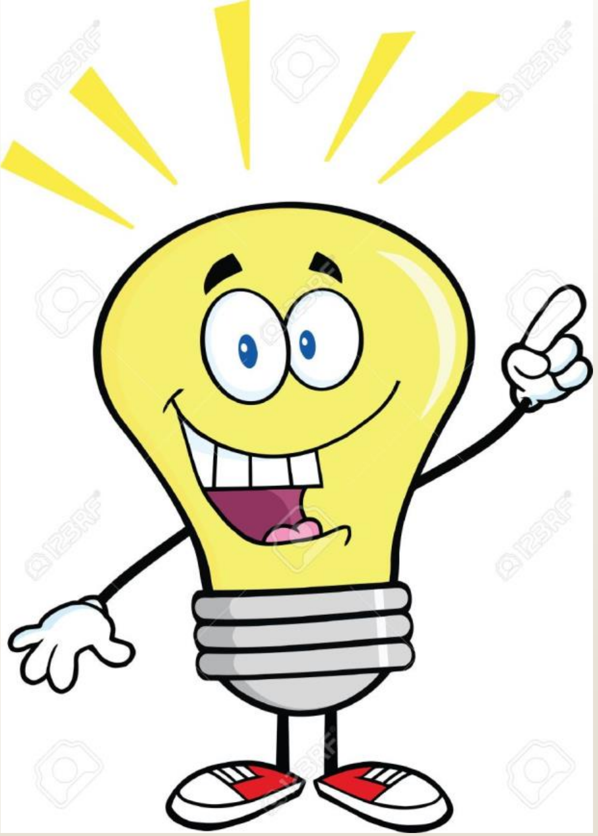
Network Security Foundations

Agenda

1. The OSI Model and TCP/IP Model
   - Why do we have various devices and they can still communicate with each other?
   - Application Layer
   - Transport Layer
   - Internet Layer
   - Network Access Layer
   - Data Encapsulation: Putting it all together
2. Cryptograhy Foundations
   - Why do we need cryptography?
   - Types of Cryptosystems
   - Symmetric Key Encryption
   - Asymmetric Key Encryption
3. HTTPS, SSL and TLS – Why and how do they work?
4. Computer Network Communications Example
   - How does it all work in real life?

The OSI Model and TCP/IP Stack

- BRAINSTORM: Why do we have various devices and they can still communicate with each other?



ANSWER:
They all implement the same networking model.

- <u>What is a networking model?</u>
- It’s a comprehensive set of documents, where individually, each document describes one small function required for a network.
- Think of a networking model just like you think of an architectural pattern for building a house.

- <u>Do I really need it?</u>
- Of course, you can build a house without the blueprint, but it can ensure the house has the right foundation and structure so it won’t fall down.
- The equivalent situation applies to computer networks.

Basic Concepts

- Protocol: A set of logical rules that devices must follow to communicate.
- Layer: Each model breaks the functions into a smaller number of categories, called layers. Each layer includes protocols and standards that relate to that category of functions.
- Host: Any device that has an IP address and connects to any TCP/IP network.

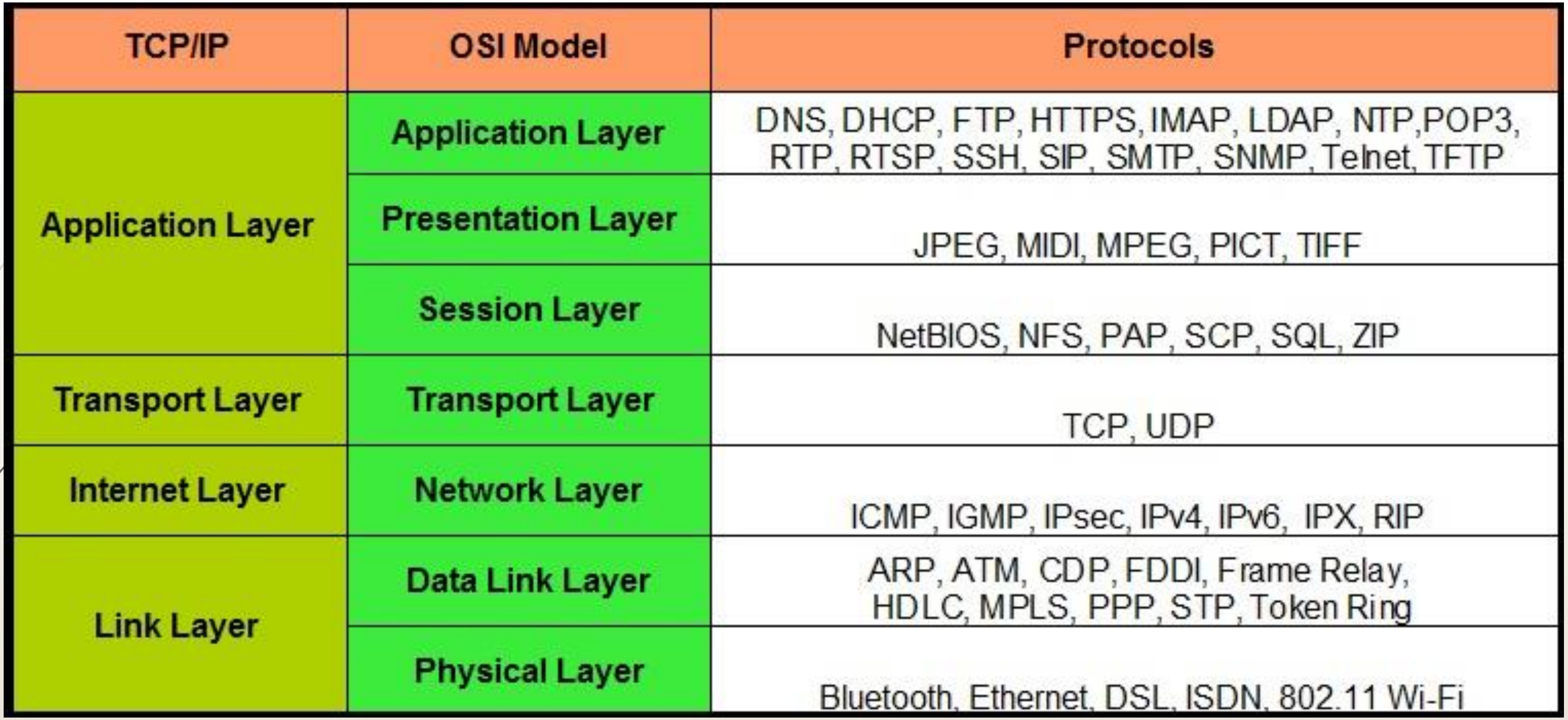
Two well-known networking models are:

✓ The OSI Model

✓ TCP/IP Model

The OSI Model & TCP/IP Model Comparison

<table>
<tr><th>OSI Model</th><th>TCP/IP Model</th></tr>
<tr><td colspan="2">Both define and reference a large collection of protocols that allow computers to communicate.</td></tr>
<tr><td></td><td>Protocols are defined in documents called Requests for Comments (RFC)</td></tr>
<tr><td>7 Layers</td><td>4 or 5 Layers</td></tr>
<tr><td colspan="2">The top layers focus more on the applications that need to send and receive data.</td></tr>
<tr><td colspan="2">The lower layers focus on the need to somehow transmit the bits from one device to another.</td></tr>
</table>

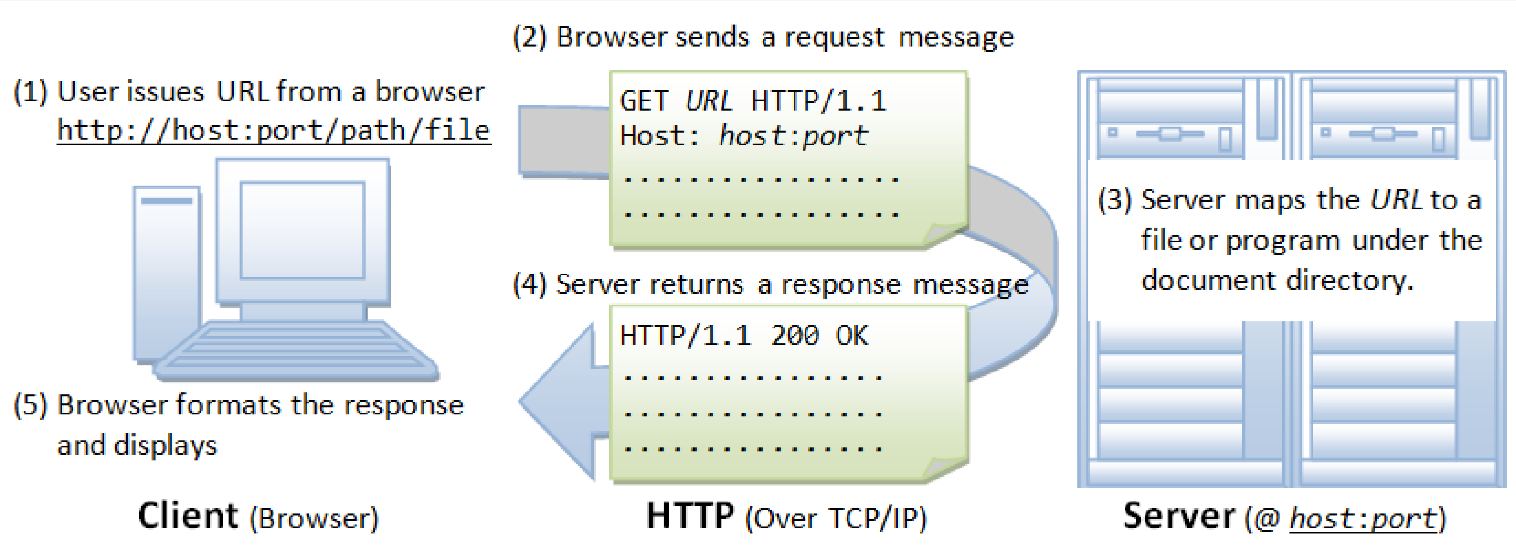


TCP/IP	OSI Model	Protocols
Application Layer	Application Layer	DNS, DHCP, FTP, HTTPS, IMAP, LDAP, NTP,POP3, RTP, RTSP, SSH, SIP, SMTP, SNMP, Telnet, TFTP
	Presentation Layer	JPEG, MIDI, MPEG, PICT, TIFF
	Session Layer	NetBIOS, NFS, PAP, SCP, SQL, ZIP
Transport Layer	Transport Layer	TCP, UDP
Internet Layer	Network Layer	ICMP, IGMP, IPsec, IPv4, IPv6, IPX, RIP
Link Layer	Data Link Layer	ARP, ATM, CDP, FDDI, Frame Relay, HDLC, MPLS, PPP, STP, Token Ring
	Physical Layer	Bluetooth, Ethernet, DSL, ISDN, 802.11 Wi-Fi

Application Layer

- Defines protocols providing services to the application software (processes) running on a computer, an interface between software running on a computer and the network itself.

  Example:
  HTTP defines how web browsers can pull the contents of a web page from a web server.



(1) User issues URL from a browser
http://host:port/path/file
(2) Browser sends a request message
GET URL HTTP/1.1
Host: host:port
(3) Server maps the URL to a file or program under the document directory.
(4) Server returns a response message
HTTP/1.1 200 OK
(5) Browser formats the response and displays
Client (Browser)
HTTP (Over TCP/IP)
Server (@ host:port)

Transport Layer

- Provides host-to-host communication services for the application layer (applications) such as:
  - Connection-oriented data stream support
  - Reliability
  - Flow control
  - Multiplexing

Most common protocols:

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

Many application layer protocols including HTTP, HTTPS, FTP and so on require a way to guarantee delivery of data across a network.

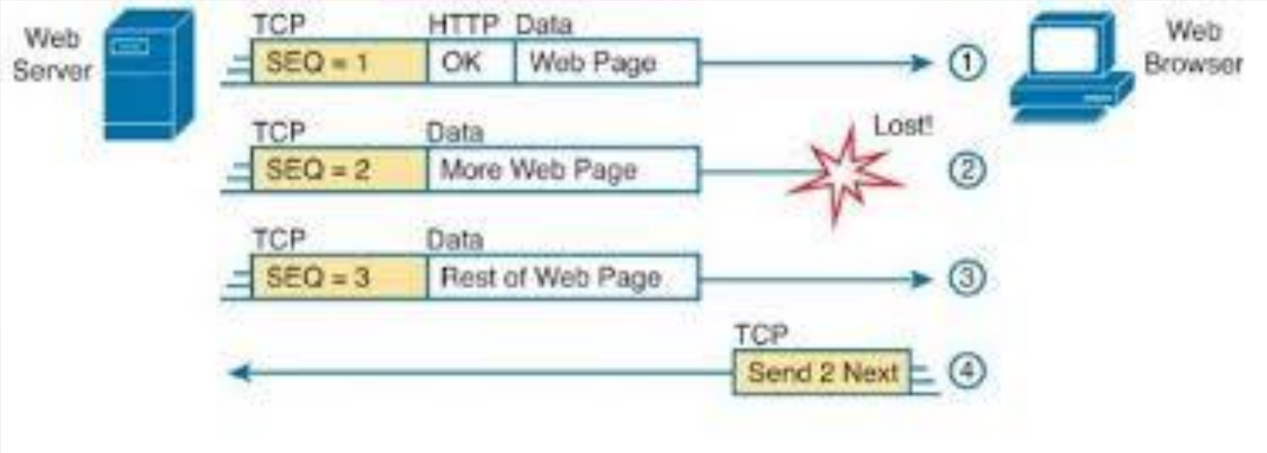
- How is it done that application layer protocols can transmit data through a reliable communication channel?

ANSWER:

- Adjacent-layer interaction: Each layer provides a service to the layer above it.

In case of TCP, it provides the following mechanisms:

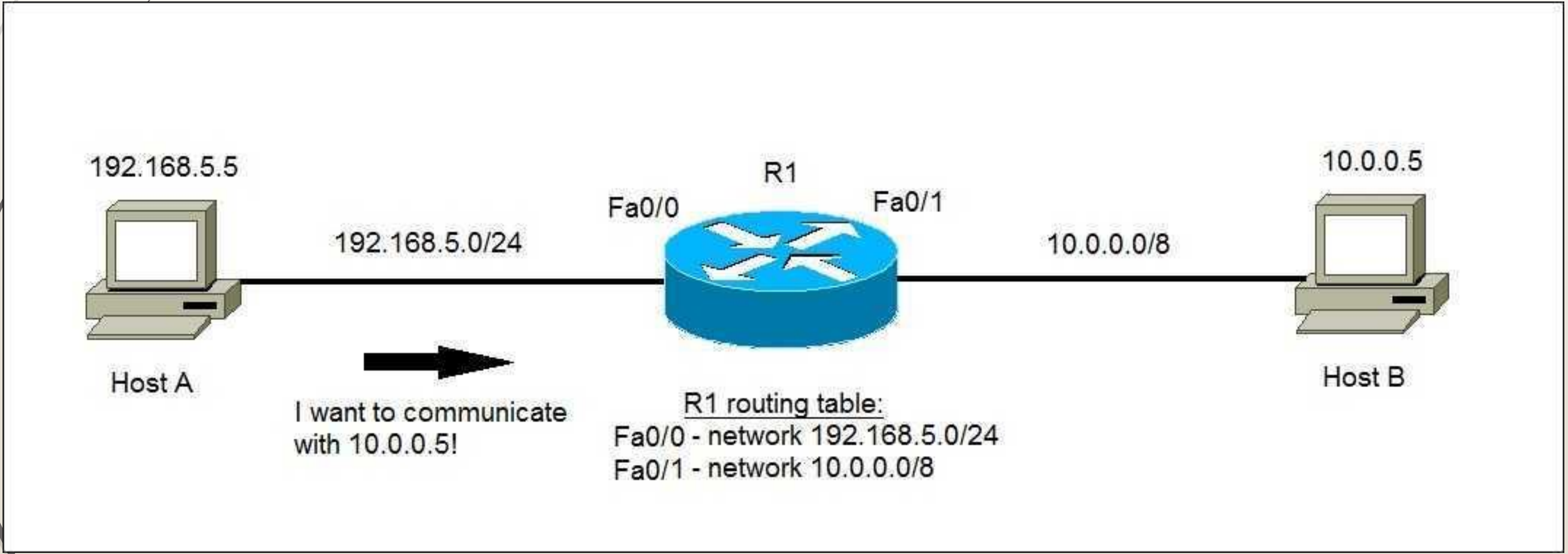
- Reliability: Packets may be lost during transport due to errors, network congestion. TCP can verify that by the acknoledgment mechanism.
- Same order delivery: The network/internet layer doesn't generally guarantee packets of data will arrive in the same order that they were sent.



Internet Layer

- Defines protocols and specifications used to transport packets/datagrams from the originating host to the destination host.
- The major protocol – IP, provides the most important features:
  - Addressing
  - Routing
- The Internet Layer works just like the postal service – to deliver messages to the correct destinations.

- Addressing: IP defines addresses so that each host computer can have a different IP address, just like the postal service defines addressing that allows unique addresses for house etc.
- Routing: IP defines the process of routing so that devices called routers can forward the packets of data so that they are delivered to the correct destinations.
- In routing, IP packets are forwarded from one device to another. Any device with an IP address can connect to the TCP/IP network and send packets.
- An IP address is basically used to uniquely identify a network interface (device) in a computer network.



R1 routing table:
Fa0/0 - network 192.168.5.0/24
Fa0/1 - network 10.0.0.0/8

TCP/IP Network Access/Link Layer

- Defines the protocols and hardware required to deliver data across some physical network.
- Defines physical media over which data can be transmitted.
- The network access layer provides services to the layer above in the model.
- When a host or router's IP process chooses to send an IP packet to another router or host, that host or router then uses network access layer details to send the packet to the next host/router.

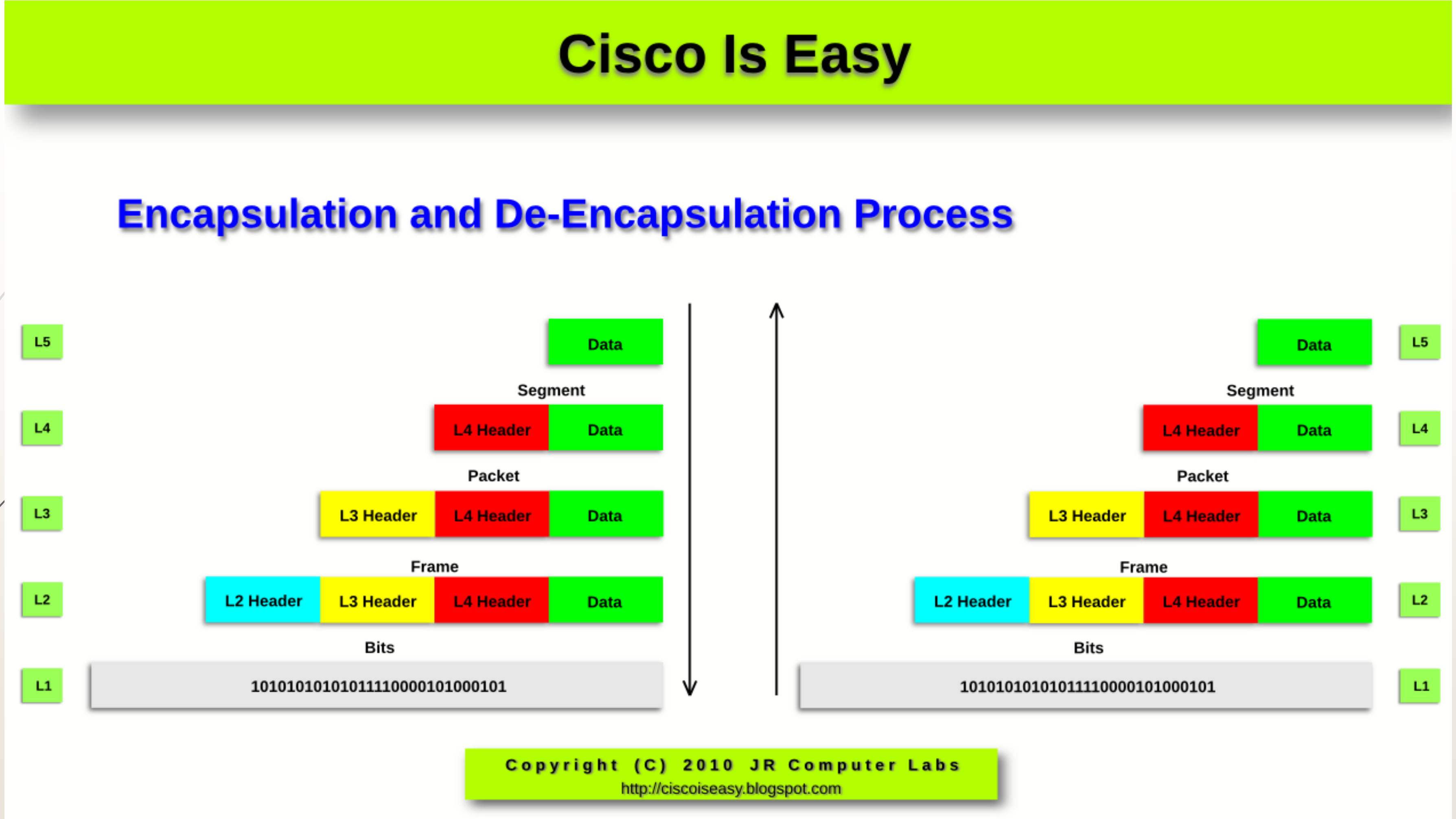
- Examples:
  - Ethernet (IEEE 802.3)
  - PPP (Point-to-point Protocol)
  - FDDI
  - IEEE 802.11 (WiFi)
  - Bluetooth

Data Encapsulation

- Encapsulation: The process of putting headers (and sometimes trailers) around some data.
- Protocols define headers and trailers for the same general reason, but headers exist at the beginning of the message and trailers exist at the end.

- Steps

1. Create and encapsulate the application data with any required application header e.g. An HTTP request with some headers and contents of a web page.
2. Encapsulate the data supplied by the application header inside a transport layer header e.g. A TCP header (SEGMENT)
3. Encapsulate the data supplied by the transport inside an IP header (PACKET)
4. Encapsulate the data supplied by the Internet layer inside a data link layer header and trailer e.g. Ethernet, WiFi (IEEE 802.11)
5. Transmit the bits. The physical layer encodes a signal onto the medium to transmit the frame.

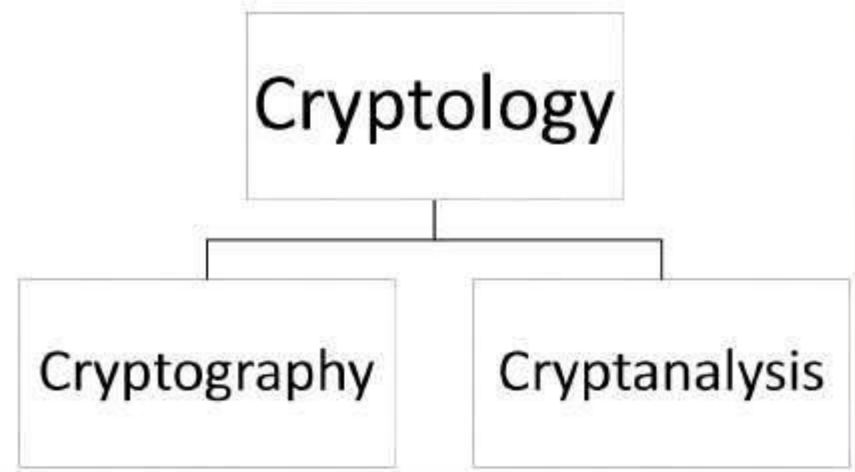


Cisco Is Easy

Encapsulation and De-Encapsulation Process

http://ciscoiseasy.blogspot.com

Cryptograhpy Foundations



Cryptology

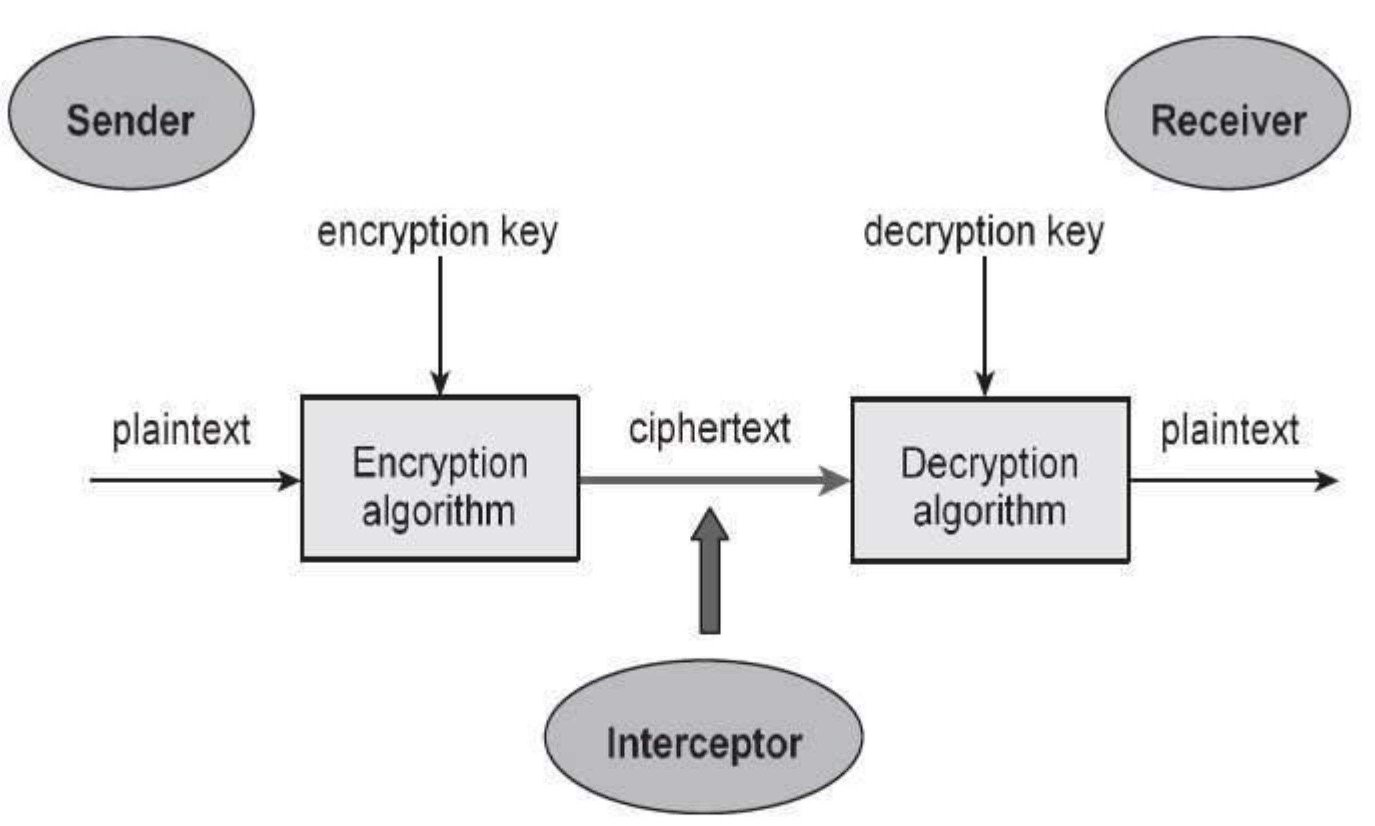
Cryptography

Cryptanalysis

- Cryptograhpy: The art and science of making a cryptosystem that's capable of providing information security.
- It deals with the actual securing of digital data, referrring to the design of mechanisms based on math algorithms providing information security services
- Cryptoanalysis: The art and science of breaking the cipher text.
- The cryptograhic process results in the ciphe text for transmission or storage.

Basic Concepts

- Cryptosystem: An implementation of cryptographic techniques and their infrastructure to provide information security services.



Components of a cryptosystem

- **Plaintext.** It is the data to be protected during transmission.
- **Encryption Algorithm.** It is a mathematical process that produces a ciphertext for any given plaintext and encryption key. It is a cryptographic algorithm that takes plaintext and an encryption key as input and produces a ciphertext.
- **Ciphertext.** It is the scrambled version of the plaintext produced by the encryption algorithm using a specific the encryption key. The ciphertext is not guarded. It flows on public channel. It can be intercepted or compromised by anyone who has access to the communication channel.
- **Decryption Algorithm,** It is a mathematical process, that produces a unique plaintext for any given ciphertext and decryption key. It is a cryptographic algorithm that takes a ciphertext and a decryption key as input, and outputs a plaintext. The decryption algorithm essentially reverses the encryption algorithm and is thus closely related to it.

- **Encryption Key.** It is a value that is known to the sender. The sender inputs the encryption key into the encryption algorithm along with the plaintext in order to compute the ciphertext.
- **Decryption Key.** It is a value that is known to the receiver. The decryption key is related to the encryption key, but is not always identical to it. The receiver inputs the decryption key into the decryption algorithm along with the ciphertext in order to compute the plaintext.
- For a given cryptosystem, a collection of all possible decryption keys is called a **key space**.
- An **interceptor** (an attacker) is an unauthorized entity who attempts to determine the plaintext. He can see the ciphertext and may know the decryption algorithm. He, however, must never know the decryption key.

Types of Cryptosystems

- Symmetric Key Encryption
- Asymmetric Key Encryption

References

- Cisco CCNA ICND1, Wendell Odom
- https://en.wikipedia.org/wiki/Internet_protocol_suite
- https://www.tutorialspoint.com/cryptography/