IN2120 Information Security

Lecture 4: Communications Security

Nils Gruschka

University of Oslo

Autumn 2018

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

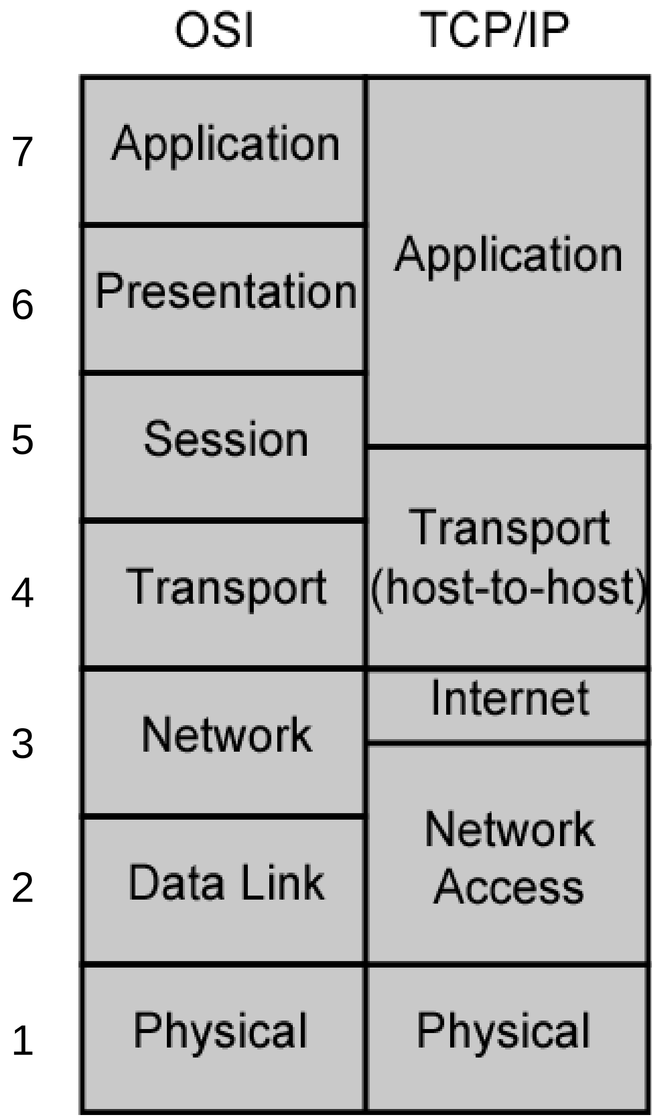
Assumes that each organisation owns a network

- 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

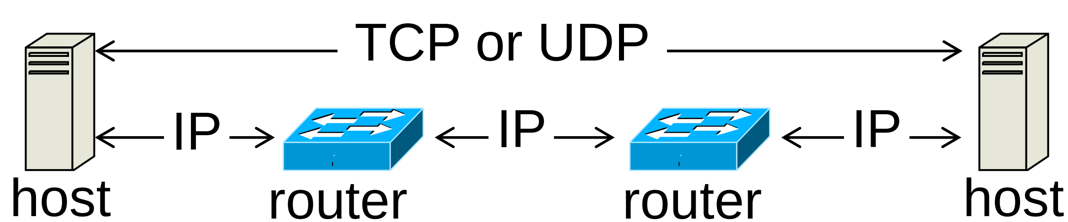
OSI model vs. TCP/IP model (The Internet)



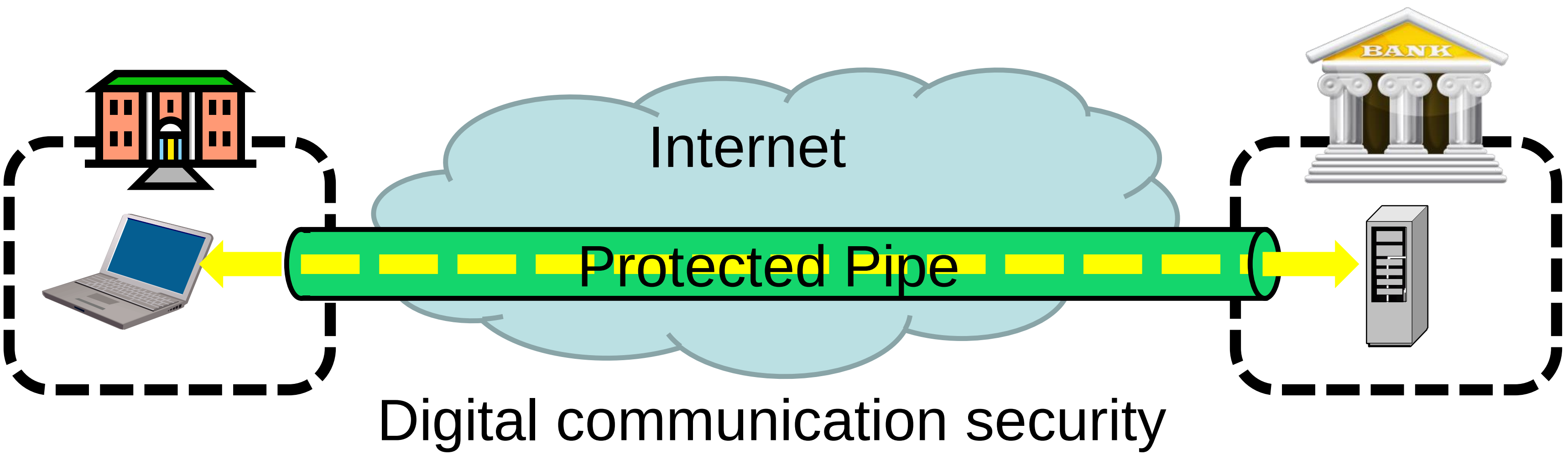
Application protocols,
e.g. http, ftp, smtp, snmp

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

IP (Internet Protocol)



Communication Security Analogy



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.

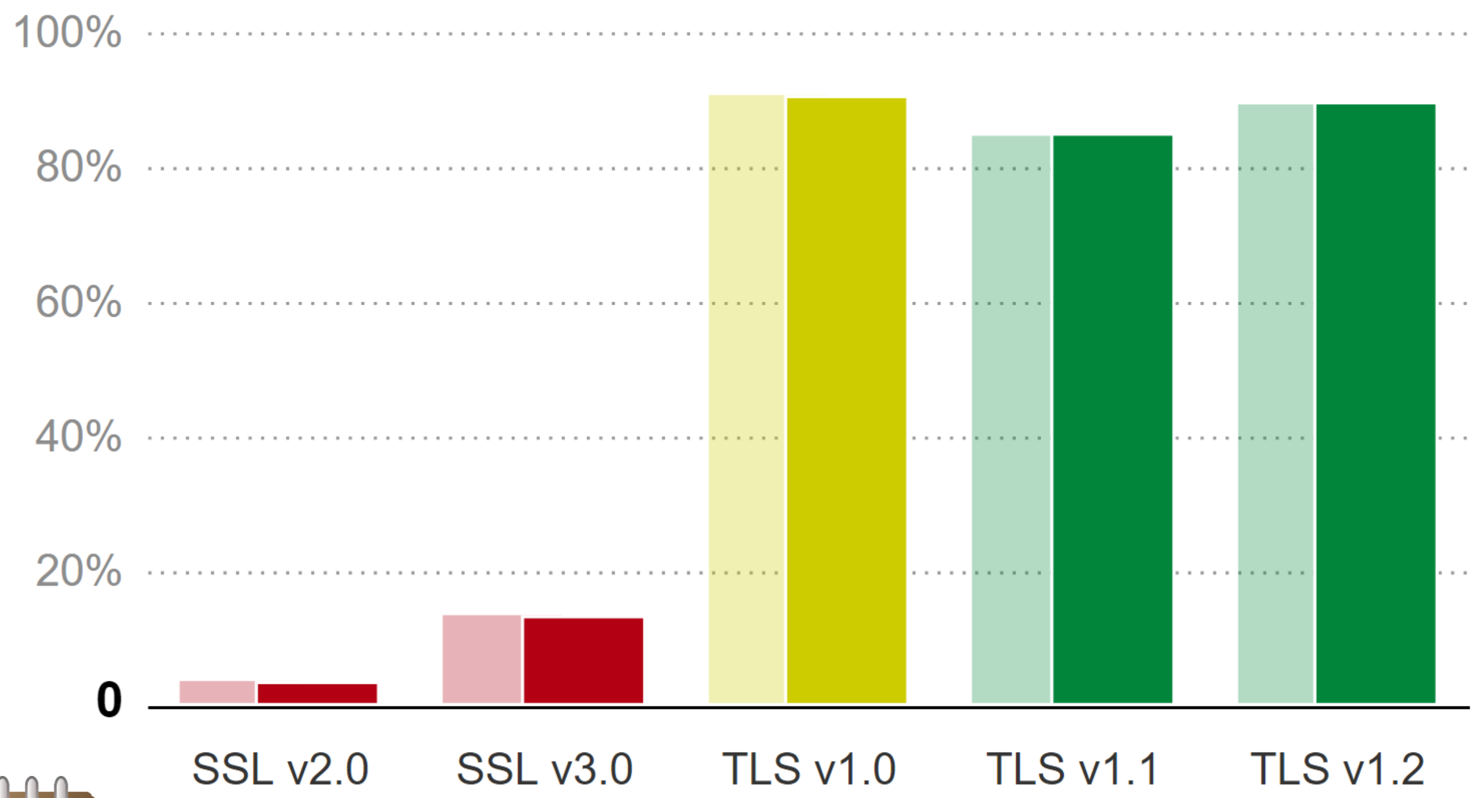
Transport Layer Security

TLS/SSL

SSL/TLS: History

- 1994: Netscape Communications developed the network authentication protocol Secure Sockets Layer, SSLv2.
  - Badly broken, officially deprecated 2011
- 1995: Netscape release their own improvements SSLv3.
  - Broken, officially deprecated 2015
- In January 1999, RFC 2246 was issued by the IETF, Transport Layer Security Protocol: TLS 1.0
  - Similar to, but incompatible with SSLv3
  - Followed by TLS 1.1 (2006) and TLS 1.2 (2008)
  - Current version: TLS 1.3 (2018), removes all old/insecure features/algorithms

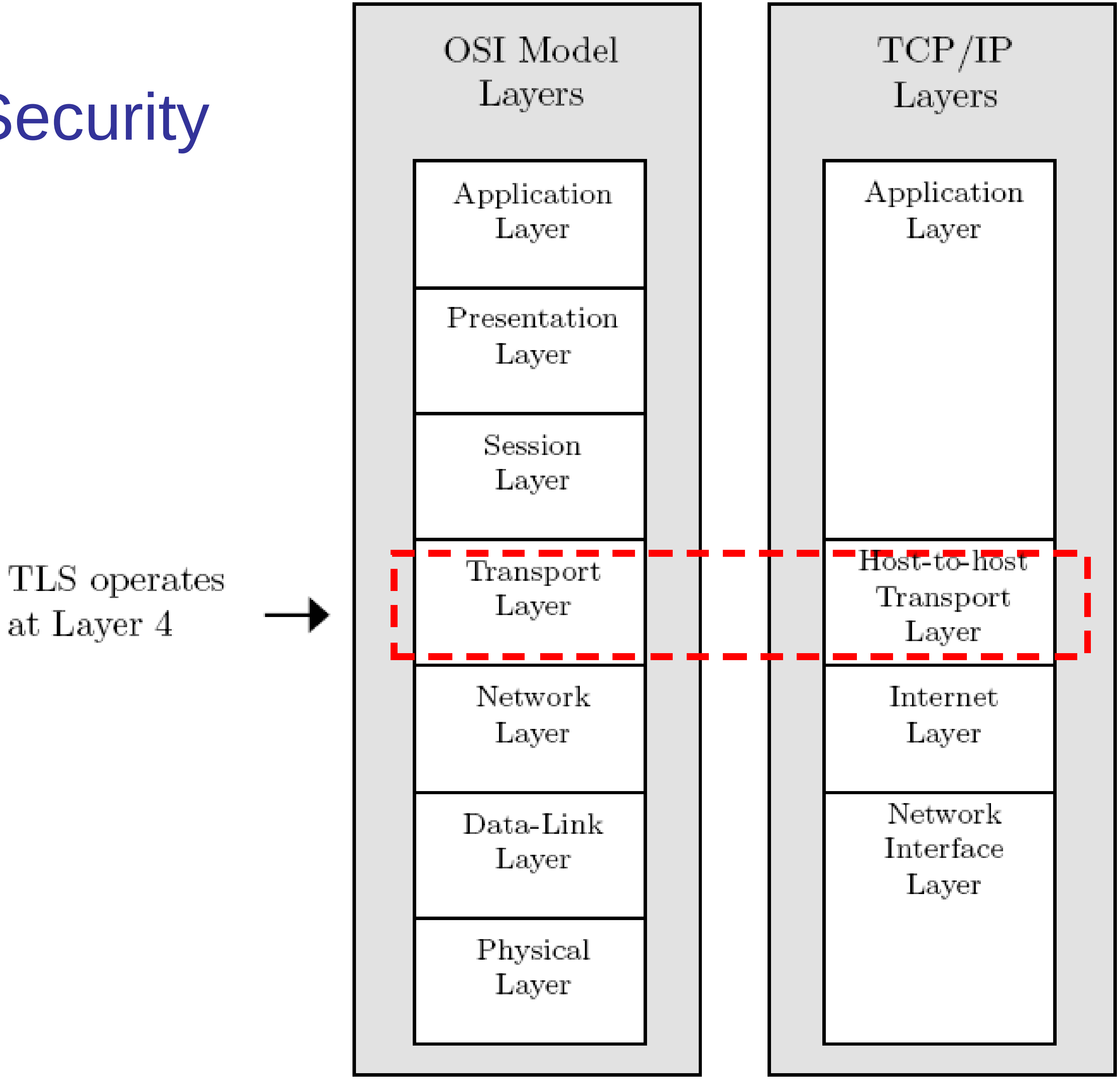
SSL/TLS Protocol versions



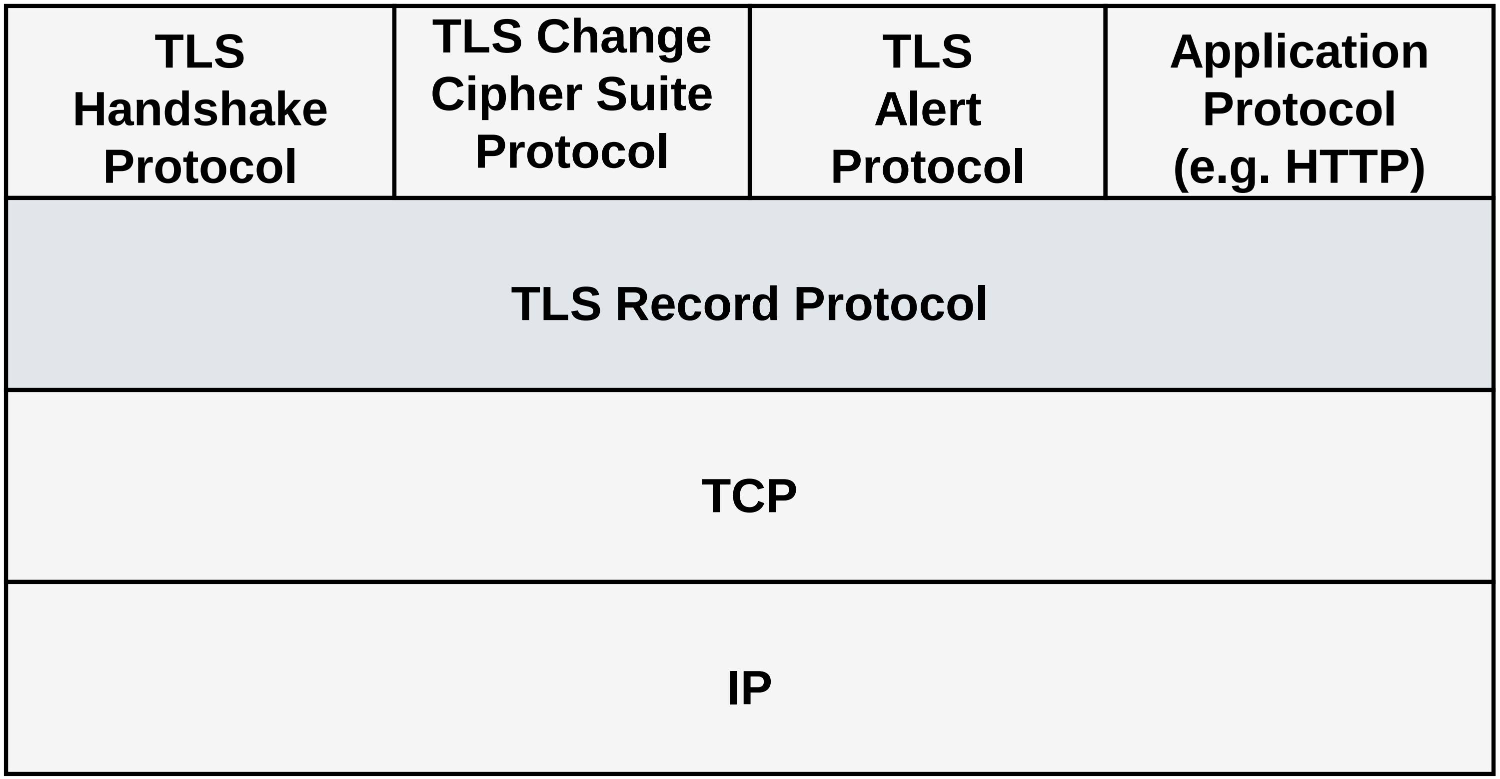
TLS: Overview

- 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



TLS: Protocol Stack



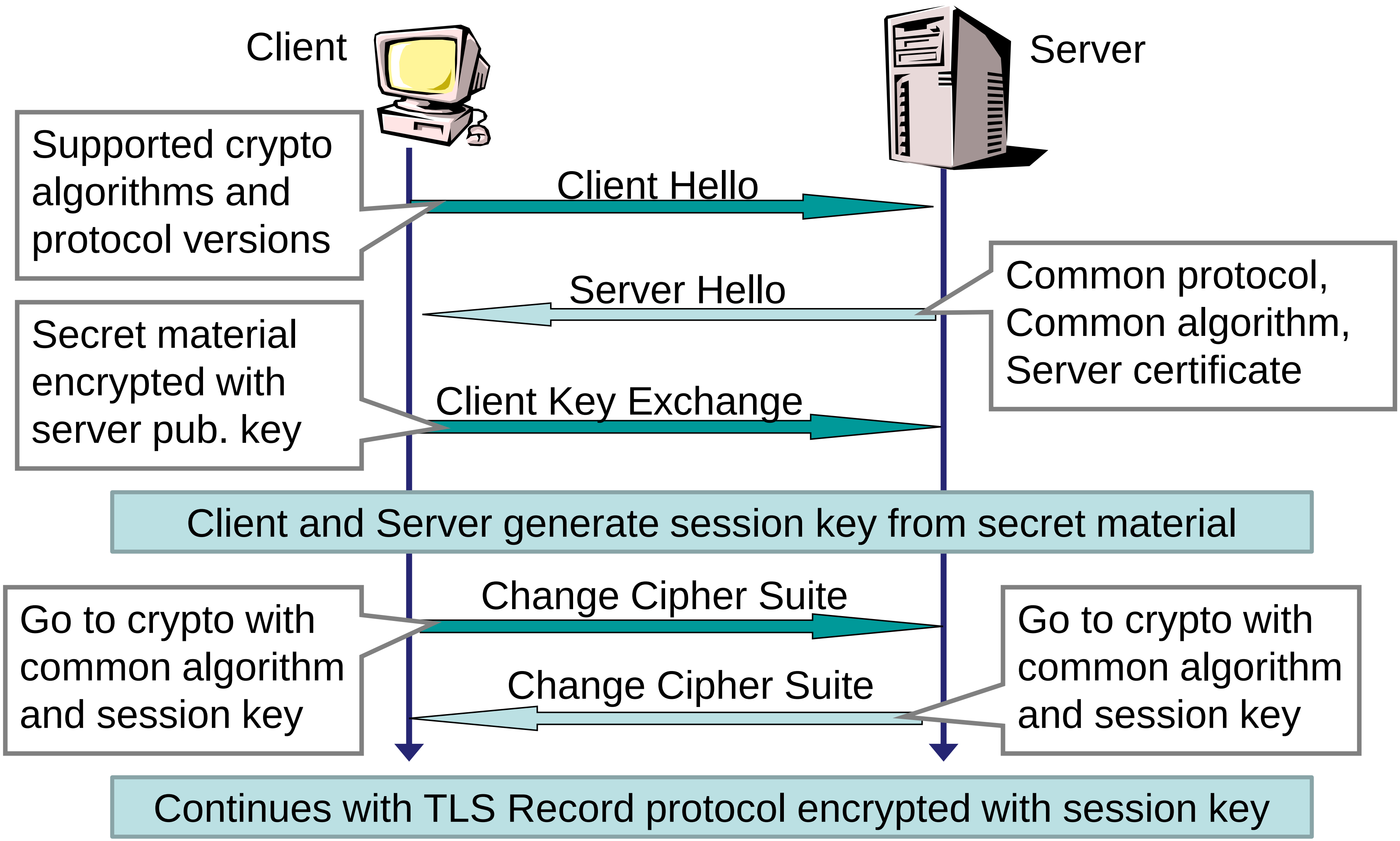
TLS: Architecture Overview

- Designed to provide secure reliable end-to-end services over TCP.
  - Confidentiality
  - Integrity
  - Authenticity
- 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

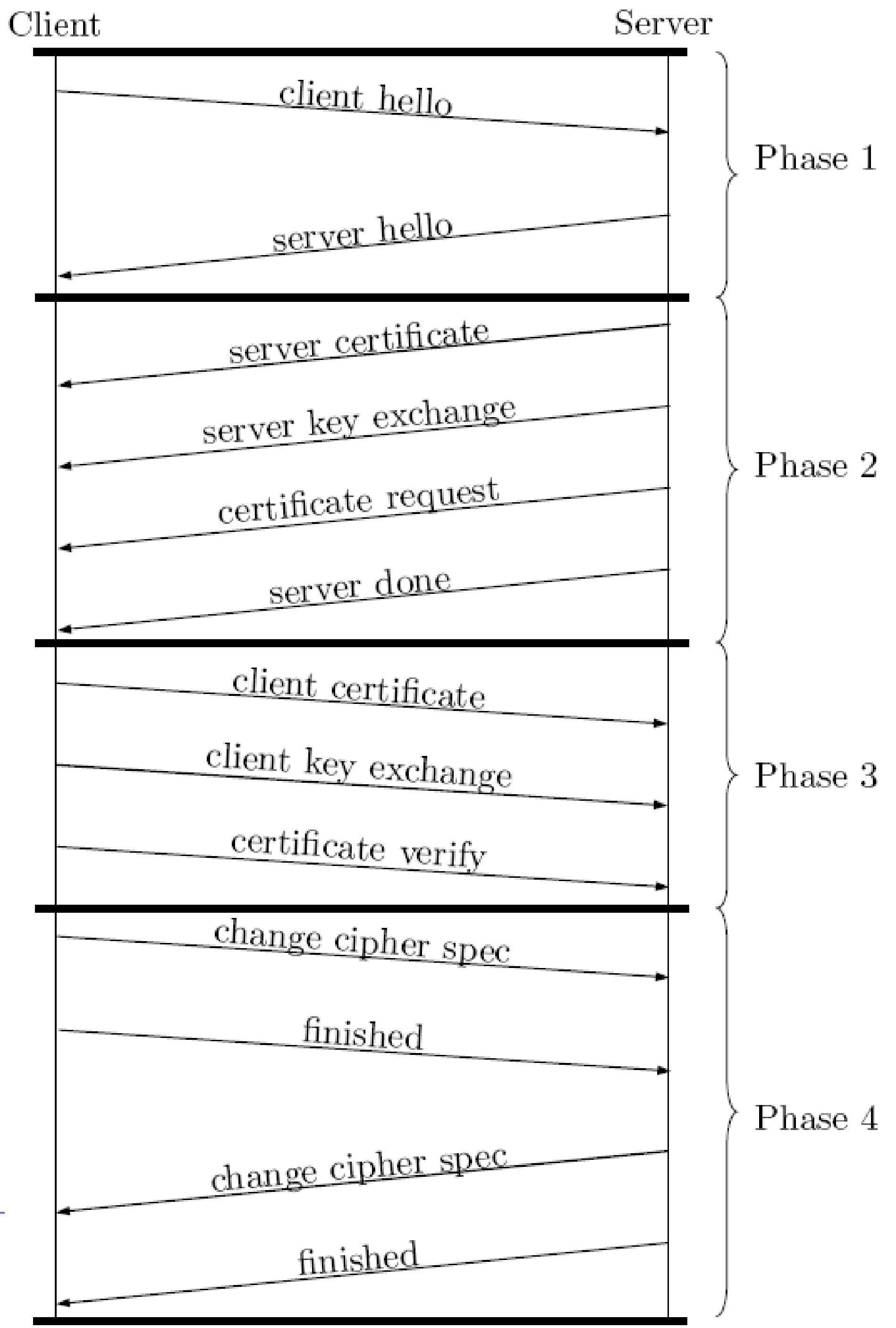
- The handshake protocol
  - Negotiates the encryption to be used
  - Establishes a shared session key
  - Authenticates the server
  - Authenticates the client (optional)
- After the handshake, application data is transmitted securely (encrypted + integrity protected
- Several variations of the handshake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

TLS: Simplified RSA-based Handshake



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: 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 1.2): TLS_RSA_WITH_AES_256_CBC_SHA256
    - RSA for key exchange
    - AES (128 bit key) with CBC mode for encryption
    - SHA256 as hash function for authentication and integrity protection
  - Example (TLS 1.3): TLS_AES_256_GCM_SHA384
    - DH for key exchange (implicit)
    - AES with GCM for encryption + integrity protection
    - SHA384 as hash function for authentication

TLS: Elements of Handshake

- 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.
- 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 TLS connections.
  - Message Confidentiality:
    - Ensure that the message contents cannot be read in transit.
    - The Handshake Protocol establishes a symmetric key used to encrypt TLS payloads using symmetric encryption.
  - 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.

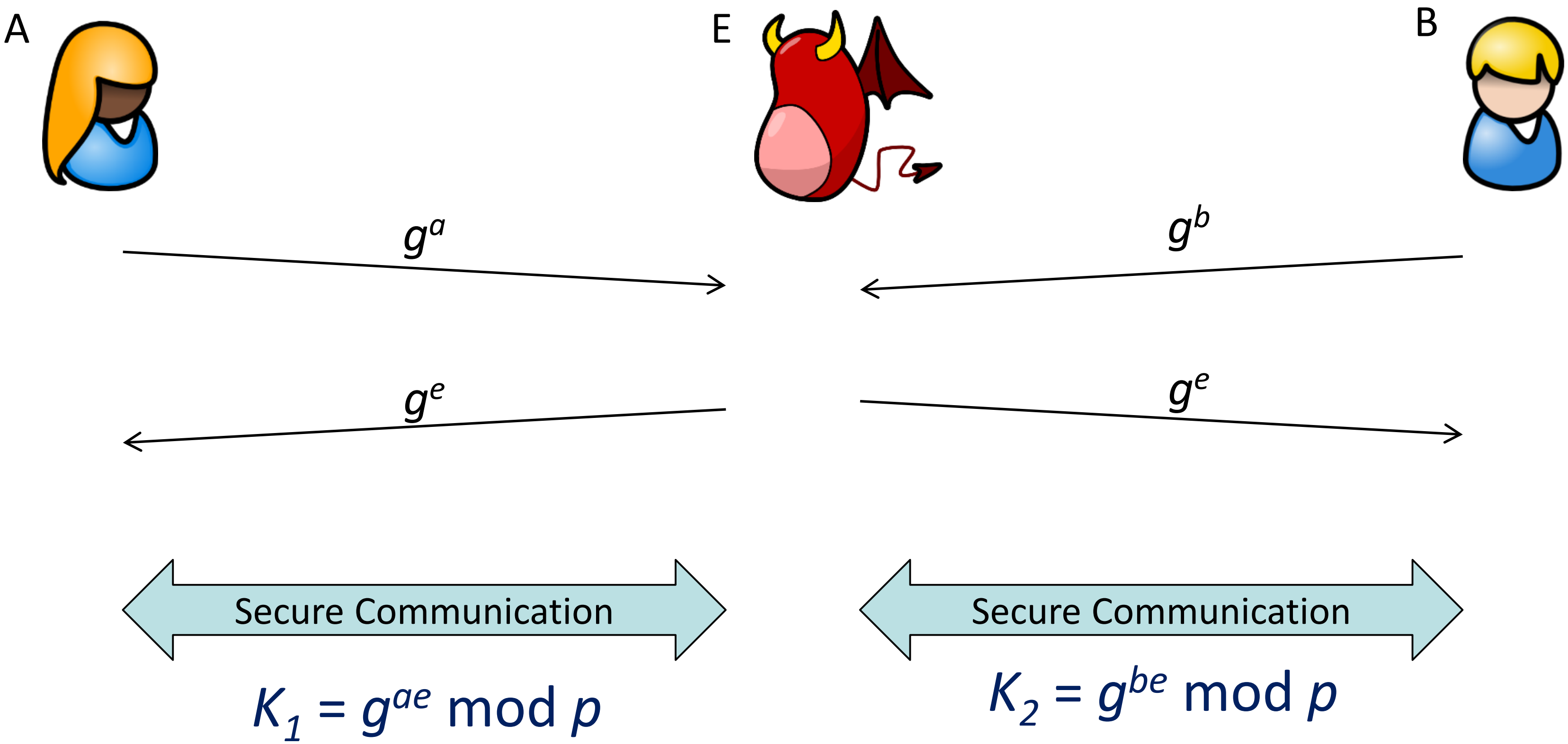
TLS: Record Protocol Operation

- 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.

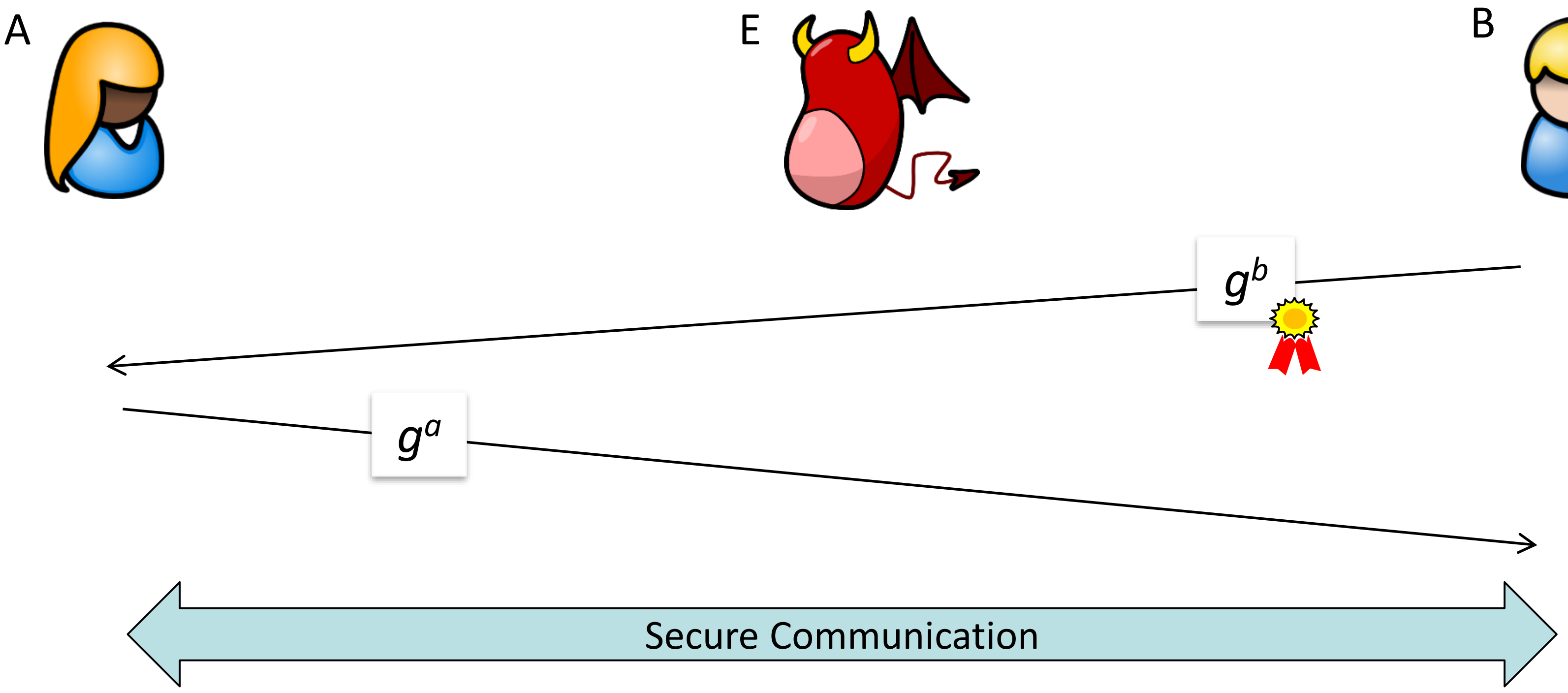
TLS: Key Exchange

- 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)

Weakness of DH Key Exchange



Countermeasure



$K = g^{ab} \bmod p$

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 server 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”

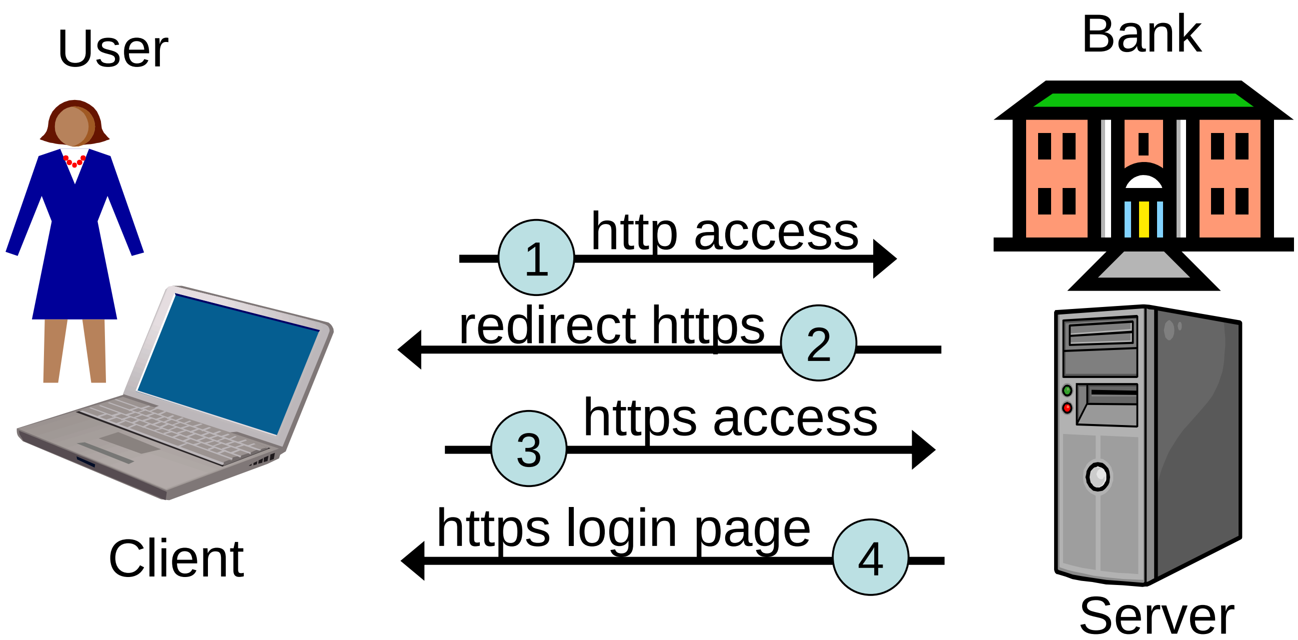
TLS Challenges

- Many vulnerabilities exist for TLS
  → keep client and server software up-to-date
- Also vulnerabilities in cryptographic algorithms
  → configure server to exclude week algorithms
- TLS provides security just for a single TCP connection
  - Browser can establish HTTP and HTTPS connections; even to the same server
  - User can be tricked to use HTTP instead of HTTPS (→ next slide)
- Relies on browser PKI which has many security issues
  - Fake server certificates difficult to detect (→ lecture “PKI”)
  - Fake root server certificate can compromise all certificates, e.g. Lenovo Komodia advare scam (→ lecture “Network Security”)

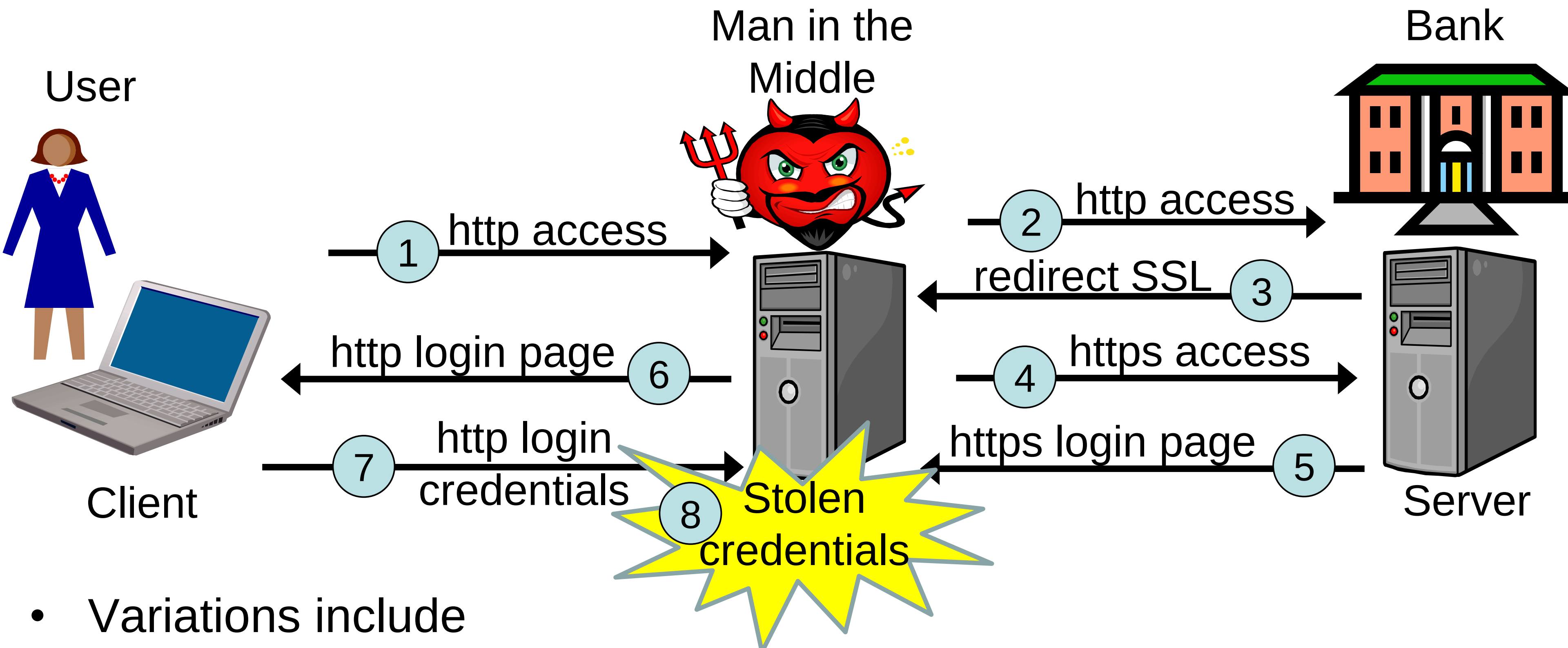
Demo

HTTPS redirect

- Typical for normal browsing behaviour



TLS Stripping Attack

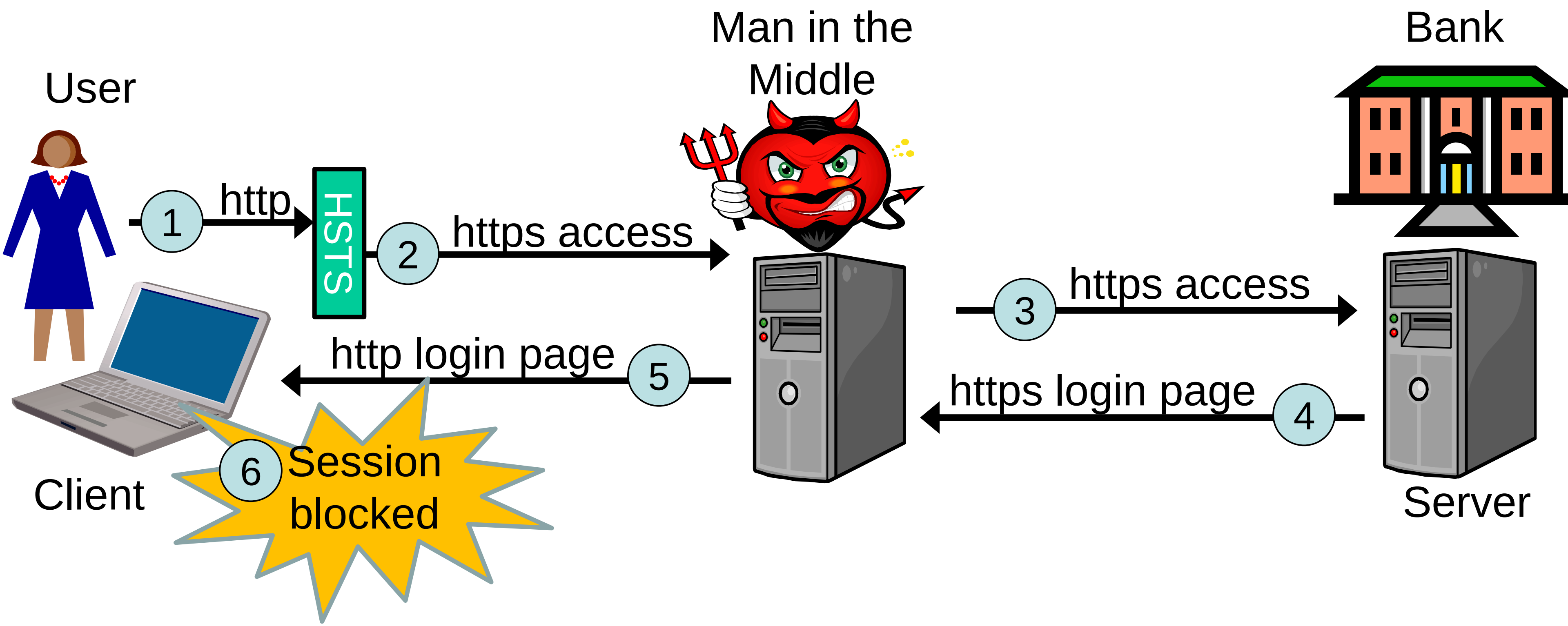


- Variations include
  - MitM server can connect to client over https in msg (6) with server certificate that has similar domain name as real server.
  - Attacker can leave the connection after stealing credentials, then the client connects directly to real server with https
  - Attacker just downgrades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

HSTS – HTTP Strict Transport Security
Preventing TLS 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

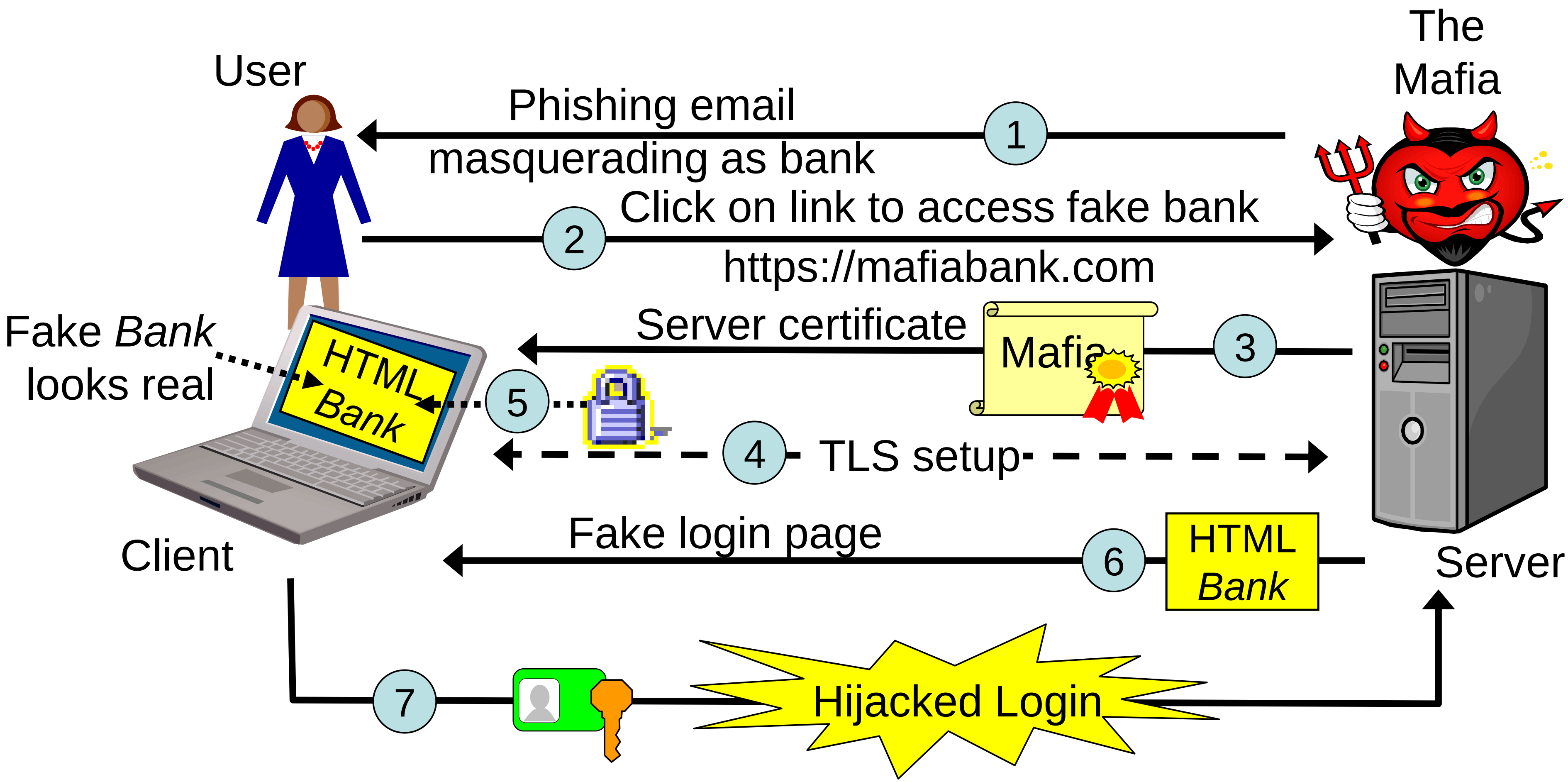
Preventing TLS Stripping with HSTS



- 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

Demo

Phishing and failed authentication



IP Layer Security

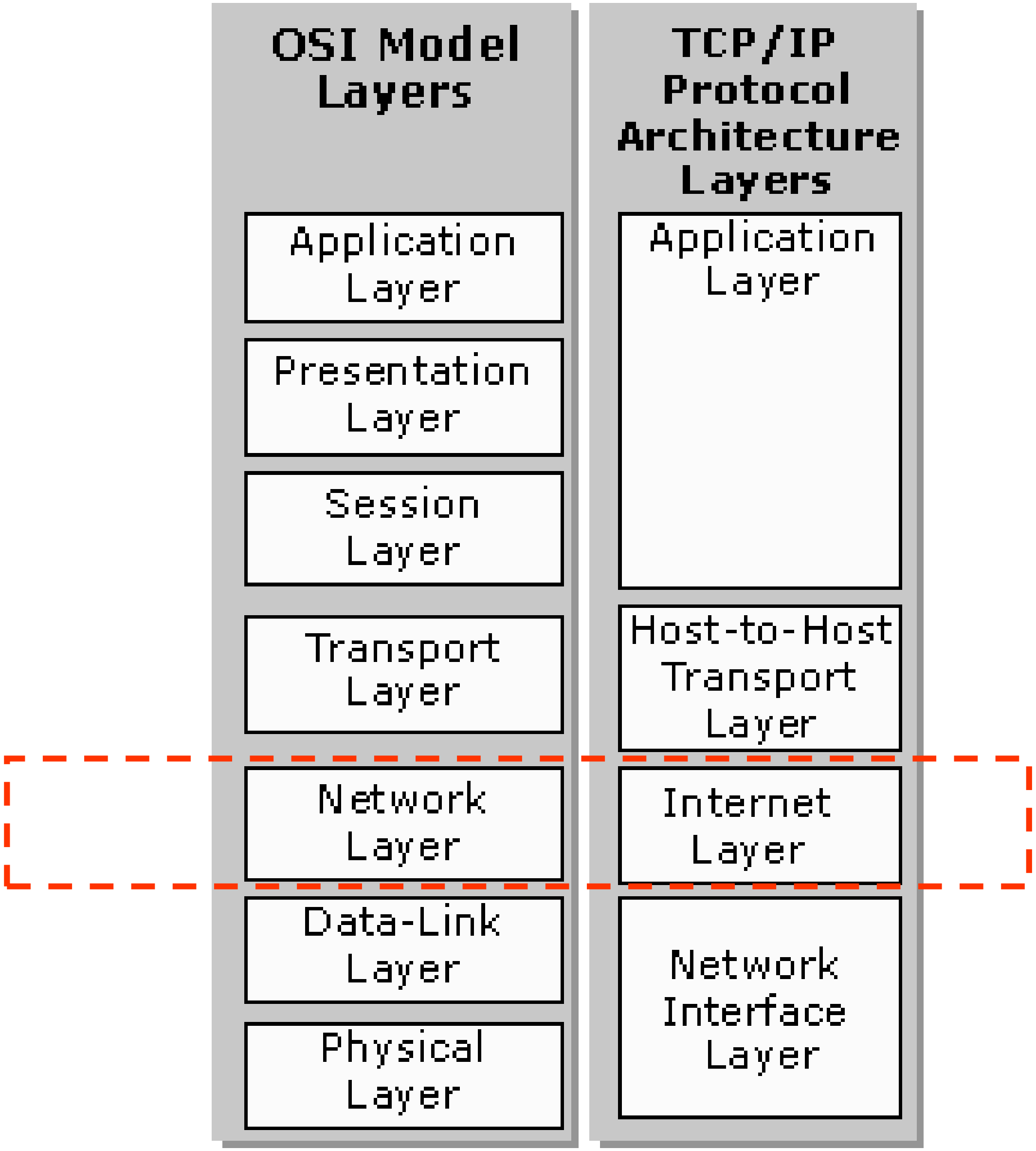
IPSec & Virtual Private Networks

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

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.

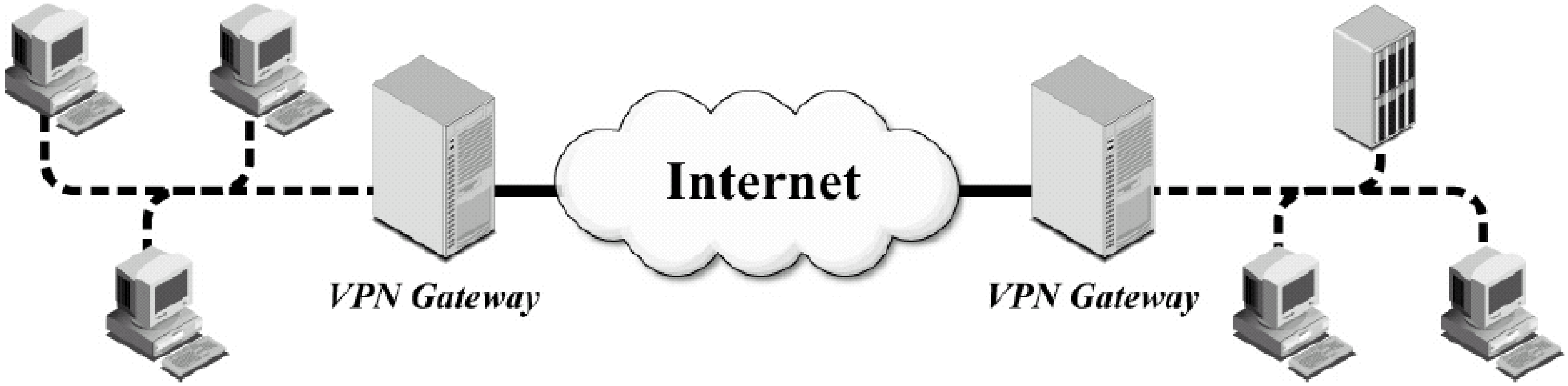
IPSec: Security Services

- 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

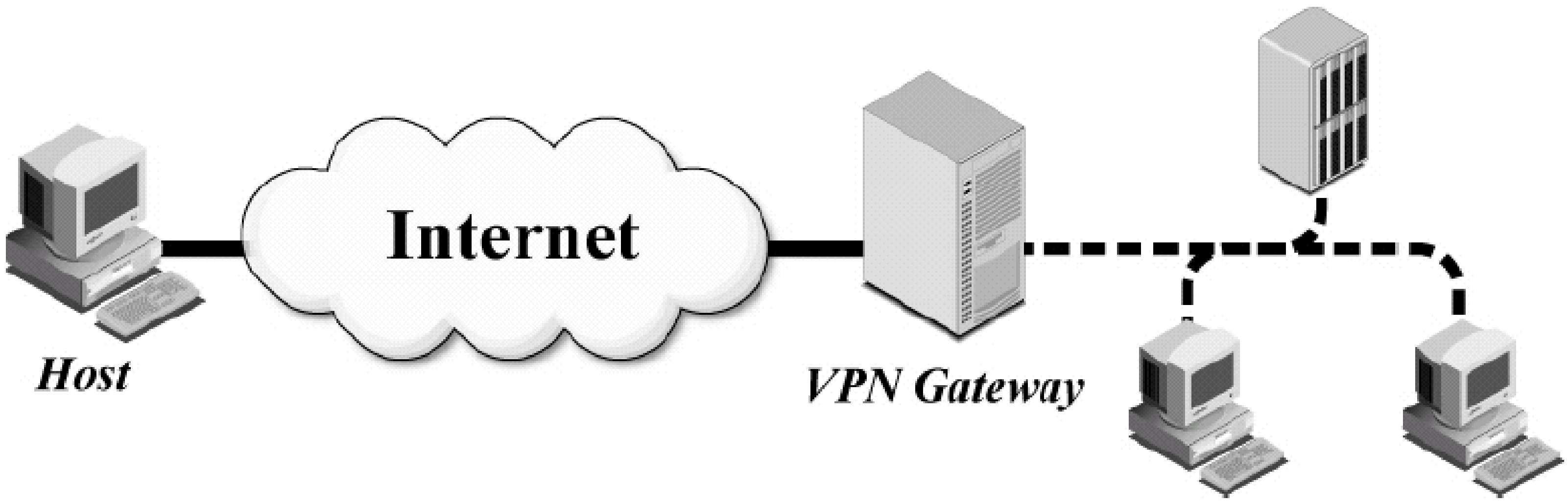
- Gateway-to-Gateway Architecture
- Host-to-Gateway Architecture
- Host-to-Host Architecture

IPSec: Gateway-to-Gateway Architecture



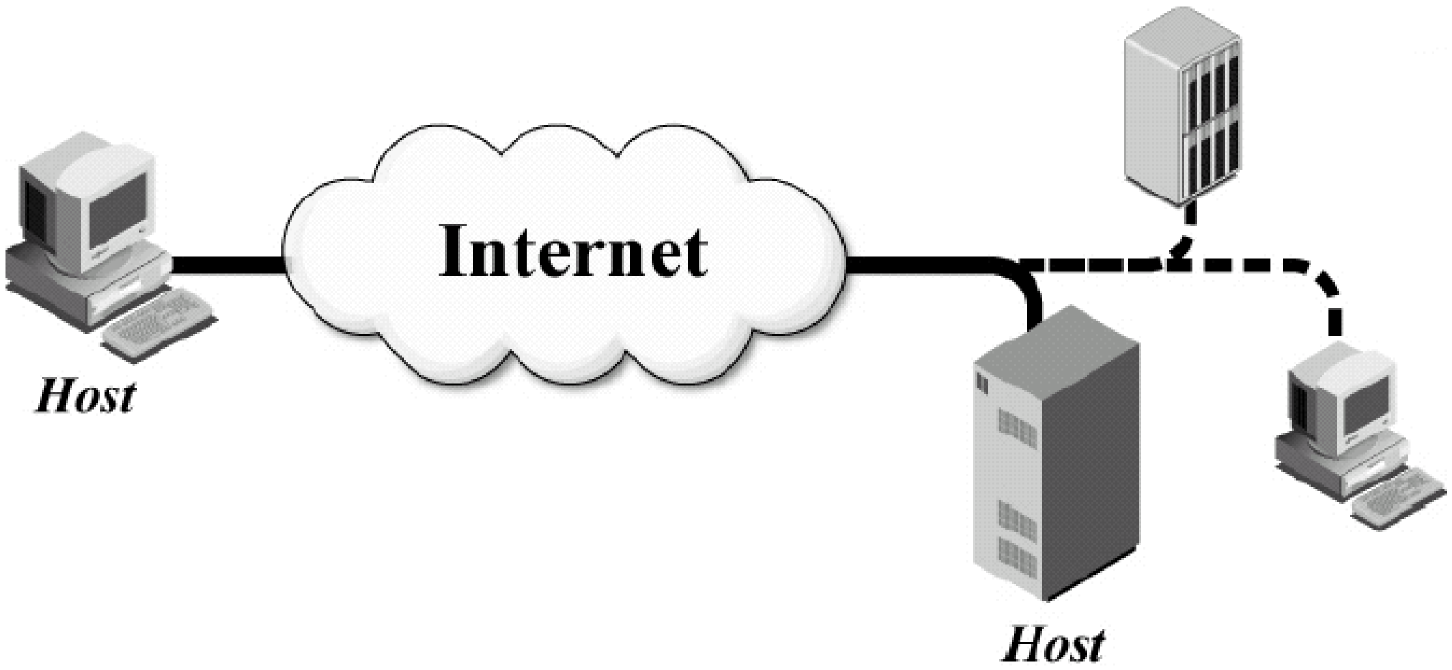
Source: NIST Special Publication 800-77

IPSec:
Host-to-Gateway Architecture



Source: NIST Special Publication 800-77

IPSec: Host-to-Host Architecture



Source: NIST Special Publication 800-77

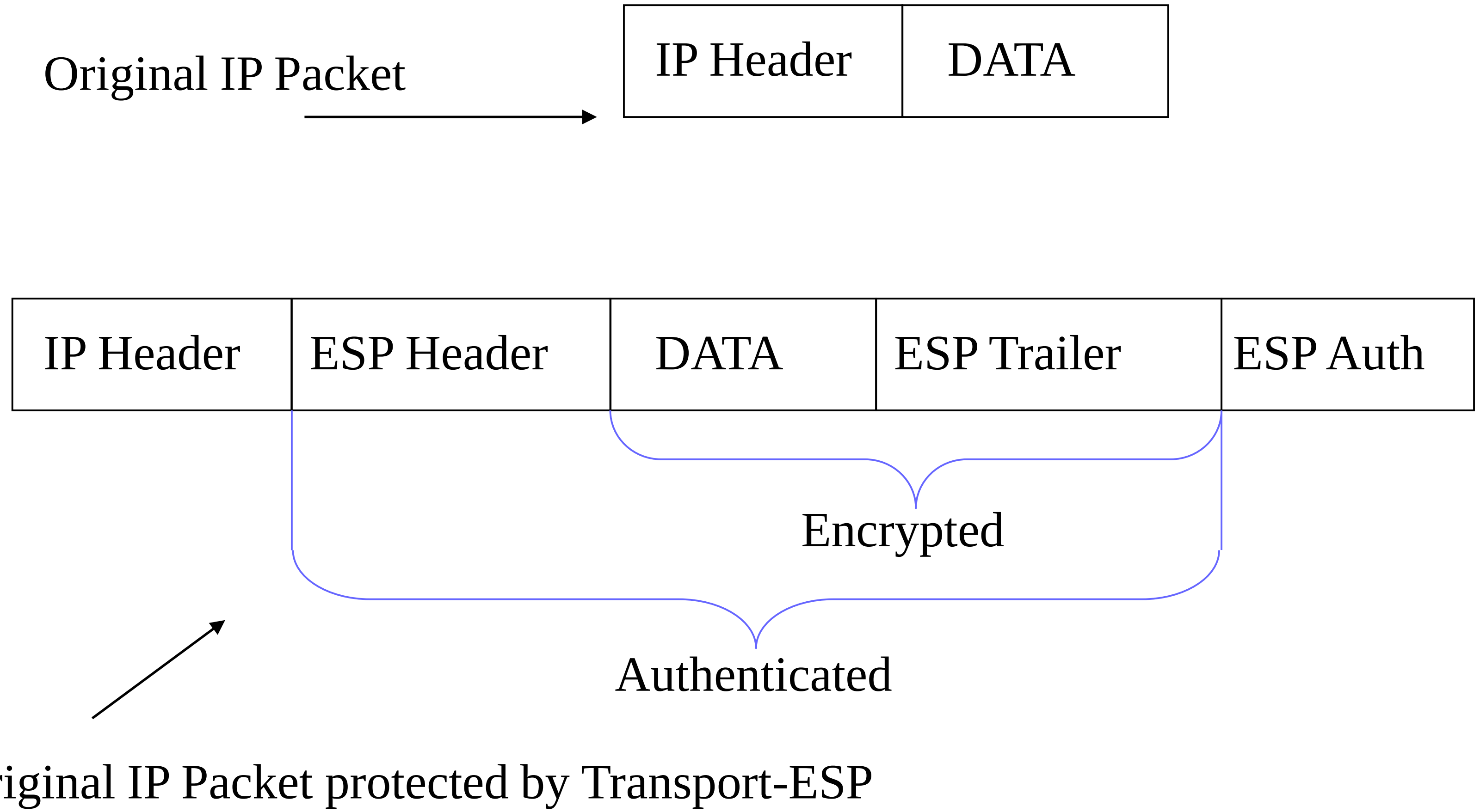
IPSec: Protocols Types

- Encapsulating Security Payload (ESP)
  - Confidentiality, authentication, integrity and replay protection
- Authentication Header (AH)
  - Authentication, integrity and replay protection. However there is no confidentiality
- Internet Key Exchange (IKE)
  - negotiate, create, and manage security associations
- A connection consists of two SA (Security Associations)
  - SA includes cryptographic keys and algorithms, key lifetimes, security parameter index (SPI), and security protocol identifier (ESP or AH).

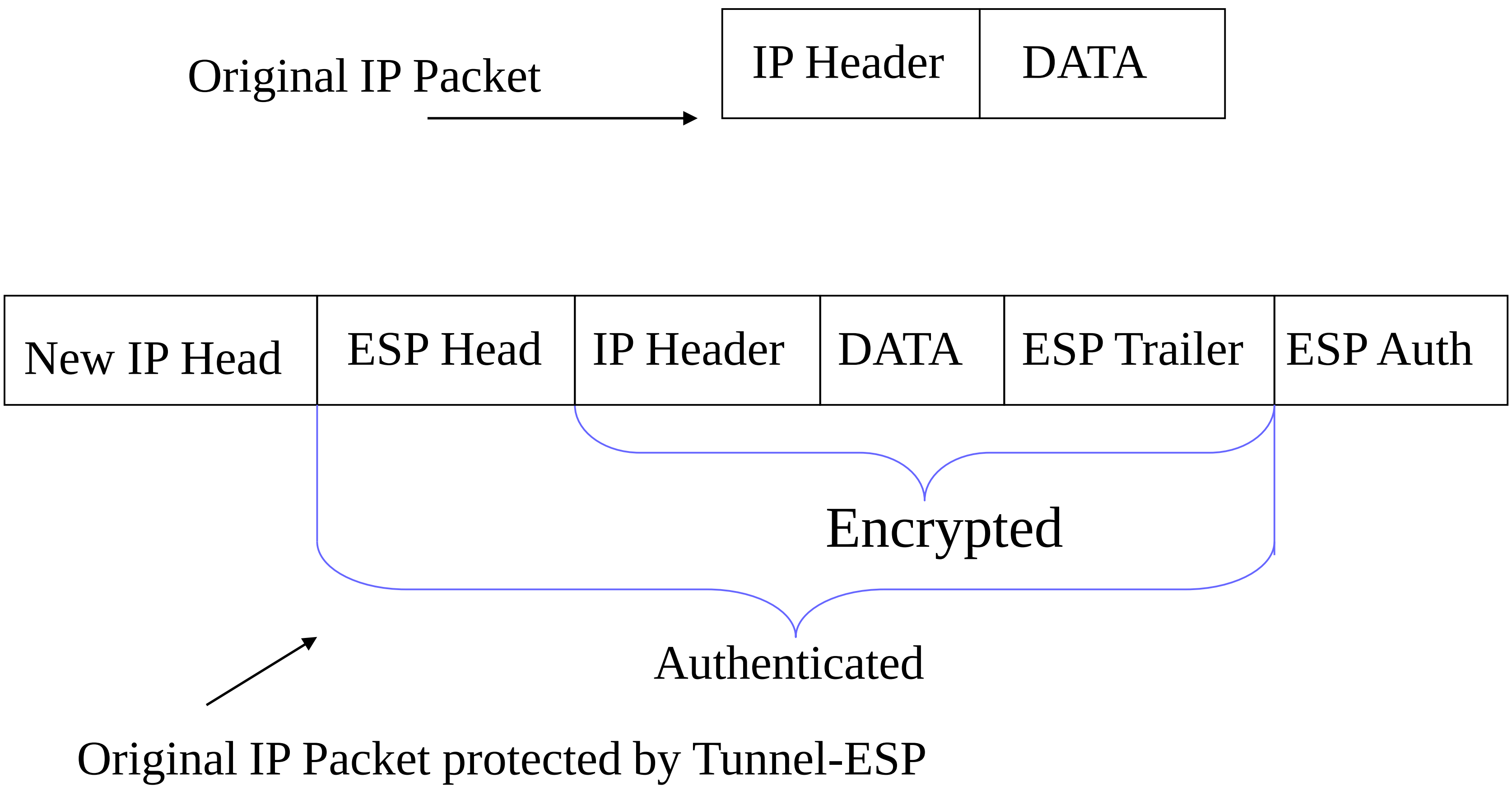
IPSec: Modes of operation

- 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

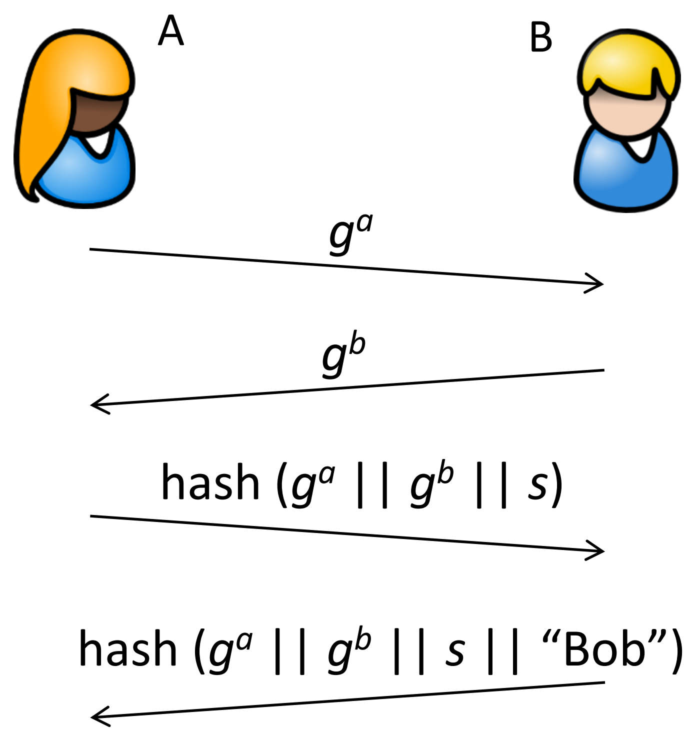


Tunnel Mode ESP

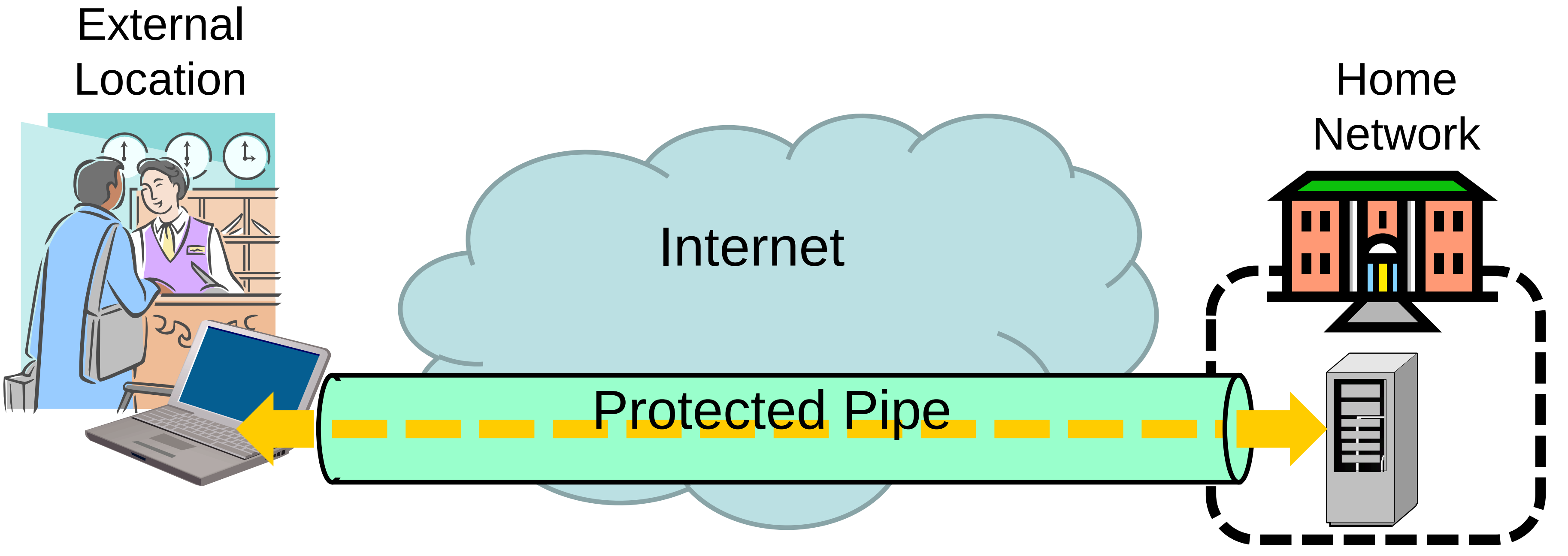


Internet Key Exchange

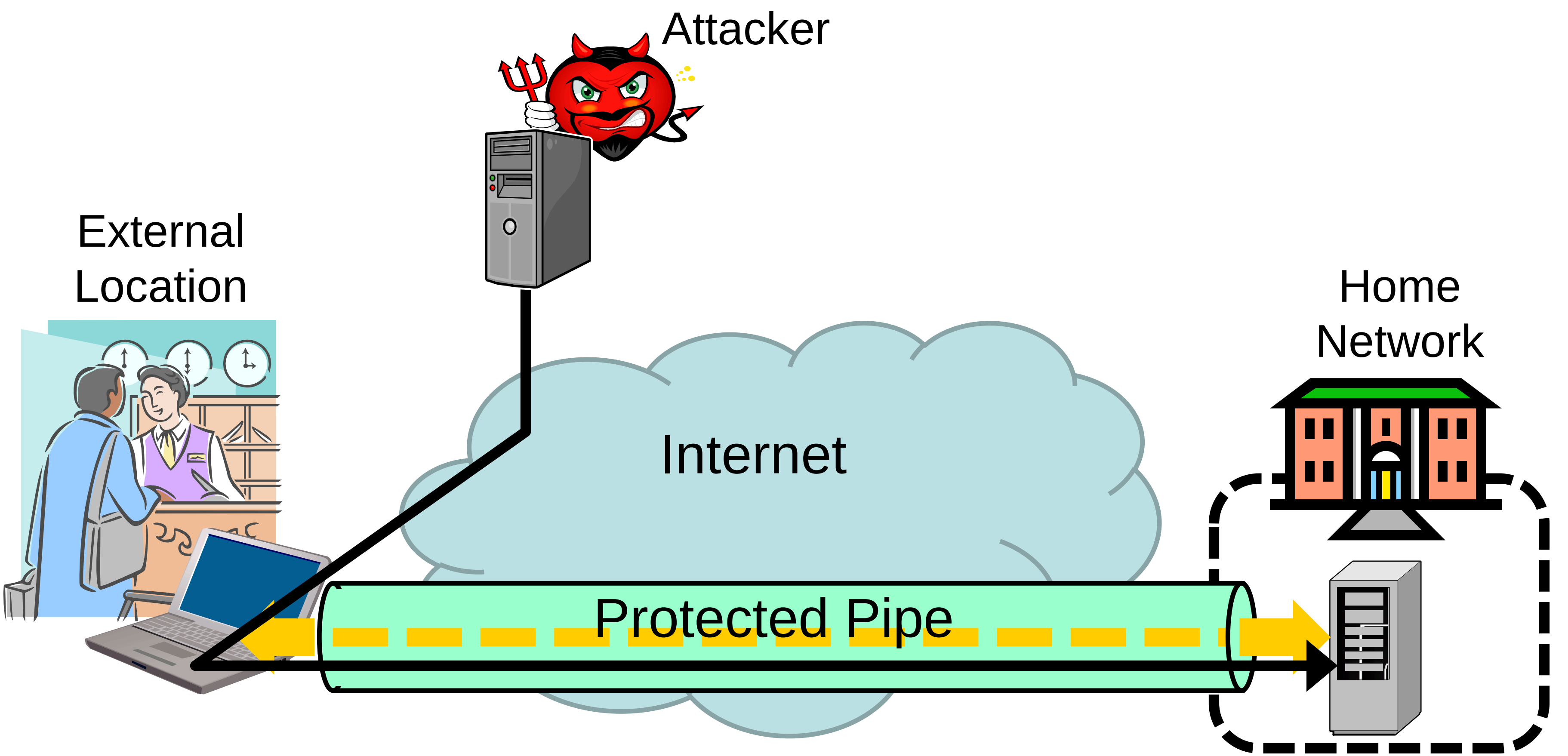
- Alice and Bob have common (long term) secret *s*
- DH exchange is **authenticated** (MITM not possible)
- After each session, session key is destroyed
- → **Perfect forward secrecy**



Typical usage of IPSec: VPN



Risk of using VPN

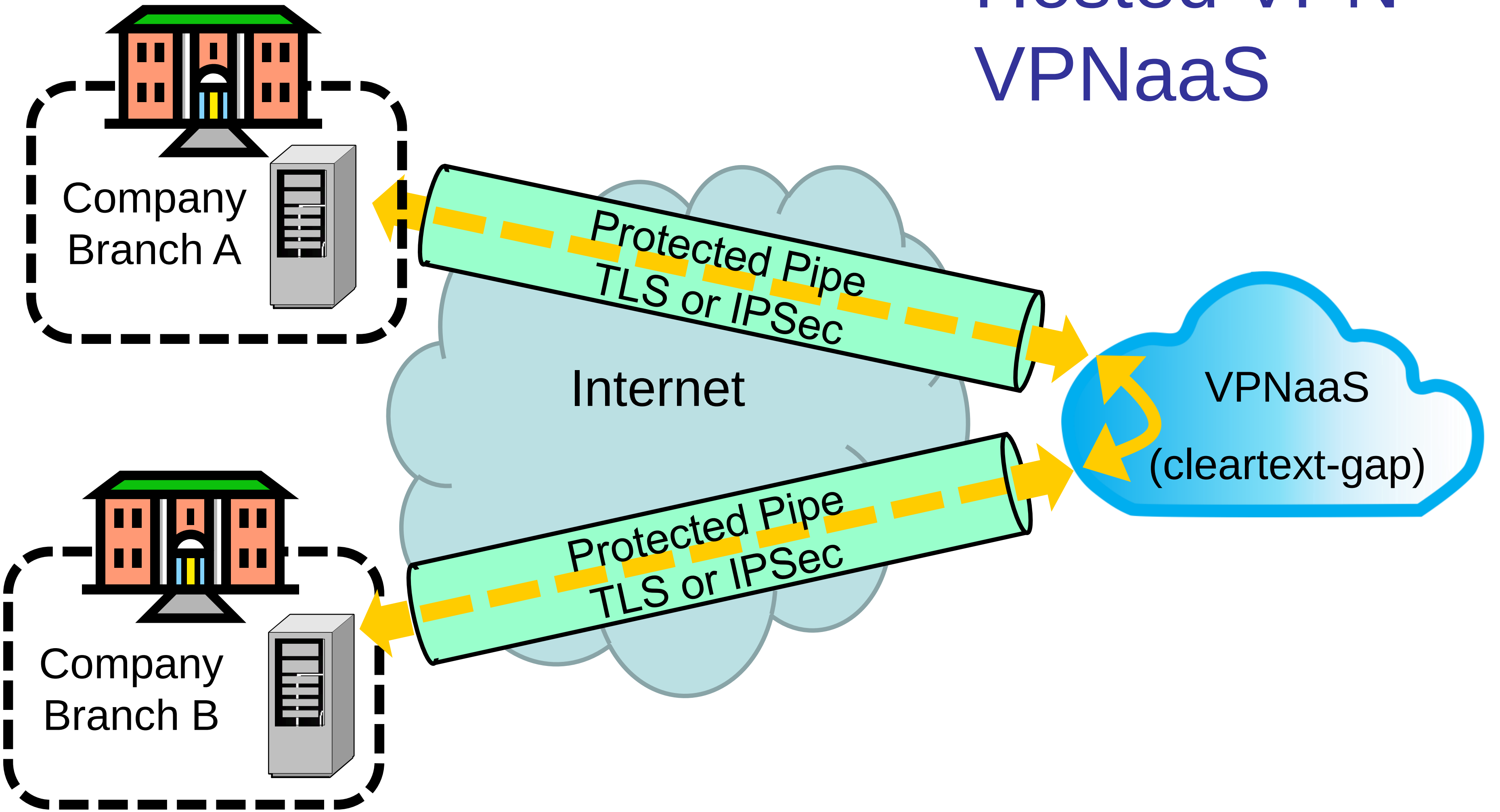


Secure pipe can be attack channel to home network !

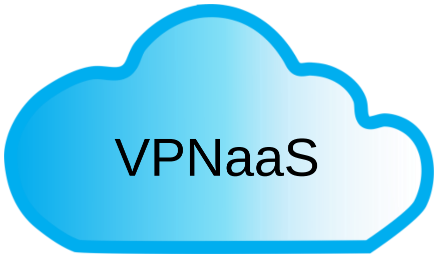
Risks of using IPSec for VPN

- 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

Cloud VPN
Hosted VPN
VPNaaS

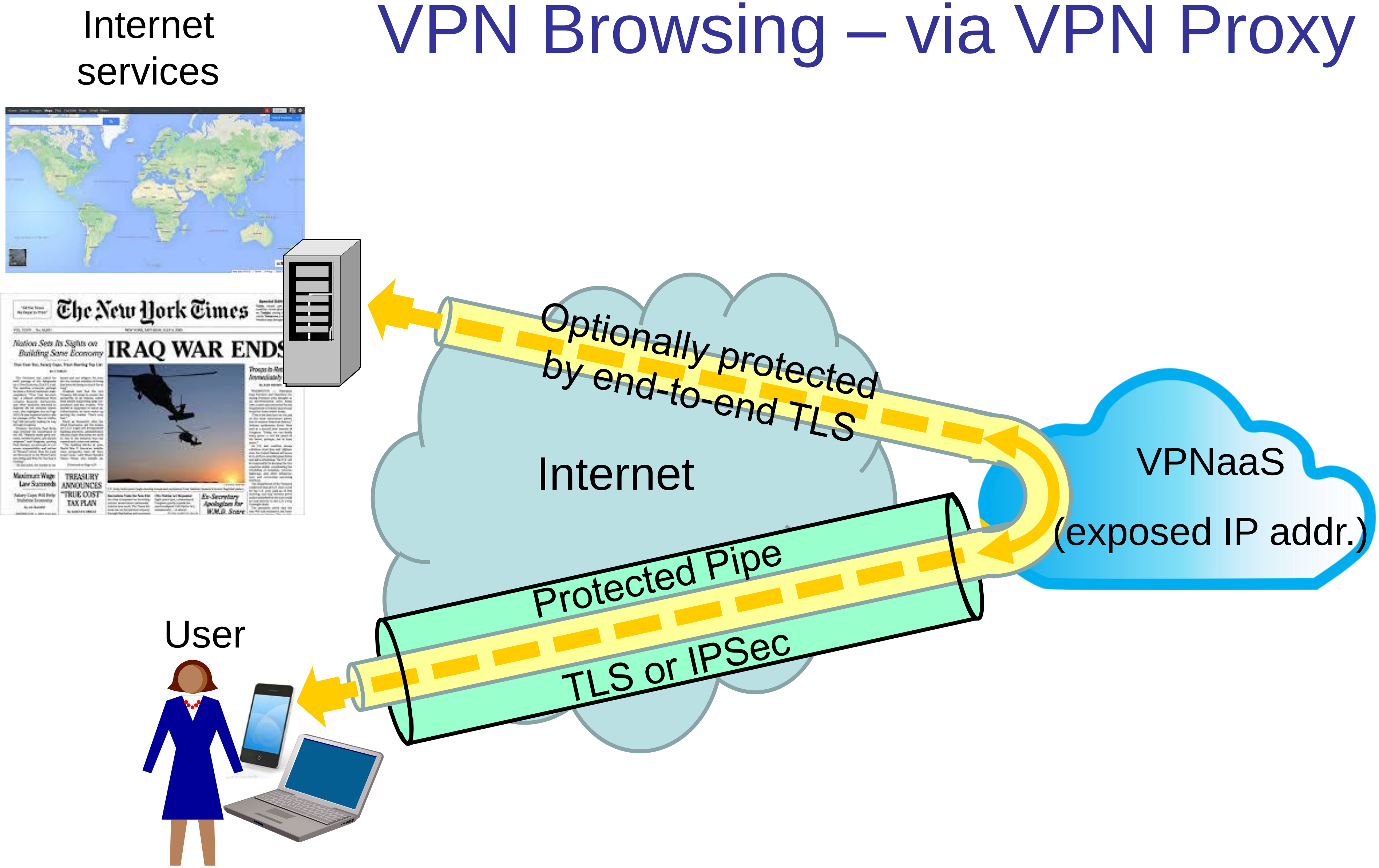


Cloud VPN



- 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?

VPN Browsing – via VPN Proxy



Tor – The Onion Router

Image courtesy indymedia.de

- 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

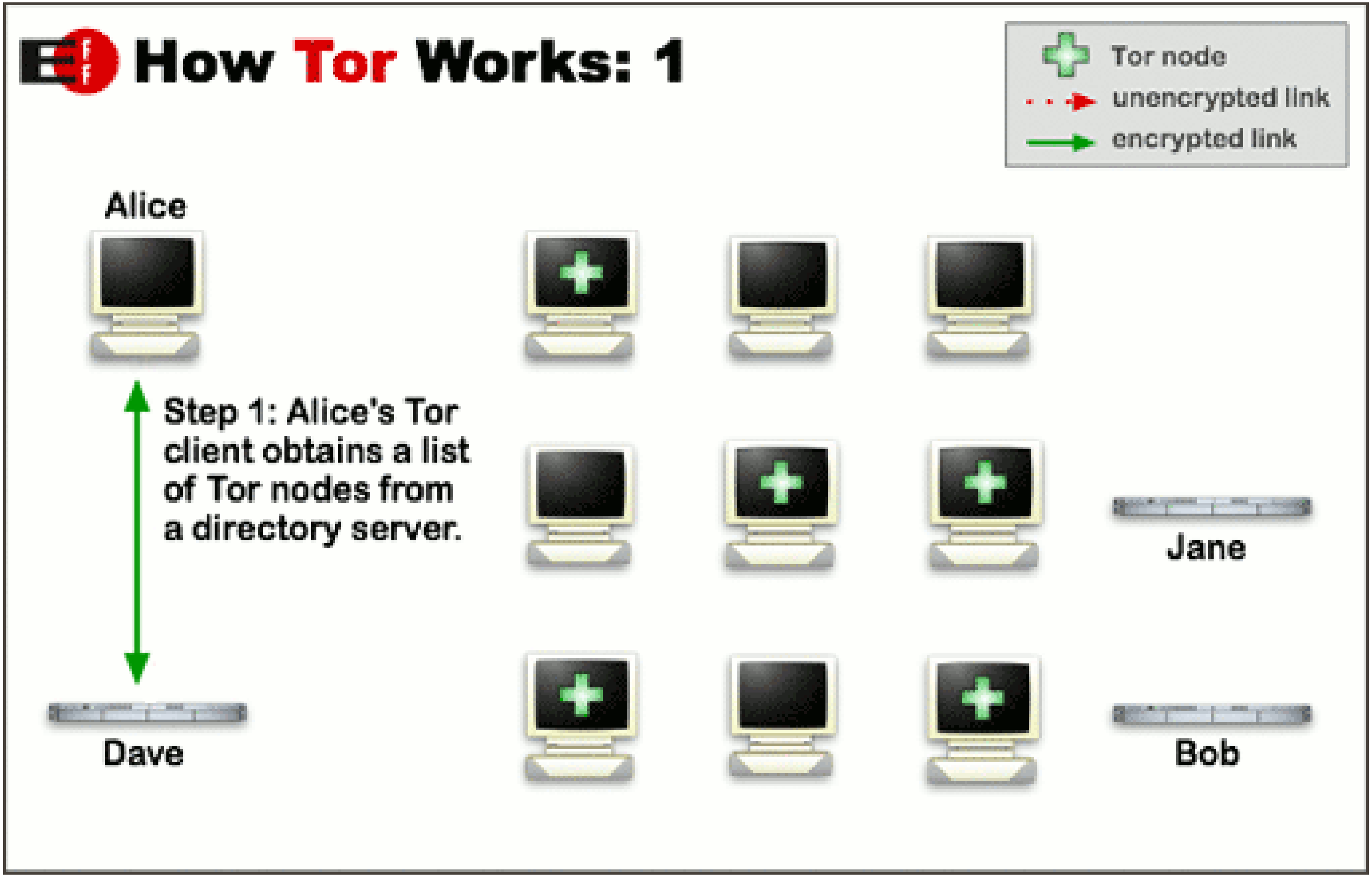


Image courtesy https://www.torproject.org

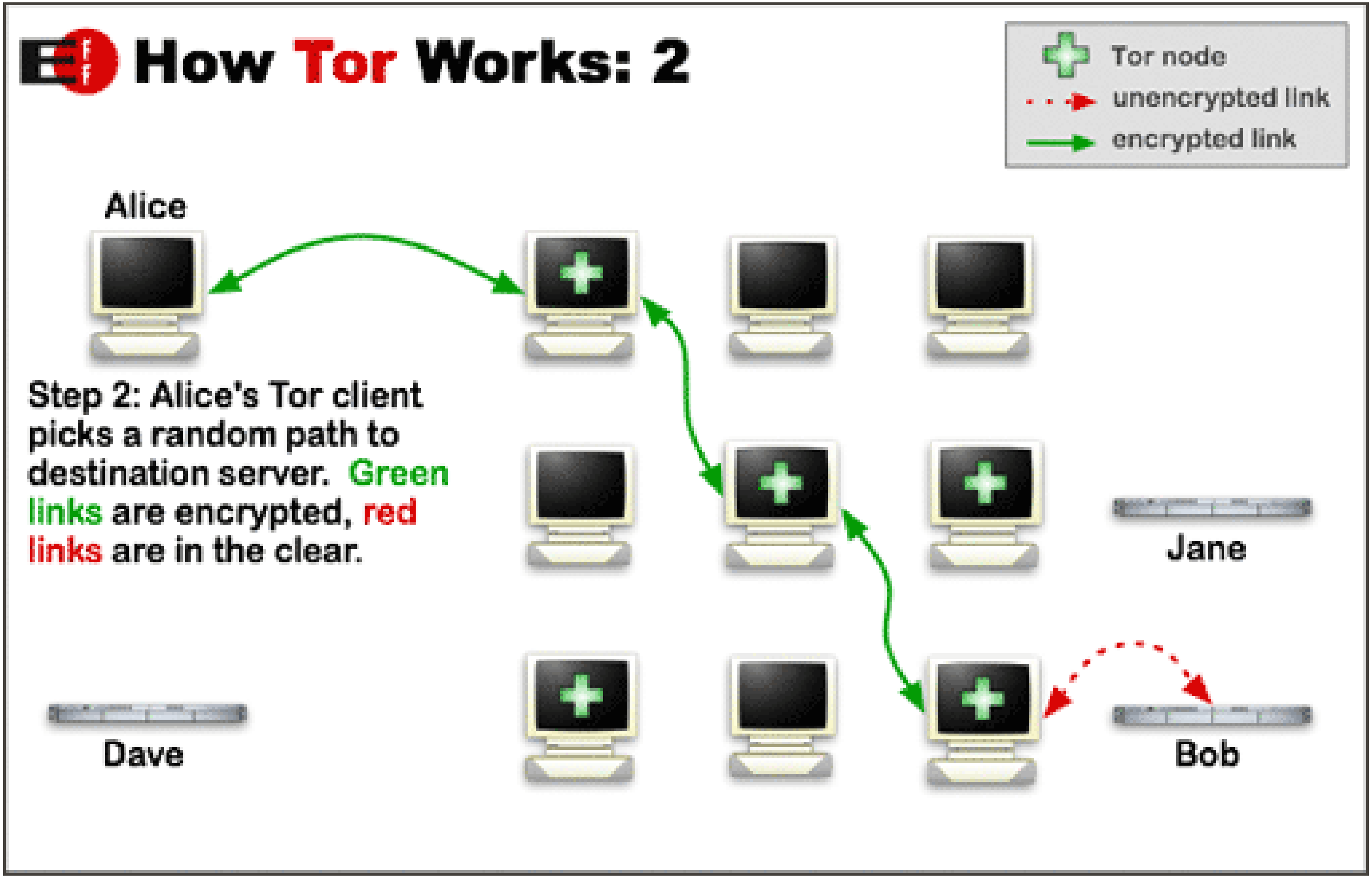


Image courtesy https://www.torproject.org

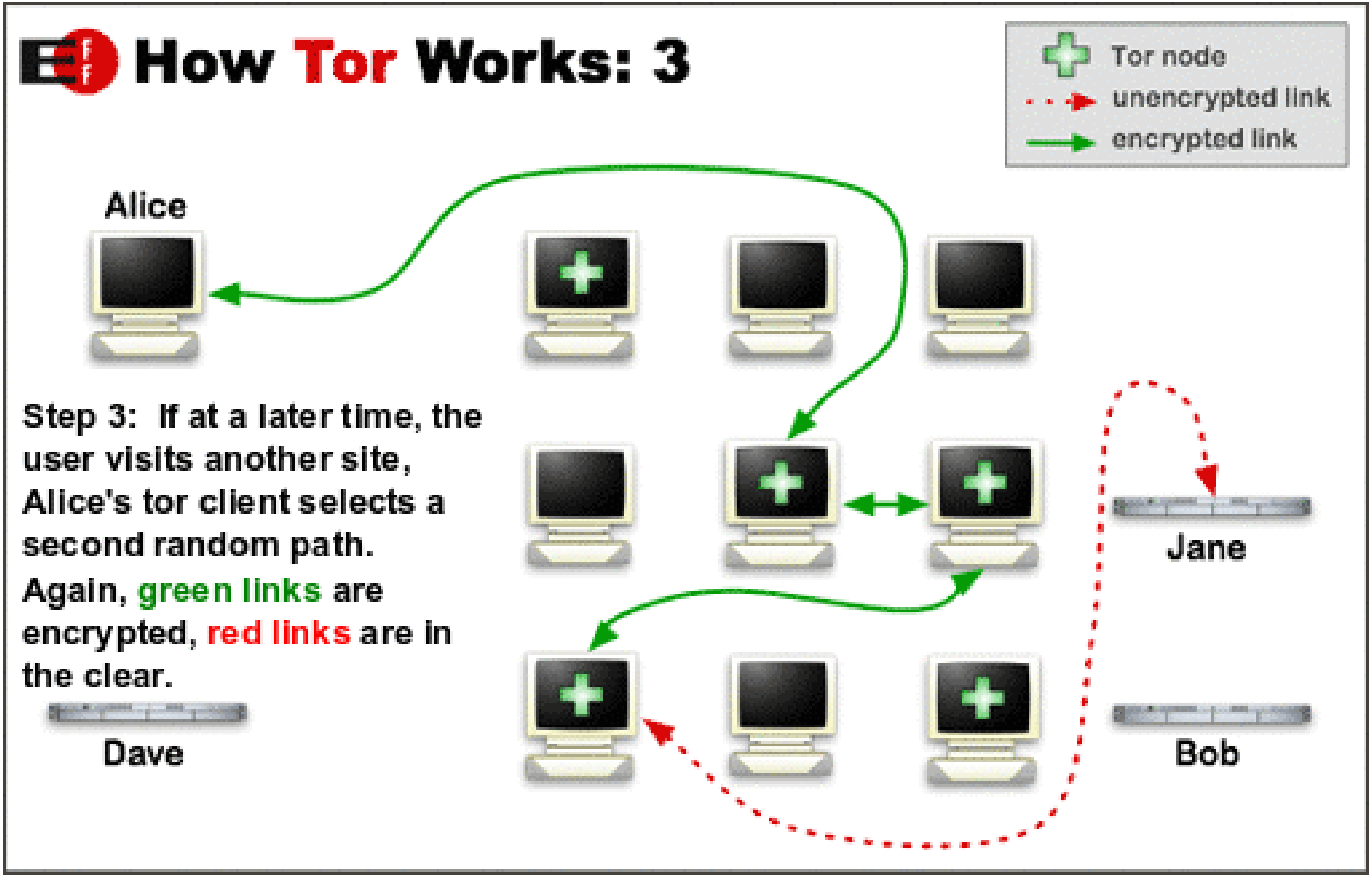
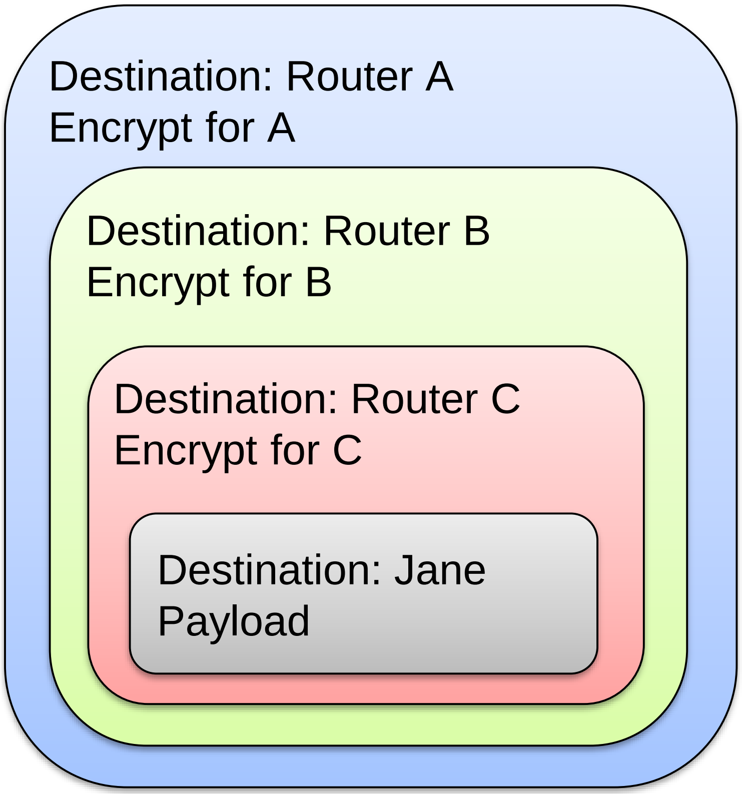
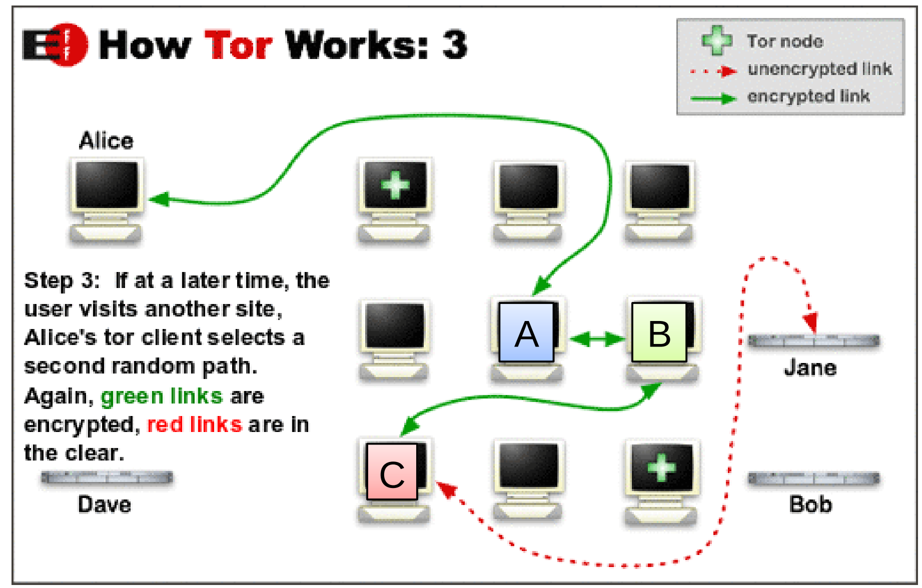
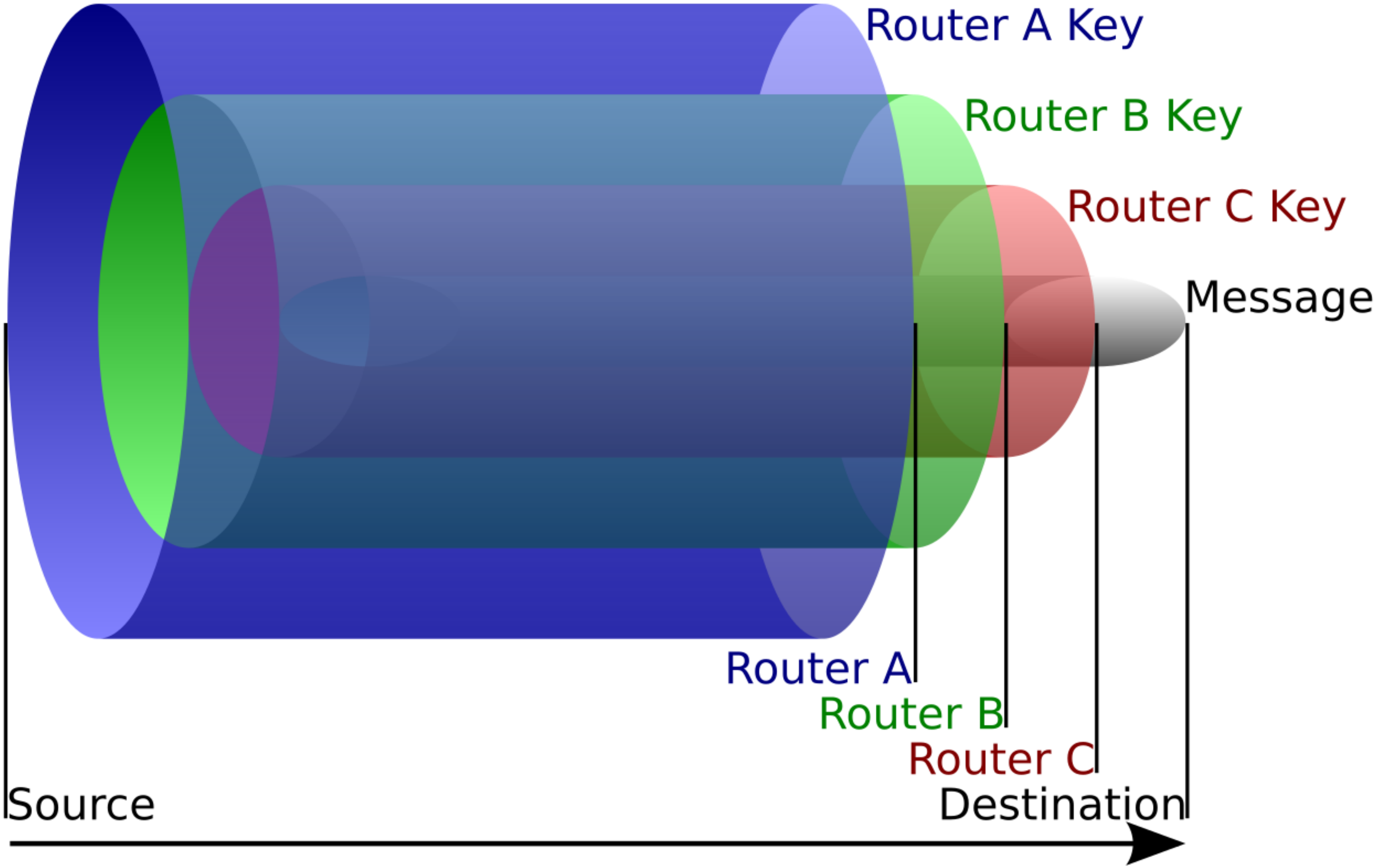


Image courtesy https://www.torproject.org

„Onion“ Message






Router A Key
Router B Key
Router C Key
Message
Router A
Router B
Router C
Source
Destination

End of lecture