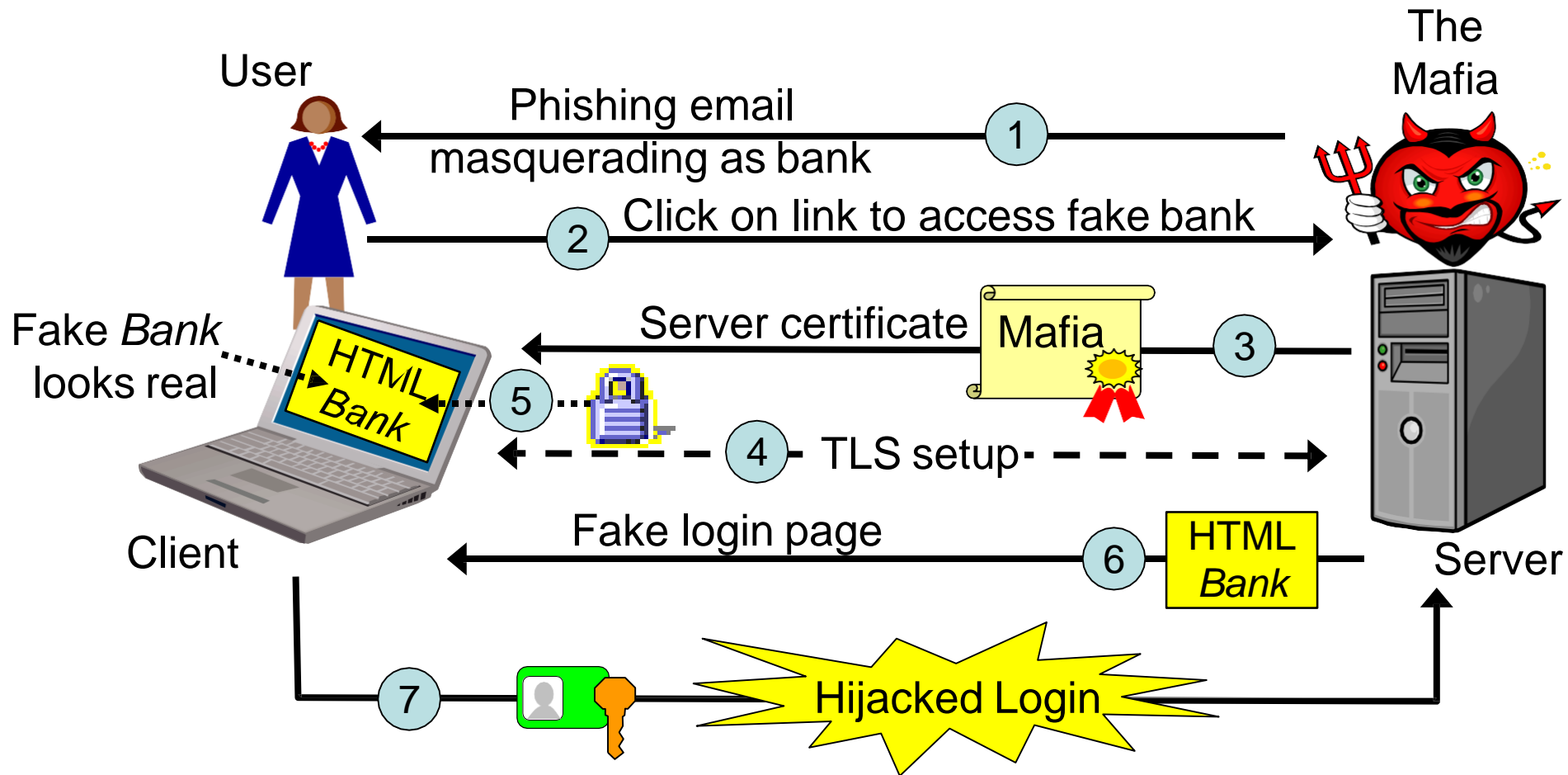
IP Layer Security

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

- Provides a security architecture for both IPv4 and IPv6
  - Mandatory for IPv6
  - Optional for IPv4

- Requires operating system support, not application support.

Layer 3 Security

OSI Model Layers	TCP/IP Protocol Architecture Layers
Application Layer	Application Layer
Presentation Layer	
Session Layer	
Transport Layer	Host-to-Host Transport Layer
Network Layer	Internet Layer
Data-Link Layer	Network Interface Layer
Physical Layer	

IP Sec Operation →

IPSec:
Security Services

- ## Message Confidentiality.
  - Protects against unauthorized data disclosure.
  - Accomplished by the use of encryption mechanisms.

- ## 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.

- ## 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.

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
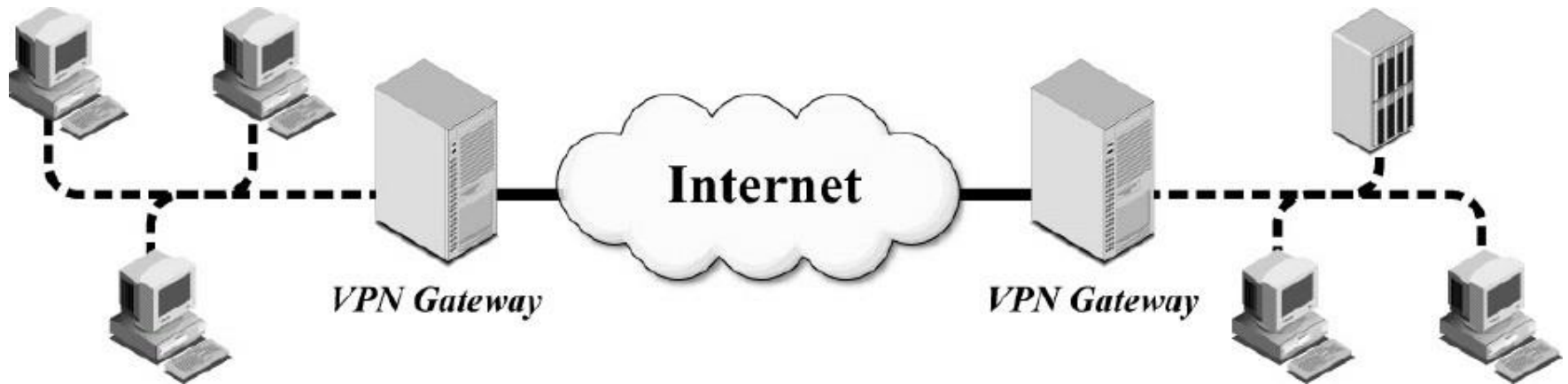
Source: NIST Special Publication 800-77

IPSec:
Host-to-Gateway Architecture
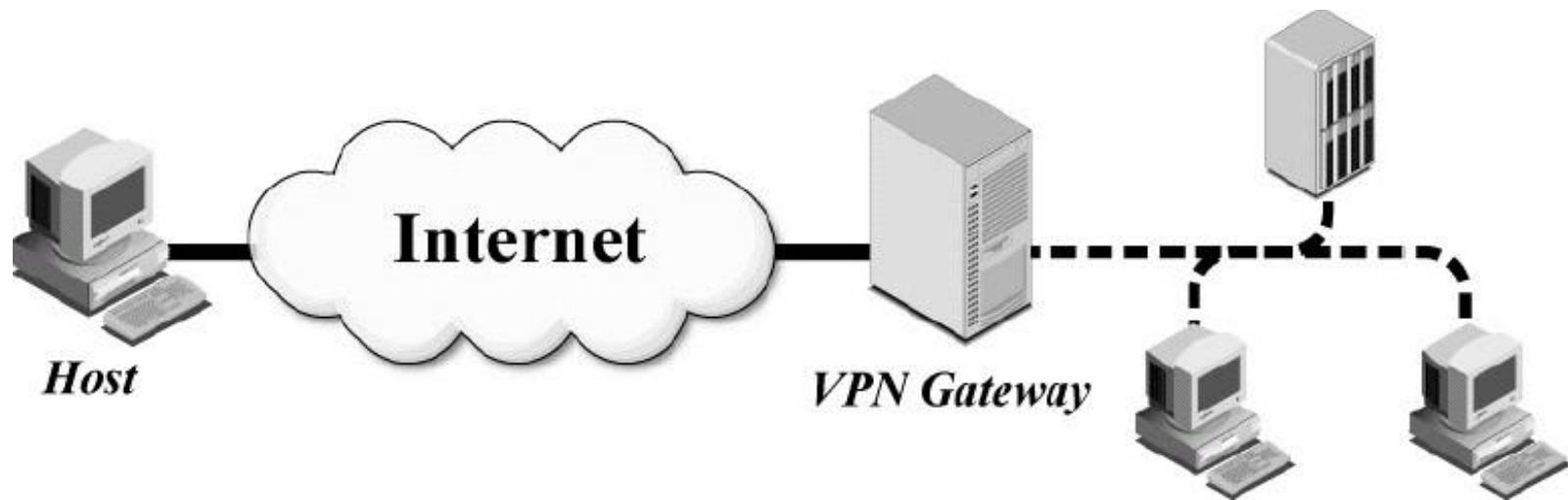
Source: NIST Special Publication 800-77

IPSec:
Host-to-Host Architecture
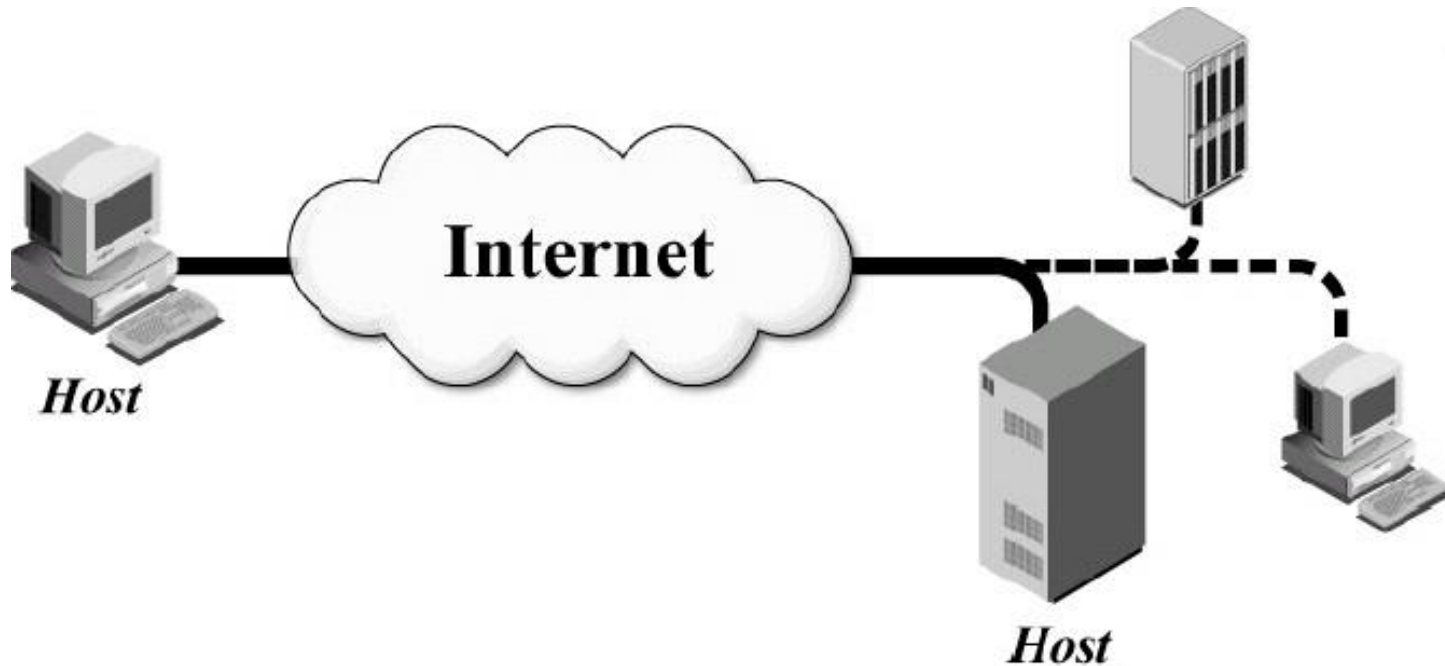
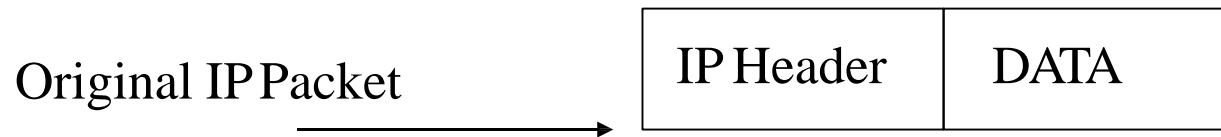Source: NIST Special Publication 800-77

IPSec:
Protocols Types

- Encapsulating Security Payload (ESP)
  - Confidentiality, authentication, integrity and replay protection

- Authentication Header (AH)
  - Authentication, integrity and replay protection. However there is <u>no confidentiality</u>

- Internet Key Exchange (IKE)
  - negotiate, create, and manage security associations

- A connection consists of two SA (Security Associations)
  - One SA for each directions
  - Each SA is described by a set of parameters

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.
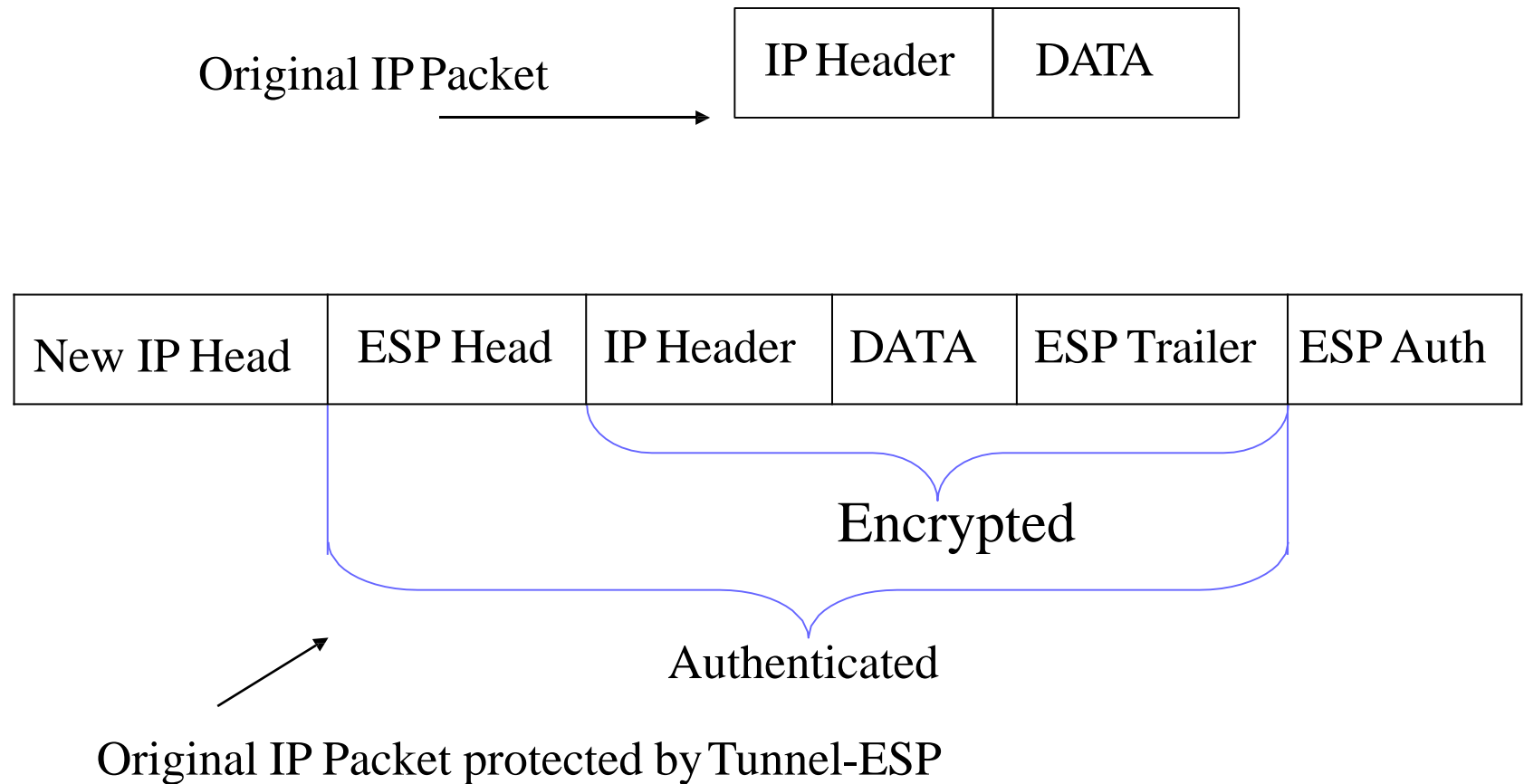 - Typical use is gateway-to-gateway and host-to-gateway architectures.

Transport Mode ESP

Original IP Packet →

IP Header	DATA

IP Header	ESP Header	DATA	ESP Trailer	ESP Auth

Encrypted

Authenticated
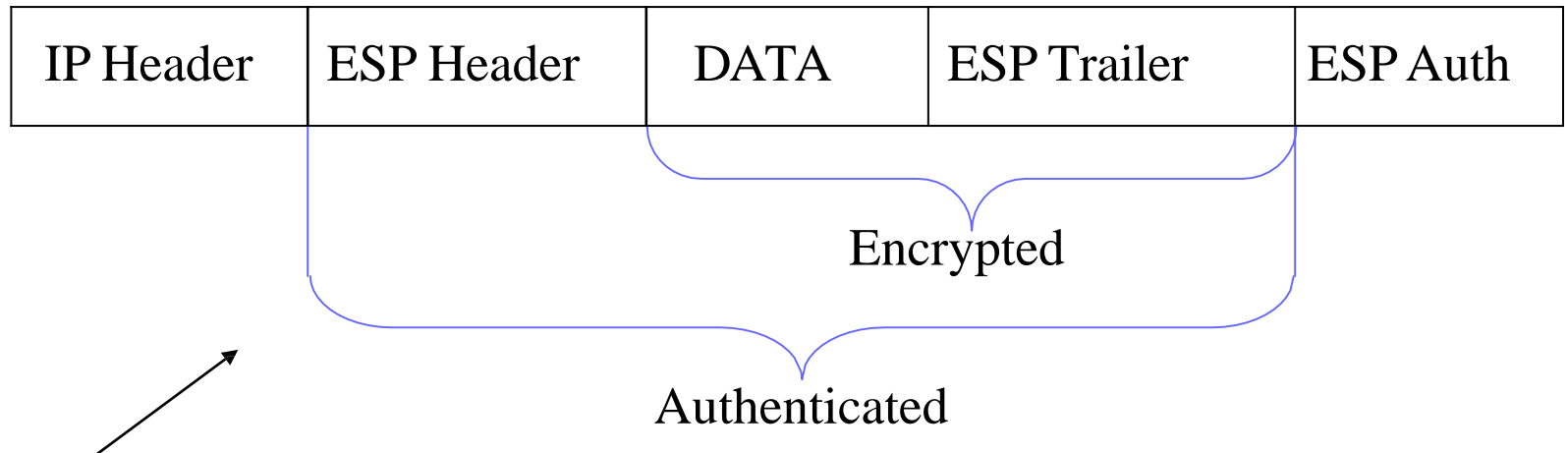
Original IP Packet protected by Transport-ESP

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

Original IP Packet →

IP Header	DATA

New IP Head	ESP Head	IP Header	DATA	ESP Trailer	ESP Auth

Encrypted

Authenticated

Original IP Packet protected by Tunnel-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.

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.

- Security Associations are simplex
  - need one for each direction of connection
  - stored in a security association database (SAD).

- Key exchange is largely automated after initial manual configuration by administrator prior to connection setup.

- (See ISAKMP, IKE, Oakley, Skeme and SAs)

Key Exchange

- Alice and Bob have common (long term) secret s
- DH exchange is **authenticated** (MITM not possible)
- After each session, session key is destroyed
- → **Perfect forward secrecy**

A   B

g^a

g^b

hash $(g^a \mathbin{||} g^b \mathbin{||} s)$

hash $(g^a \mathbin{||} g^b \mathbin{||} s \mathbin{||} \text{“Bob”})$
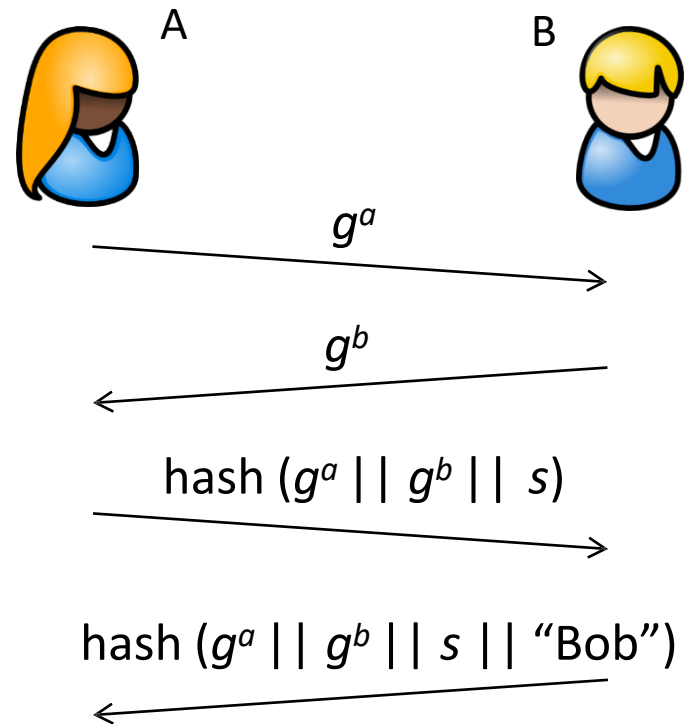
IPSEC - INTERNET PROTOCOL SECURITY

UNDERSTANDING VPN IPSEC TUNNEL MODE AND IPSEC TRANSPORT MODE - WHAT'S THE DIFFERENCE?

Typical usage of IPSec: VPN
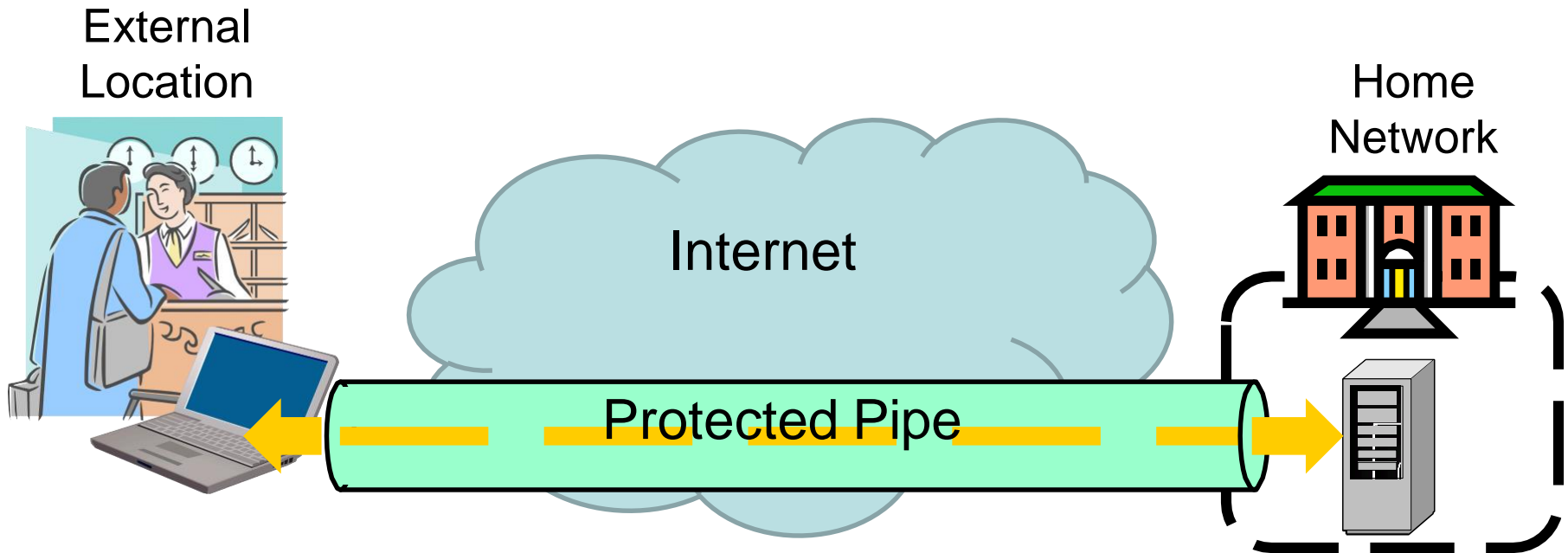
Risks of using IPSec for VPN

- IPSec typically used for VPN (Virtual Private Networks)
- A VPN client at external location may be connected to the Internet (e.g. from hotel room or café) while at the same time being connected to home network via VPN.
  - VPN gives direct access to resources in home network.
- Internet access from external location may give high exposure to cyber threats
  - No network firewall, no network IDS
- Attacks against the VPN client at external location can directly access the home network through VPN tunnel

Risk of using VPN
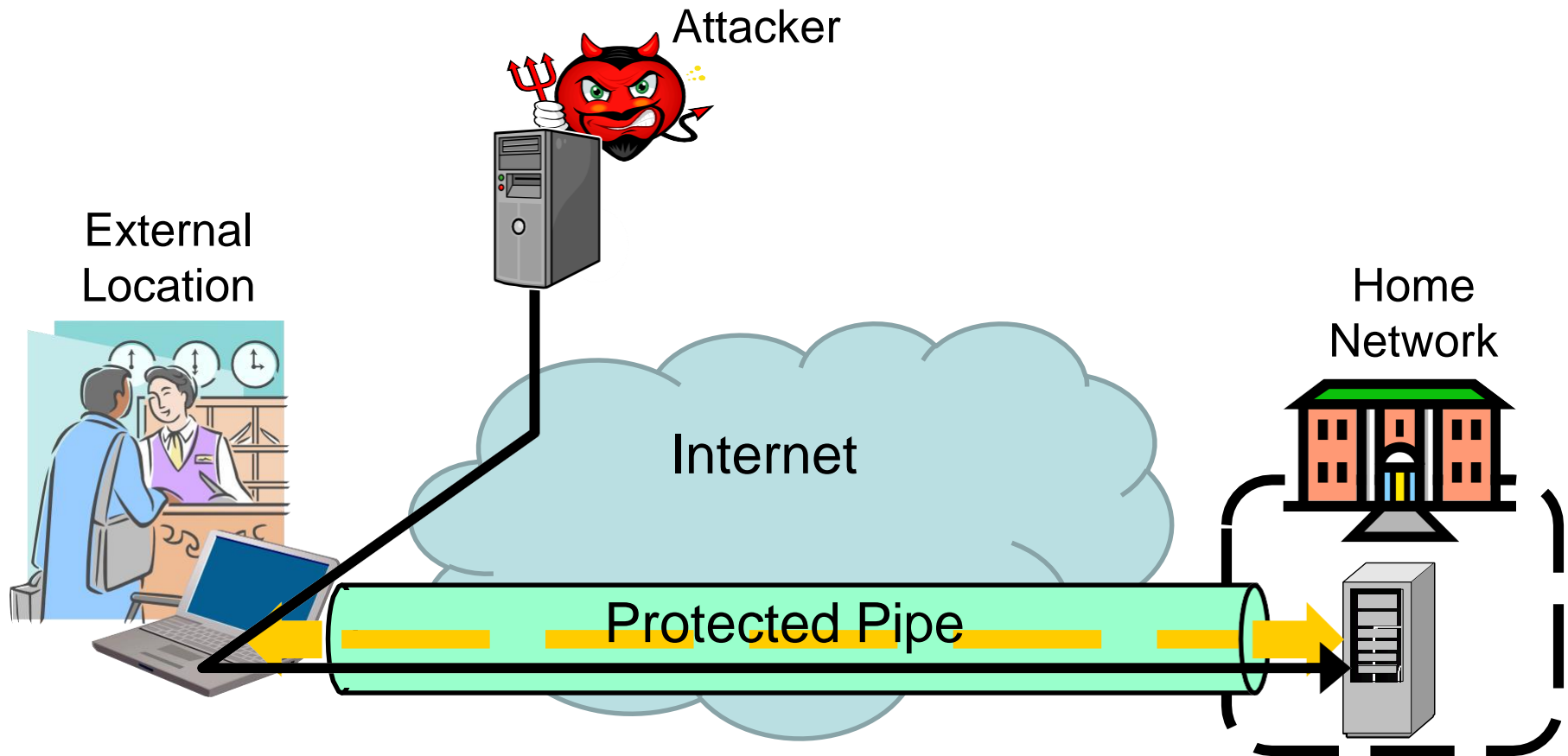
Secure pipe can be attack channel to home network !

Cloud VPN

VPNaaS

- A cloud-based infrastructure for VPN.

- A.k.a.:   - Hosted VPN

  - VPNaaS (Virtual Private Network as a Service)

- Cloud VPNs provide security and globally accessible VPN service access without the need for any VPN infrastructure on the user's end.

- The user connects to the cloud VPN through the provider's website or a desktop/mobile app.

- The pricing of cloud VPN is based on pay-per-usage or a flat-fee subscription.

- Disadvantages /risks
  - Cleartext-gap at the VPN provider
  - VPN provider knows Internet usage profile
  - Malicious VPN service?

Cloud VPN
Hosted VPN
VPNaaS

Company Branch A

Company Branch B

Internet

Protected Pipe
TLS or IPSec

Protected Pipe
TLS or IPSec

VPNaaS

(cleartext-gap)
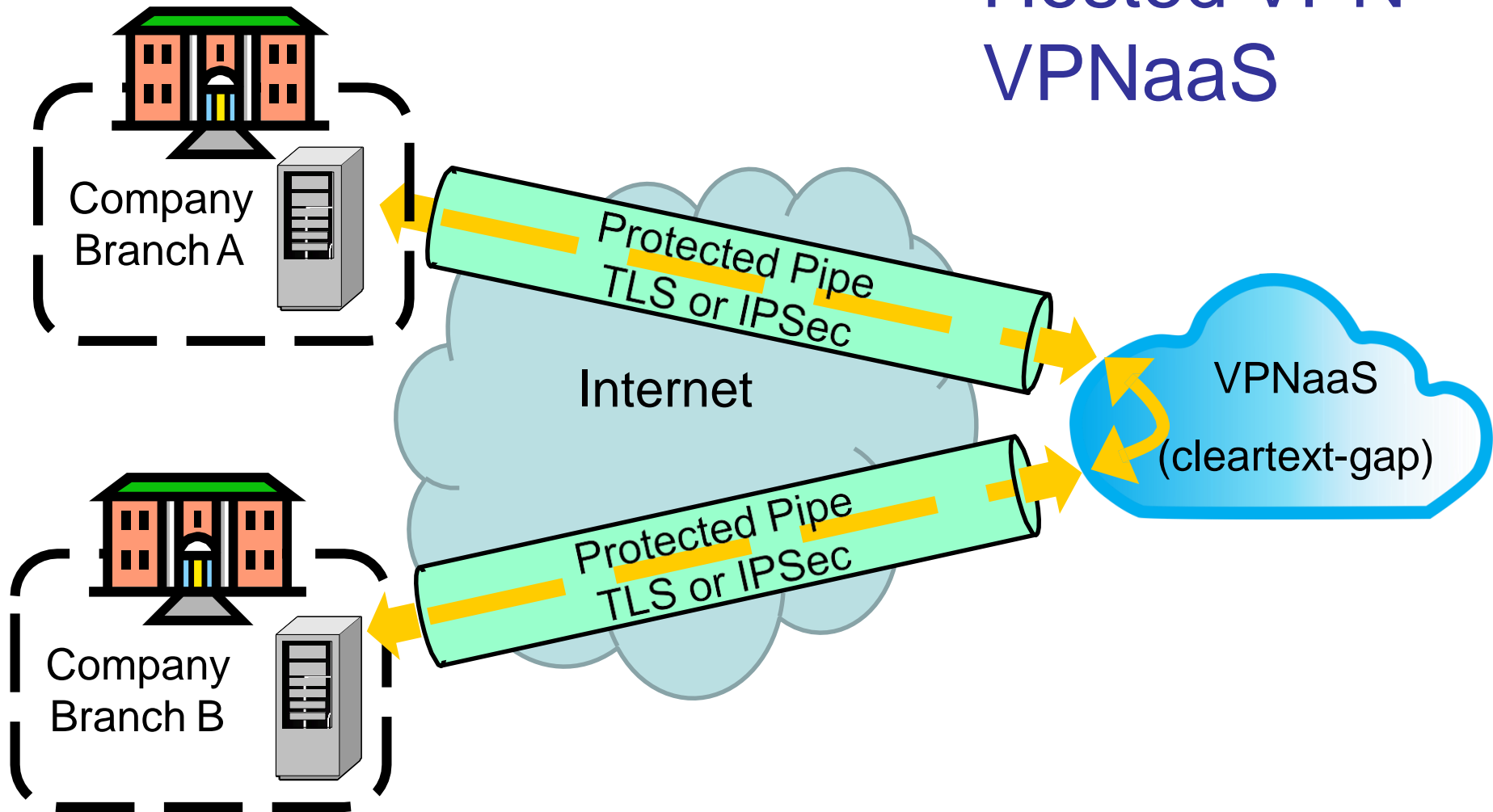
VPN Browsing – via VPN Proxy

Internet services

Internet

Optionally protected by end-to-end TLS

VPNaaS

(exposed URLs)

Protected Pipe

TLS or IPSec

User
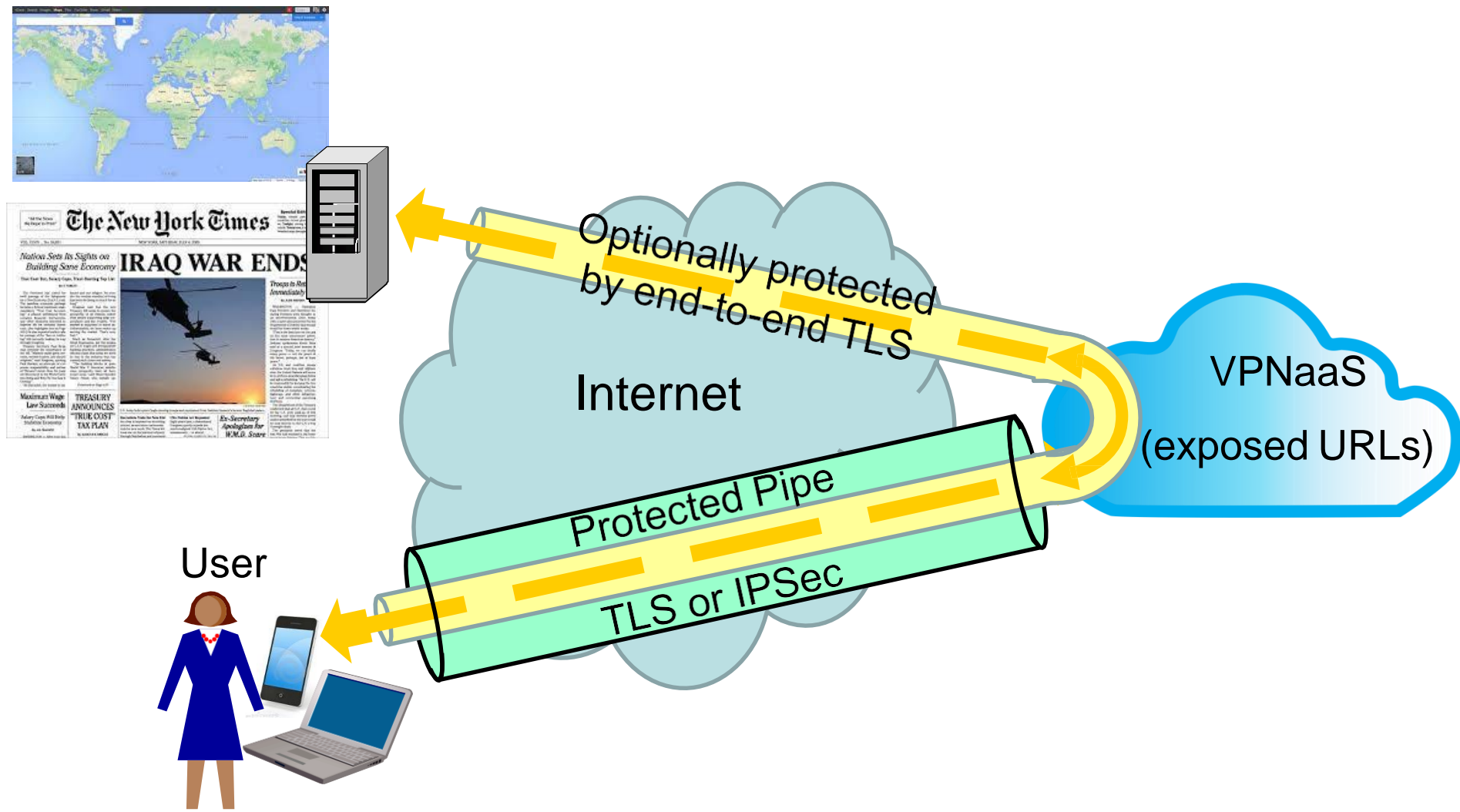
Tor – The Onion Router
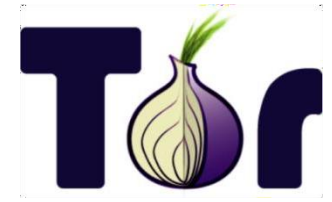
- An anonymizing routing protocol
- Originally sponsored by the US Naval Research Laboratory
- From 2004 to 2006 was supported by EFF
- Since 2006 independent nonprofit organisation

- Creates a multi-hop proxy circuit through the Internet from client to destination.
- Each hop "wraps" another encryption layer thereby hiding the next destination.
- No cleartext-gap, except at the exit-node.
- No node knows end-to-end client-server association
- Full technical details: https://www.torproject.org/
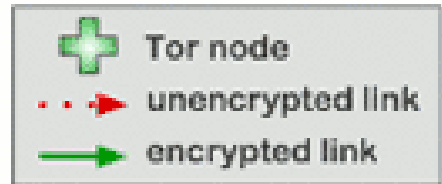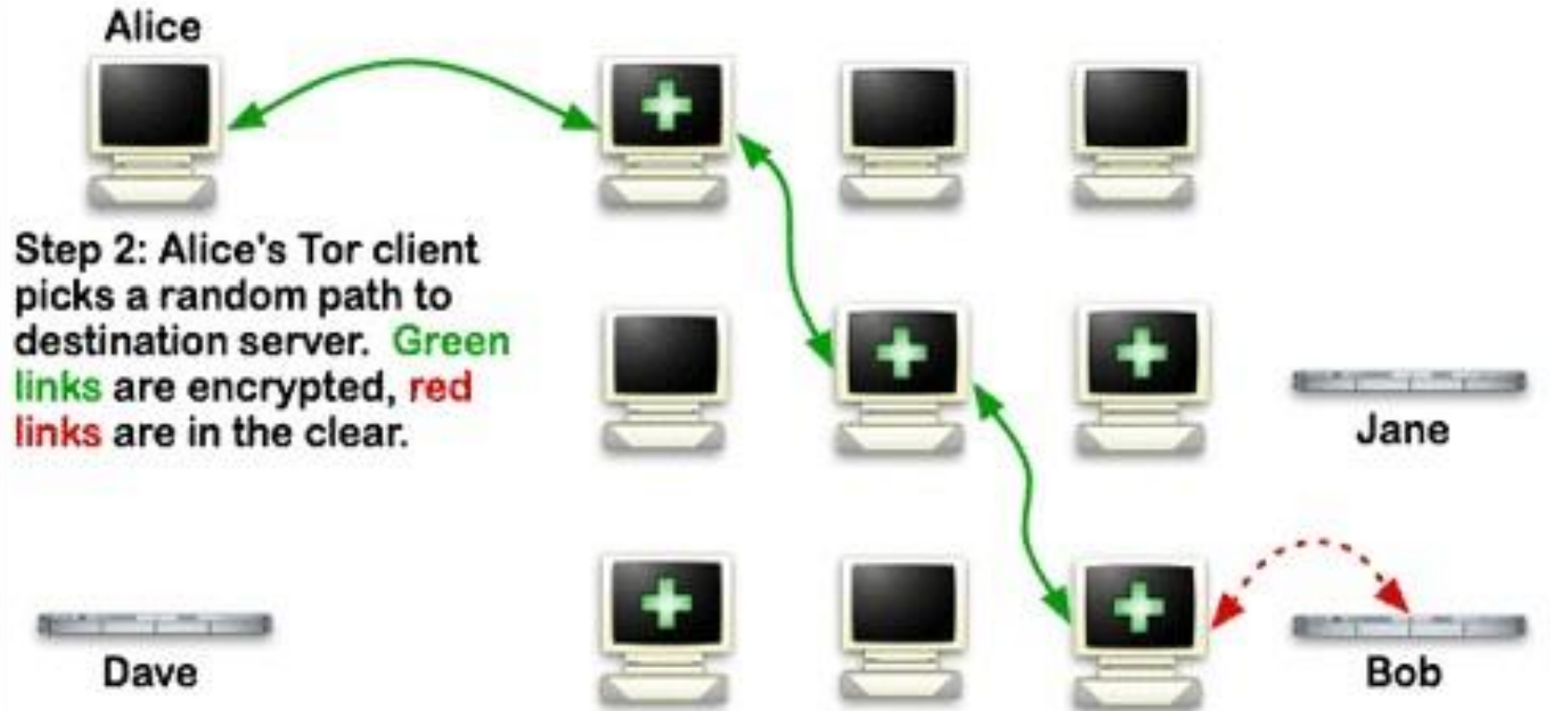
Image courtesy torproject.org

Image courtesy torproject.org

Image courtesy torproject.org

72

„Onion" Message

Destination: Router A
Encrypt for A

Destination: Router B
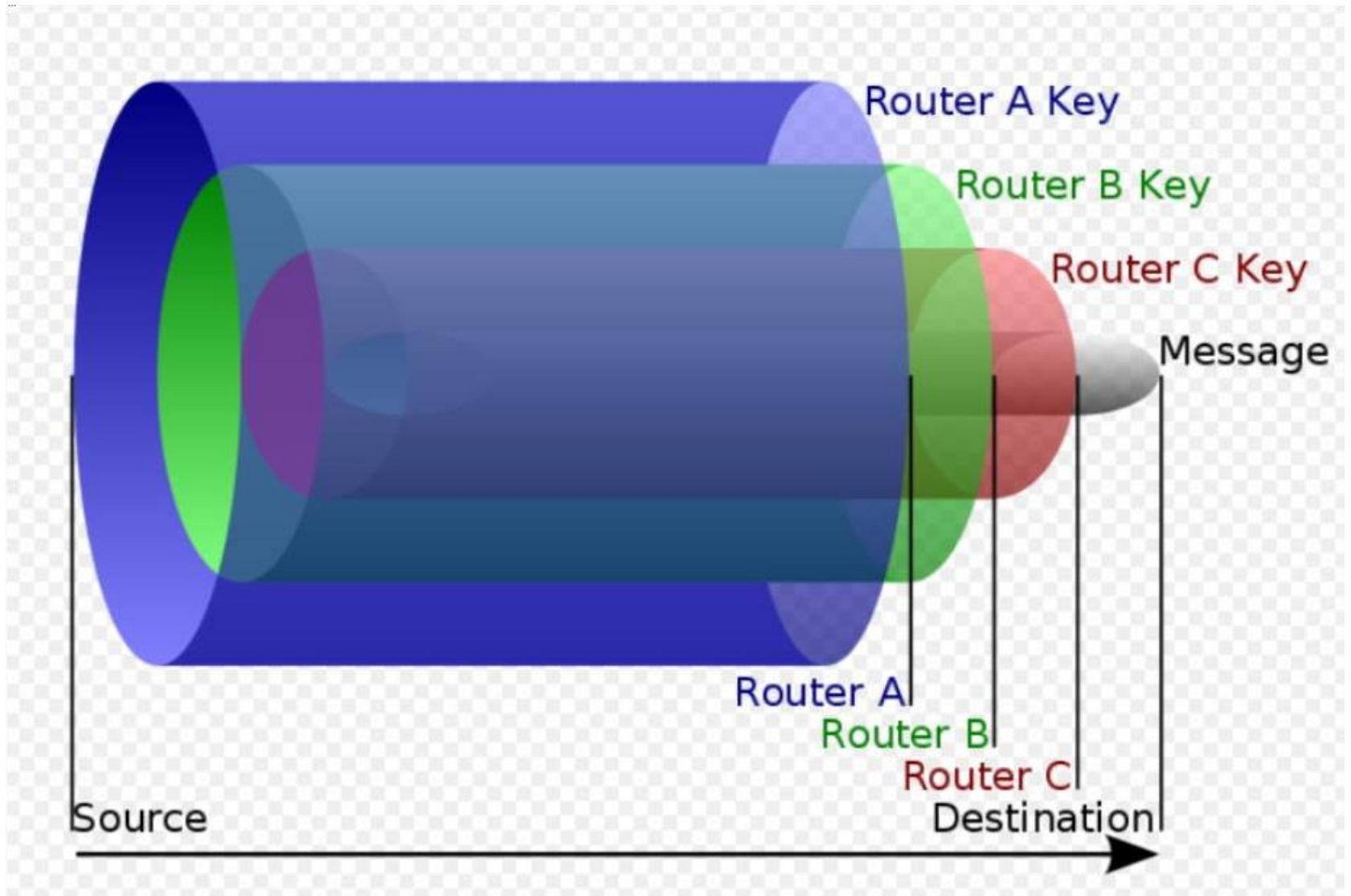Encrypt for B

Destination: Router C
Encrypt for C

Destination: Jane
Payload
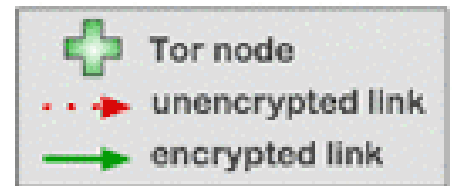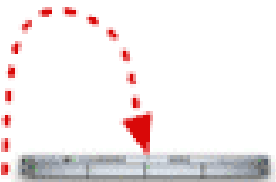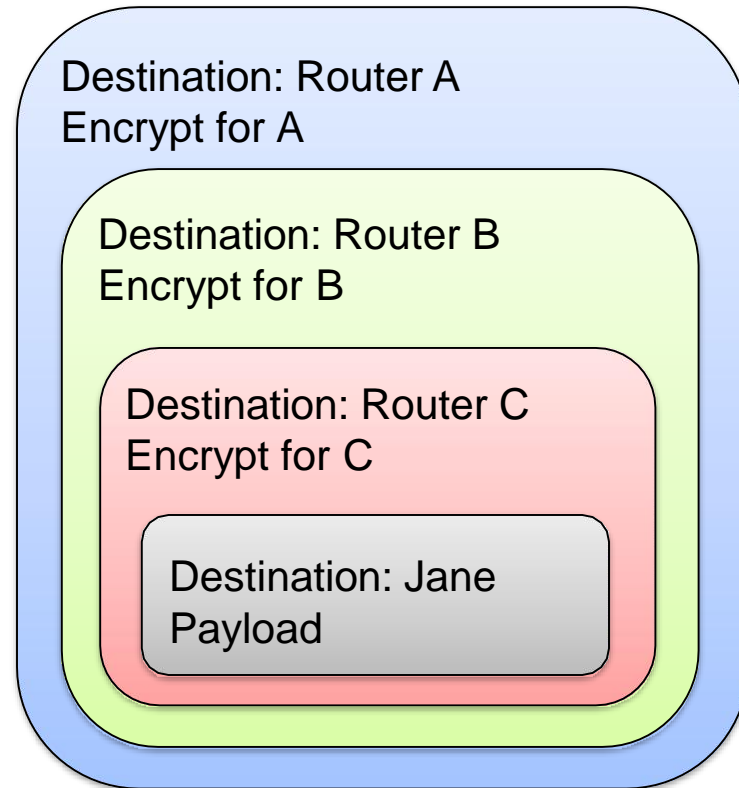
Diagram courtesy Wikimedia Commons

End of lecture