CryptoWorks21

Fundamentals of Network Security

3. Network Security Protocols

CryptoWorks21 • July 14, 2021

Dr Douglas Stebila

UNIVERSITY OF WATERLOO

https://www.douglas.stebila.ca/teaching/cryptoworks21

Fundamentals of Network Security

1. Basics of Information Security
   - Security architecture and infrastructure; security goals (confidentiality, integrity, availability, and authenticity); threats/vulnerabilities/attacks; risk management
2. Cryptographic Building Blocks
   - Symmetric crypto: ciphers (stream, block), hash functions, message authentication codes, pseudorandom functions
   - Public key crypto: public key encryption, digital signatures, key agreement
3. **Network Security Protocols & Standards**
   - **Overview of networking and PKI**
   - **Transport Layer Security (TLS) protocol**
   - **Overview: SSH, IPsec, Wireless (Tool: Wireshark)**
4. Offensive and defensive network security
   - Offensive: Pen-tester/attack sequence: reconnaissance; gaining access; maintaining access; denial of service attacks (Tool: nmap)
   - Defensive: Firewalls and intrusion detection
5. Access Control & Authentication; Web Application Security
   - Access control: discretionary/mandatory/role-based; phases
   - Authentication: something you know/have/are/somewhere you are
   - Web security: cookies, SQL injection
   - Supplemental material: Passwords

Network Security Protocols

- Public Key Infrastructure
- Networking
- Transport Layer Security (TLS)
- Other protocols
  – Secure Shell (SSH)
  – IPsec
  – Wireless networking

Assignment 1

1a) Secure email - PGP

- Generate a public key / private key pair
- Send me an encrypted email using PGP

Assignment 0

Downloading and installing VirtualBox and Kali Linux

1b) HTTPS connections

- Inspect X.509 certificates used in a browser
- Use Wireshark to examine the messages in a TLS connection
  - Can do in Kali Linux or in a local installation of Wireshark

1c) Choosing network security protocols

- Discuss the use of different protocols

https://www.douglas.stebila.ca/teaching/cryptoworks21/

Problem: How does Bob get Alice’s public key to begin with?

PUBLIC KEY INFRASTRUCTURES (PKI)

The key establishment problem

- Symmetric ciphers and message authentication codes provide confidentiality and integrity against man-in-the-middle attacks
- But require a shared key between the sender and the receiver

- How to establish a shared key without a secure communication channel?

Key distribution

With asymmetric encryption

- For n parties, each party:
  - Creates their asymmetric key pair
  - Publishes their public key
  - Keeps the private key secret
- For n parties, only n key pairs must be created
- Distribute them **authentically** through out-of-band method

With symmetric encryption

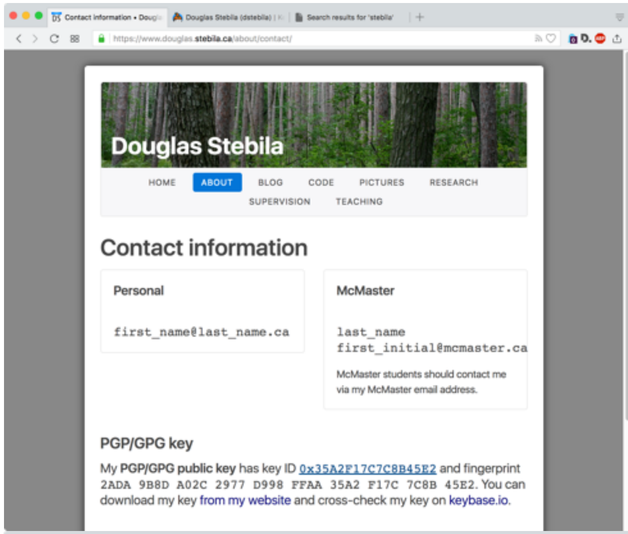
- Someone creates a secret key for each pair of communicating paties
- For n parties, n^2 secret keys must be created
- Distribute them **confidentially** through out-of-band method

Public key distribution problem

- Man-in-the-middle who replaces public keys can then decrypt

- How can we distribute public keys authentically?
  - Especially if we don't have a basis of trust to begin with?

How to distribute public keys?



Public key trust models

User-centric model

- Web of trust

- Each user maintains a **key ring** containing public keys of other users they trust
- Users are completely responsible for deciding which public keys to trust

- Examples:
  – PGP (Pretty Good Privacy)
  – GPG (GNU Privacy Guard - open source version of PGP)

Trusted authority model

- Public key infrastructure

- Trusted authorities perform checks and issue **certificates** endorsing public keys
- User trusts all certificates issued by an authority

- Examples:
  – PKI in web browsers

Certificates and certificate authorities

- Relies on trusted authorities (called **certificate authorities**) to vouch that public keys belong to certain subjects

- A **certificate** is an assertion by a trusted third party that a particular public key belongs to a particular entity.

- A **digital certificate** contains
  - The subject's identity
  - The subject's public key
  - Additional information (e.g., validity period)
  - The issuer's digital signature

Certificates and certificate authorities

The **certificate authority** generates a certificate by

1. Obtaining the subject's public key by some trusted mechanism.
2. Verifying that the subject really is who she says she is.
3. Signing (using the certificate authority's private key) the subject's public key and name.

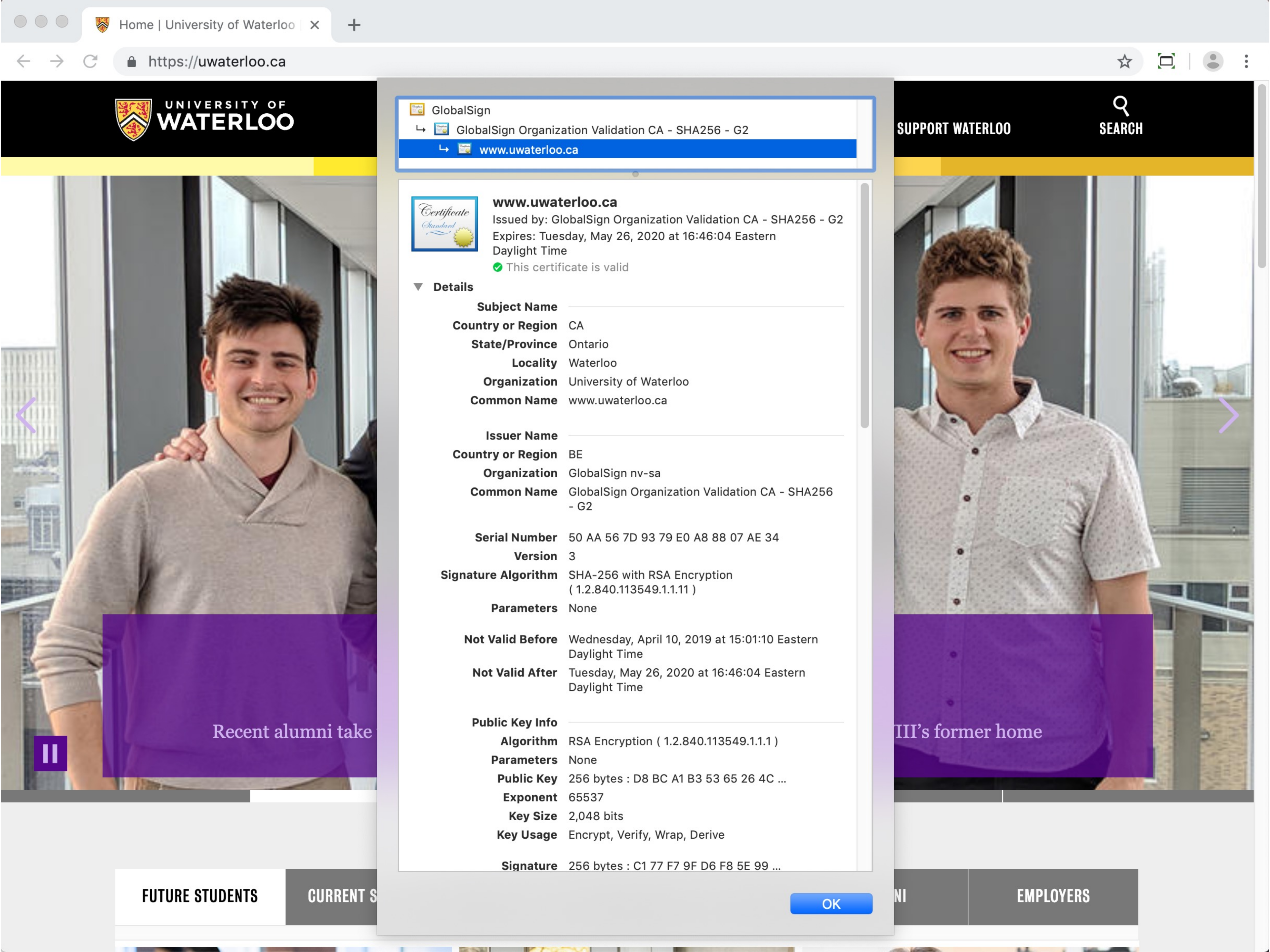
This allows two parties who have never met to establish trust between them:

- Exchange certificates.
- Do authentication using digital signatures.
- If they each trust the certificate authority that signed the other party's certificate, they can now be certain who the other party is.

X.509 certificates

- X.509 is a standard format for digital certificates
- Current version: v3
- Standardized by International Telecommunication Union (ITU-T)

- Important fields in X.509 digital certificates are:
  - Version number
  - Serial Number (set by the CA)
  - Signature Algorithm identifier (Algorithm used for dig sigs)
  - Issuer (Name of the CA)
  - Subject (Name of entity to which certificate has been issued)
  - Subject Public Key Information
  - Validity period (certificate should not be used outside this time)
  - Digital signature (of the certificate, signed by the CA)


Home | University of Waterloo
https://uwaterloo.ca
UNIVERSITY OF WATERLOO
SUPPORT WATERLOO
SEARCH
GlobalSign
GlobalSign Organization Validation CA - SHA256 - G2
www.uwaterloo.ca
Certificate Standard
www.uwaterloo.ca
Issued by: GlobalSign Organization Validation CA - SHA256 - G2
Expires: Tuesday, May 26, 2020 at 16:46:04 Eastern Daylight Time
This certificate is valid
Details
Subject Name
Country or Region CA
State/Province Ontario
Locality Waterloo
Organization University of Waterloo
Common Name www.uwaterloo.ca
Issuer Name
Country or Region BE
Organization GlobalSign nv-sa
Common Name GlobalSign Organization Validation CA - SHA256 - G2
Serial Number 50 AA 56 7D 93 79 E0 A8 88 07 AE 34
Version 3
Signature Algorithm SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters None
Not Valid Before Wednesday, April 10, 2019 at 15:01:10 Eastern Daylight Time
Not Valid After Tuesday, May 26, 2020 at 16:46:04 Eastern Daylight Time
Public Key Info
Algorithm RSA Encryption (1.2.840.113549.1.1.1)
Parameters None
Public Key 256 bytes : D8 BC A1 B3 53 65 26 4C ...
Exponent 65537
Key Size 2,048 bits
Key Usage Encrypt, Verify, Wrap, Derive
Signature 256 bytes : C1 77 F7 9F D6 F8 5E 99 ...
OK
Recent alumni take
III's former home
FUTURE STUDENTS
EMPLOYERS

X.509 certificates

A standardized format for certificates.

Uses a strange (old) format called ASN.1 and a strange binary encoding.

Domain name

Certificate authority

Validity period

Public key

Revocation information

CA's signature on everything

www.uwaterloo.ca
Issued by: GlobalSign Organization Validation CA - SHA256 - G2
Expires: Monday, March 26, 2018 at 16:46:04 Eastern Daylight Time
This certificate is valid

Details

Subject Name	
Country	CA
State/Province	Ontario
Locality	Waterloo
Organization	University of Waterloo
Common Name	www.uwaterloo.ca
Issuer Name	
Country	BE
Organization	GlobalSign nv-sa
Common Name	GlobalSign Organization Validation CA - SHA256 - G2
Serial Number	2E 4B 75 8D 35 75 9F A0 27 1F F1 BC
Version	3
Signature Algorithm	SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters	none
Not Valid Before	Thursday, December 1, 2016 at 16:26:05 Eastern Standard Time
Not Valid After	Monday, March 26, 2018 at 16:46:04 Eastern Daylight Time
Public Key Info	
Algorithm	RSA Encryption (1.2.840.113549.1.1.1)
Parameters	none
Public Key	256 bytes : D8 BC A1 B3 53 65 26 4C 39 D5 92 56 84 66 37 CC 1F 57 FD 5B 87 A7 38 36 D5 05 83 3E 6C 96 02 12 3E 6A C3 CF F3 BC 3C 6D BF FE BB BB 08 02 8C 97 AF D9 86 2A 6B F6 EE D7 0C DA E8 2F DA B1 14 E8 B5 EA 04 FF 12 3D BA ED 42 FA CE A7 93 AC 15 29 66 63 2E 39 7F F2 69 D7 82 01 CB B8 92 81 75 B3 F9 4A 87 32 05 67 E0 42 78 55 1F 03 17 A8 8F 6E 85 56 1B 1C AF 8E 35 A5 14 91 E9 25 61 AC 05 6F 9A FC 58 F8 7F 64 BE C7 D4 6A EB 2A BC 47 D6 30 35 51 1D BD 57 09 49 19 9E BC 43 09 F1 58 0C 88 E5 D1 9C CB 00 AA A8 66 E8 4B C9 CE AA 63 63 5A A9 AF 3D 63 90 E8 7A 2F 95 1B CC EC 2E 48 16 4A 0E B8 1F 69 45 82 3C F1 09 53 2C B6 69 8C 70 4C 99 89 6F 4E CA 0C 8D F5 1E 3A 5F 07 46 7D 63 ED 3D 38 B7 0E 88 ED 4F FD 00 C2 76 35 F7 99 5B 39 CE 26 CC C4 19 CA 47 DA 6D 80 61 7E 01 8E 96 DD
Exponent	65537
Key Size	2048 bits
Key Usage	Encrypt, Verify, Wrap, Derive
Signature	256 bytes : 5B 01 1C 81 17 01 07 2F ...

Extension	Key Usage (2.5.29.15)
Critical	YES
Usage	Digital Signature, Key Encipherment
Extension	Basic Constraints (2.5.29.19)
Critical	NO
Certificate Authority	NO
Extension	Extended Key Usage (2.5.29.37)
Critical	NO
Purpose #1	Server Authentication (1.3.6.1.5.5.7.3.1)
Purpose #2	Client Authentication (1.3.6.1.5.5.7.3.2)
Extension	Subject Key Identifier (2.5.29.14)
Critical	NO
Key ID	DB B9 21 BC CD 3E AF 70 C9 E9 3D 9B FF 42 B0 C8 88 8F 78 C3
Extension	Authority Key Identifier (2.5.29.35)
Critical	NO
Key ID	96 DE 61 F1 BD 1C 16 29 53 1C C0 CC 7D 3B 83 00 40 E6 1A 7C
Extension	Subject Alternative Name (2.5.29.17)
Critical	NO
DNS Name	www.uwaterloo.ca
DNS Name	uwaterloo.ca
Extension	Certificate Policies (2.5.29.32)
Critical	NO
Policy ID #1	(1.3.6.1.4.1.4146.1.20)
Qualifier ID #1	Certification Practice Statement (1.3.6.1.5.5.7.2.1)
CPS URI	https://www.globalsign.com/repository/
Policy ID #2	(2.23.140.1.2.2)
Extension	CRL Distribution Points (2.5.29.31)
Critical	NO
URI	http://crl.globalsign.com/gs/gsorganizationvalsha2g2.crl
Extension	Certificate Authority Information Access (1.3.6.1.5.5.7.1.1)
Critical	NO
Method #1	CA Issuers (1.3.6.1.5.5.7.48.2)
URI	http://secure.globalsign.com/cacert/gsorganizationvalsha2g2r1.crt
Method #2	Online Certificate Status Protocol (1.3.6.1.5.5.7.48.1)
URI	http://ocsp2.globalsign.com/gsorganizationvalsha2g2
Fingerprints	
SHA-256	C7 DC B4 CD 45 9E D5 1A AA 03 86 73 31 4B F8 A9 53 A6 9C F1 B9 C4 35 A3 AD C6 4F 87 97 93 AD D6
SHA-1	53 9E 06 03 64 F9 24 F6 ED 9B A1 0E F9 46 81 1A E0 88 F8 A5

Certificate revocation

- Once a certificate's been issued, what happens if the user's private key has been compromised?

- We would like to be able to **revoke** the certificate, or indicate that it should no longer be trusted.

Certificate revocation mechanisms

Certificate Revocation Lists (CRLs)

- Each CA can publish a file containing a list of certificates that have been revoked.
- Have to download whole list.
- CRL address often included in certificate.

Online Certificate Status Protocol (OCSP)

- An online service run by a CA for checking in real-time if a certificate has been revoked.
- Don’t have to download whole list.
- Not widely implemented.
- Compromises user privacy

Public key infrastructure

A **public key infrastructure (PKI)** is

- a set of systems (hardware, software, policies, procedures)
- for managing (creating, distributing, storing, revoking)
- digital certificates.

Includes:

- one or more certificate authorities
- users
- relying parties
- possibly a timestamp server
- possibly a directory server storing certificates (e.g., LDAP server, Active Directory server)

Public key infrastructure

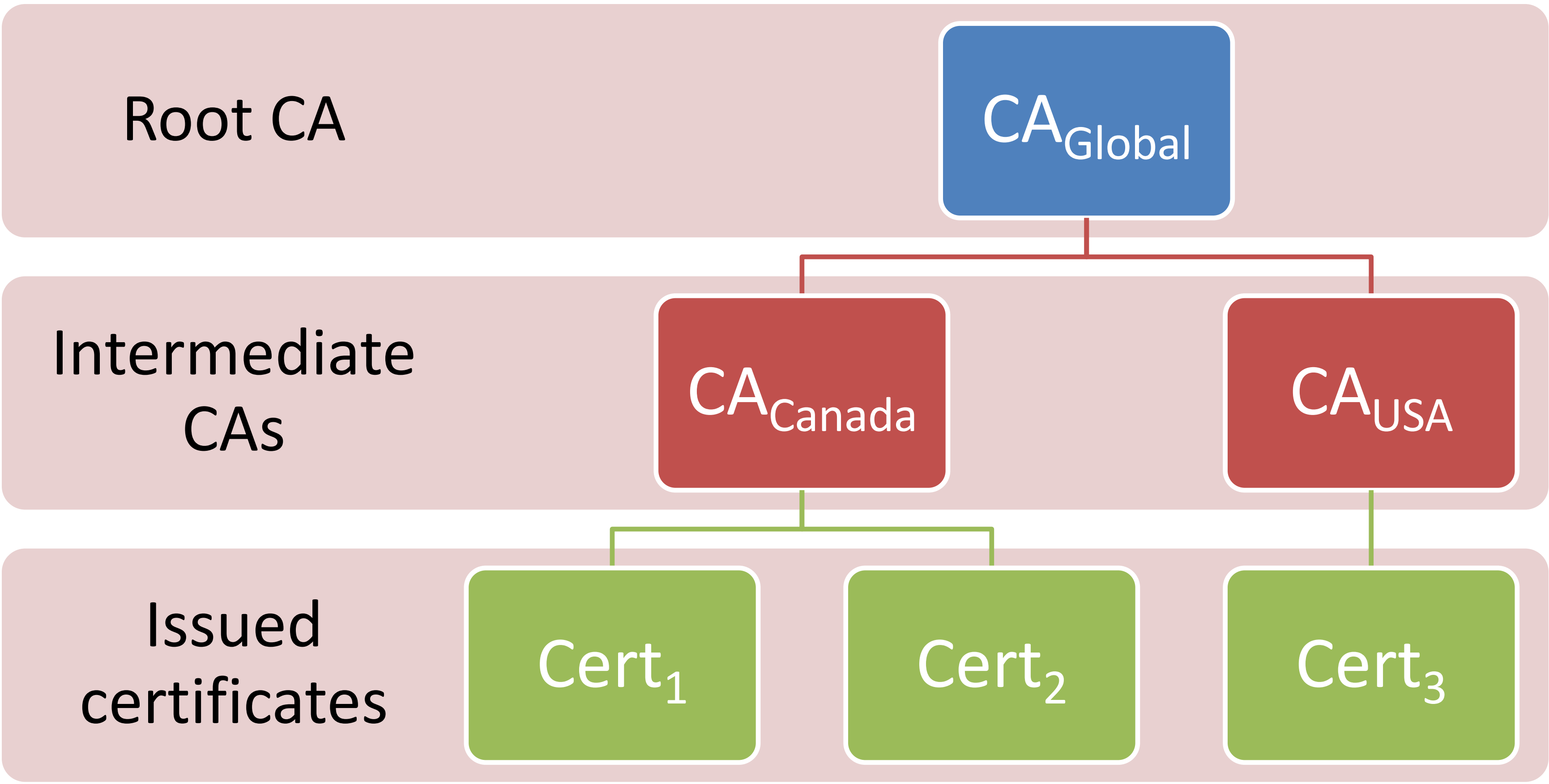
A **public key infrastructure (PKI)** is

- a set of systems (hardware, software, policies, procedures)
- for managing (creating, distributing, storing, revoking)
- digital certificates.

Includes:

- one or more certificate authorities
- subjects
- users
- relying parties
- possibly a timestamp server
- possibly a directory server storing certificates (e.g., LDAP server, Active Directory server)

Hierarchical CAs



Using certificates for confidentiality

- Suppose Alice wants to send a message confidentially to Bob
  1. Alice needs Bob's public key
     1. Alice obtains $Cert_{Bob}$, signed by CA_1
     2. Alice checks that the identity in $Cert_{Bob}$ is the Bob she wants
     3. Alice verifies CA_1's signature on $Cert_{Bob}$ using CA_1's public key
     4. Alice extracts pk_{Bob} from $Cert_{Bob}$
  2. Alice uses pk_{Bob} to encrypt message M for Bob

- Does this provide confidentiality – can only Bob read the message?
  – If Alice trusts the CA that issued $Cert_{Bob}$ to
    - Check the identity of subjects before issuing certificates
    - Not issue fraudulent certificates
  – And Alice is certain of the CA's public key
  – Then Alice can be sure that only Bob will be able to decrypt the message

Using certificates for authentication/integrity

- Suppose Alice wants to check if a message really came from Bob
  1. Alice needs Bob's public key
     1. Alice obtains $Cert_{Bob}$, signed by CA_1
     2. Alice checks that the identity in $Cert_{Bob}$ is the Bob she wants
     3. Alice verifies CA_1's signature on $Cert_{Bob}$ using CA_1's public key
     4. Alice extracts pk_{Bob} from $Cert_{Bob}$
  2. Alice uses pk_{Bob} to verify the signature on a given message supposedly from Bob

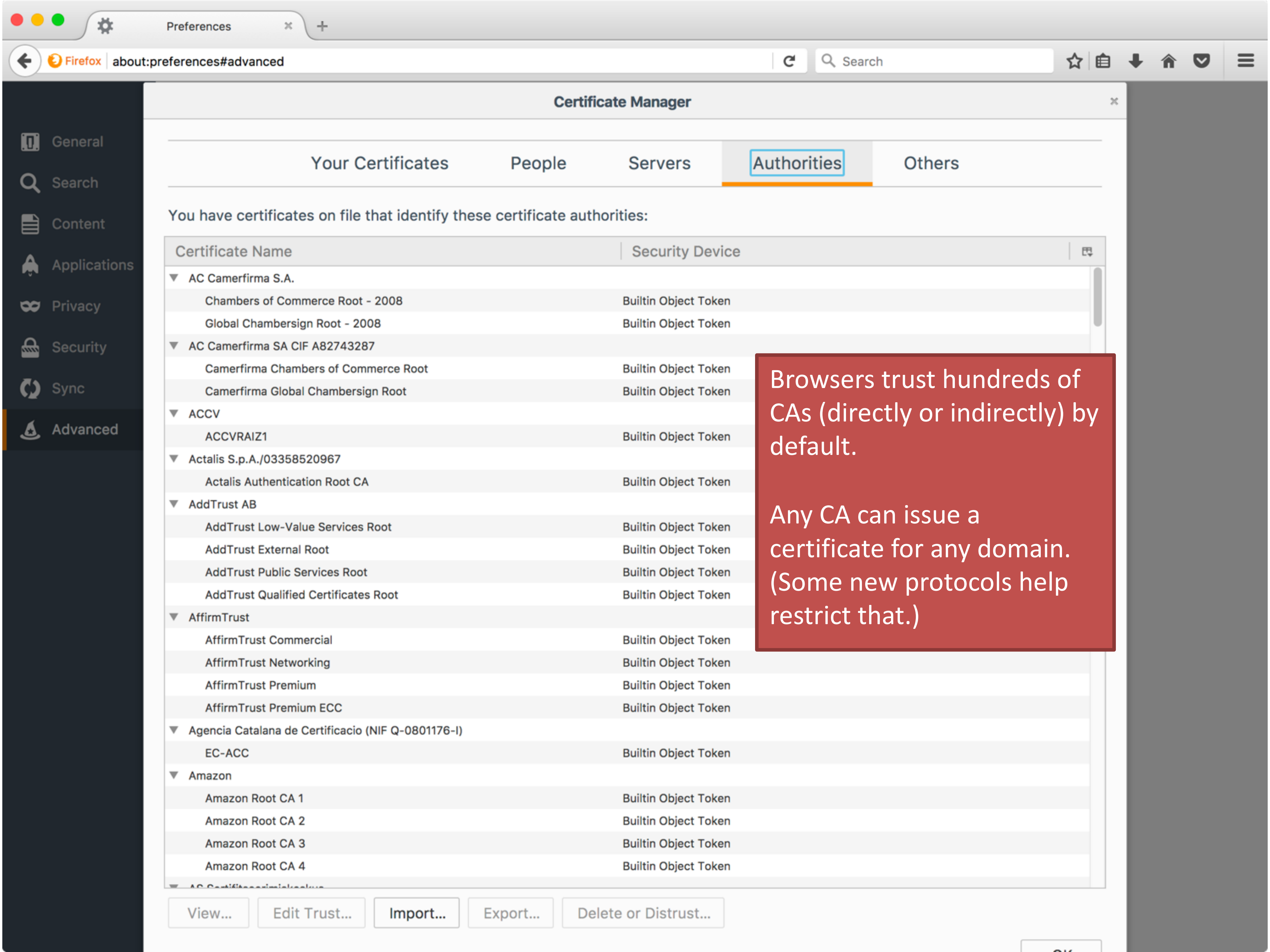
- Does this provide integrity– can only Bob send messages?
  - If Alice trusts the CA that issued $Cert_{Bob}$ to
    - Check the identity of subjects before issuing certificates
    - Not issue fraudulent certificates
  - And Alice is certain of the CA's public key
  - Then Alice can be sure that only Bob will be able to sign messages that verify

Trustworthiness of CAs

- We assume that CAs
  - Check the identity of subjects before issuing certificates
  - Don't issue fraudulent certificates
  - Protect their own signing key

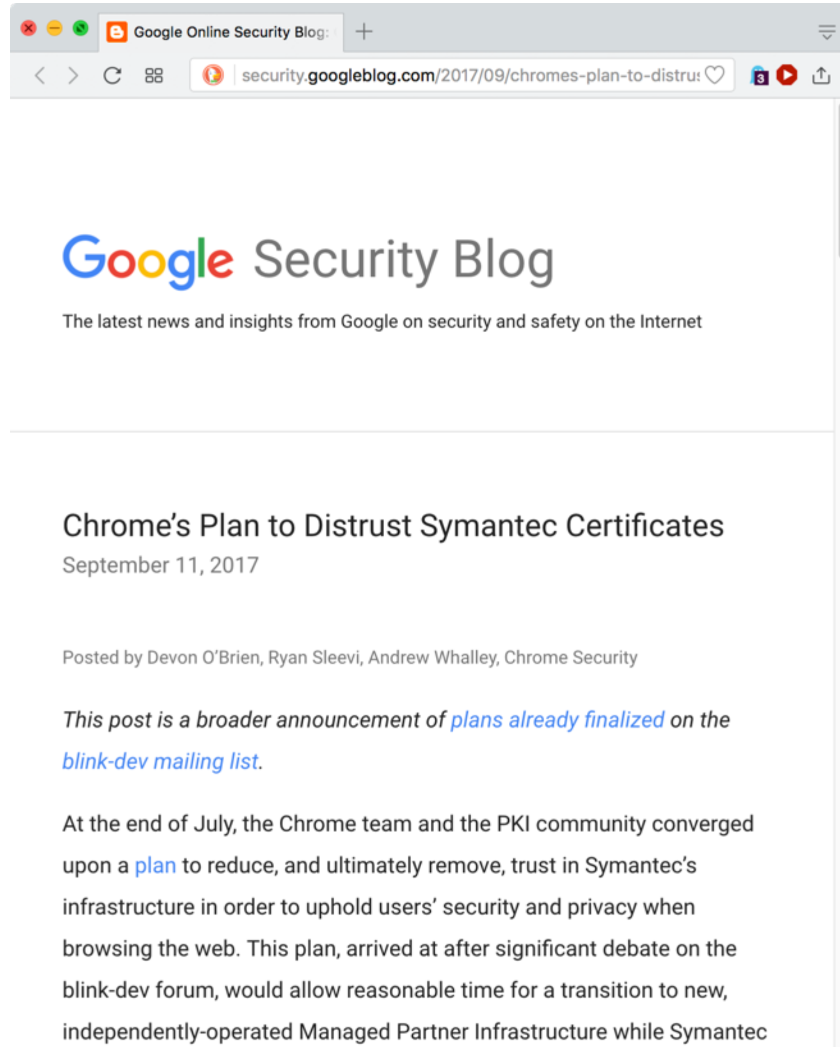
Applications of PKI

- Web site authentication (TLS)
- Email authentication (S/MIME, PGP)
- Domain names (DNSSEC)
- Digital identities
  - e.g., national identity cards (Belgium, Spain, Germany)
- Business-to-business e-commerce
  - e.g., digitally signing transactions, XML signatures


Preferences
Firefox about:preferences#advanced
Search
General
Search
Content
Applications
Privacy
Security
Sync
Advanced
Certificate Manager
Your Certificates
People
Servers
Authorities
Others
You have certificates on file that identify these certificate authorities:
Certificate Name
Security Device
AC Camerfirma S.A.
Chambers of Commerce Root - 2008 Builtin Object Token
Global Chambersign Root - 2008 Builtin Object Token
AC Camerfirma SA CIF A82743287
Camerfirma Chambers of Commerce Root Builtin Object Token
Camerfirma Global Chambersign Root Builtin Object Token
ACCV
ACCVRAIZ1 Builtin Object Token
Actalis S.p.A./03358520967
Actalis Authentication Root CA Builtin Object Token
AddTrust AB
AddTrust Low-Value Services Root Builtin Object Token
AddTrust External Root Builtin Object Token
AddTrust Public Services Root Builtin Object Token
AddTrust Qualified Certificates Root Builtin Object Token
AffirmTrust
AffirmTrust Commercial Builtin Object Token
AffirmTrust Networking Builtin Object Token
AffirmTrust Premium Builtin Object Token
AffirmTrust Premium ECC Builtin Object Token
Agencia Catalana de Certificacio (NIF Q-0801176-I)
EC-ACC Builtin Object Token
Amazon
Amazon Root CA 1 Builtin Object Token
Amazon Root CA 2 Builtin Object Token
Amazon Root CA 3 Builtin Object Token
Amazon Root CA 4 Builtin Object Token
View...
Edit Trust...
Import...
Export...
Delete or Distrust...
Browsers trust hundreds of CAs (directly or indirectly) by default.
Any CA can issue a certificate for any domain. (Some new protocols help restrict that.)

CA/Browser Forum

- Voluntary consortium of CAs and browser vendors
- Issue guidelines for CA management and procedures
  - Effectively requirements for CAs to have their certificates installed in browsers



https://security.googleblog.com/2017/09/chromes-plan-to-distrust-symantec.html

Secure email

- X.509 certificates can also be used to send secure email:
  - digitally signed
  - encrypted
- **S/MIME** (Secure/Multipurpose Internet Mail Extensions):
  - Supported in most desktop mail programs.
  - Relies on a public key infrastructure.
- **PGP** (Pretty Good Privacy):
  - Available as an add-on to most desktop mail programs.
  - Uses public keys, but doesn't require CAs: users manually distribute their keys in a "web of trust"
- Not widely used:
  - Users must know how set up public keys and obtain S/MIME X.509 certificate or distribute PGP public keys.
  - Little to no support in webmail.

NETWORKING

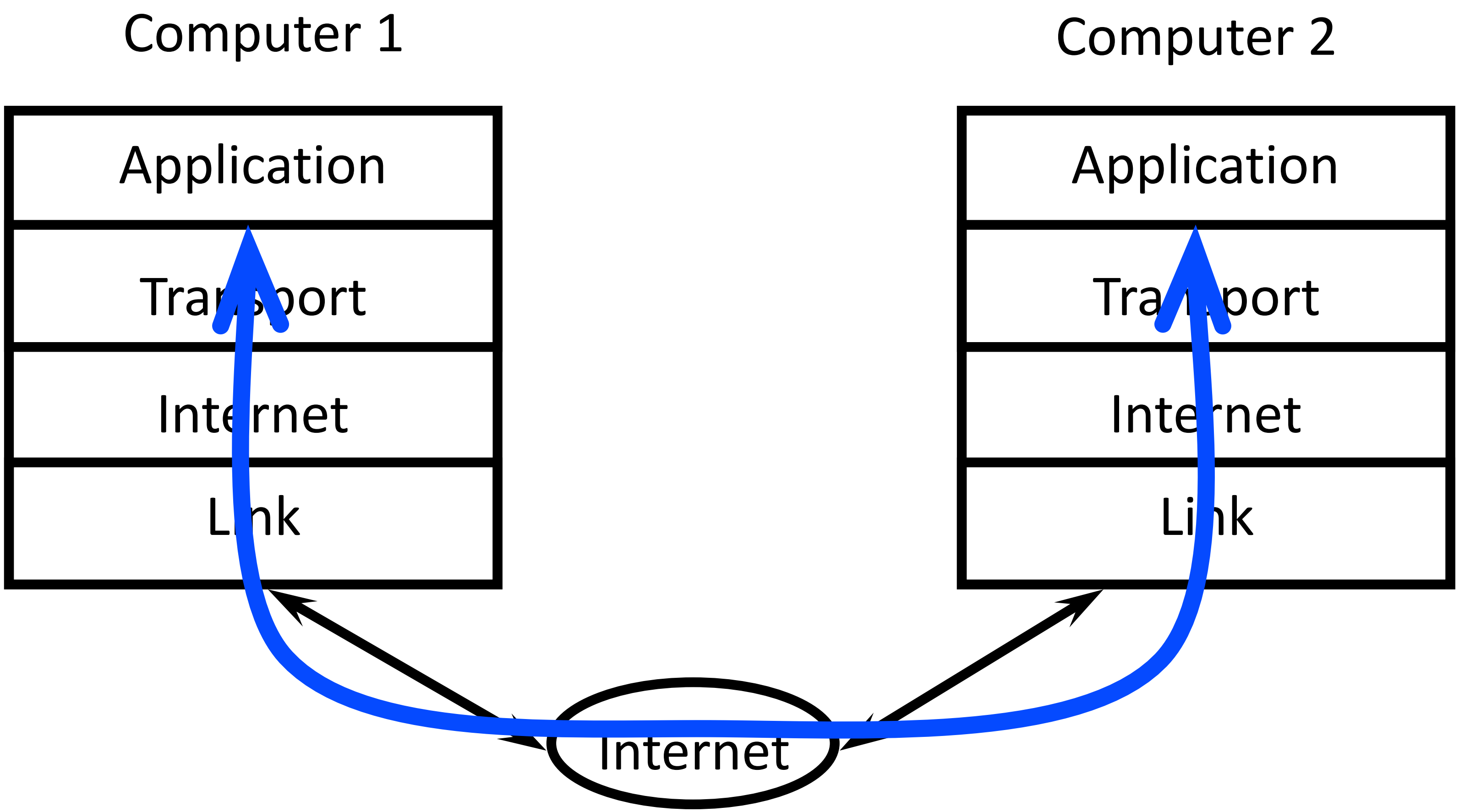
IETF Internet Protocol suite

Layer	Examples
Application	web (HTTP, HTTPS) email (SMTP, POP3, IMAP) login (SSH, Telnet)
Transport	connection-oriented (TCP) connectionless (UDP)
Internet	addressing and routing: • IPv4, IPv6 control (ICMP) security (IPsec)
Link	packet framing (Ethernet) physical connection • WLAN (WEP, WPA) • ADSL • GSM/3G

Most defined by the Internet Engineering Task Force (IETF)

There's also the 7-layer OSI model.

Internet Communication - Basics



Link (a.k.a. network access) layer

The **link** or **network access layer** is the physical layer and is associated with computer hardware.

Goal:
- provide addressing and delivery in a local network

Addressing:
- physical addresses identify network nodes
  - Ethernet MAC address

Computer networks can use a large number of connections and transmission media
- Telephone wires
- Ethernet (twisted pair) cables
- Optic Fibre cables
- Satellite communications
- Mobile phone networks
- Wireless networks
- Bluetooth

Internet (a.k.a. network) layer

The **Internet layer** runs a low level protocol called the Internet Protocol (IP) (plus a few extra helpers, e.g. ICMP).

- IPv4 (1981), IPv6 (1996)

Goal:

- provide global
  addressing and delivery

Internet (a.k.a. network) layer

Addressing

- Each host has a unique IP address:
  - IPv4, 32 bit,
    e.g., 131.181.118.220
  - IPv6, 128 bit,
    e.g., 2001:0db8:85a3:0000:
    0000:8a2e:0370:7334

Packet routing

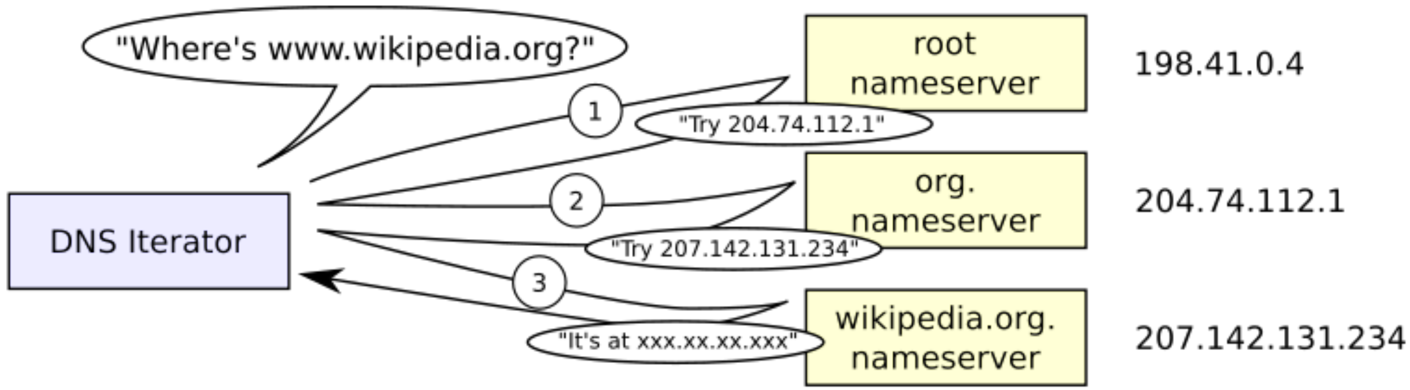
- Organizations are assigned a range of IP addresses that they manage and assign to their computers.

Internet layer addressing: Domain Name System (DNS)

- Hierarchical directory service for domain names
- Main feature: translates domain names into IP addresses

- A domain name record can provide a variety of additional information
  - Authorized name servers
  - Mail server addresses
  - Anti-spam information
  - Public keys

Hierarchical domain name resolution



To find the IP address for domain www.wikipedia.org:

- Browser asks its local DNS server
- If local DNS server has the answer cached, it returns it.
- Else, local DNS server asks a root nameserver (10 global)

1. Root nameserver looks up nameserver for .org and redirects to it
2. Nameserver for .org looks up nameserver for wikipedia.org and redirects to it
3. Nameserver for wikipedia.org responds with IP address for www.wikipedia.org

https://en.wikipedia.org/wiki/File:Example_of_an_iterative_DNS_resolver.svg

Transport Layer

The **transport layer** establishes basic data channels for applications.

Goal:

- Establish channels for applications between hosts

Addressing:

- Ports identify different applications on same computer
  - 16-bit number

- Two main protocols:
  - TCP: Transmission Control Protocol
  - UDP: User Datagram Protocol

Transport Layer

TCP (Transmission Control Protocol)

- **connection-oriented** protocol
  - back-and-forth, ongoing connections
- **reliable**
  - long messages split into packets
  - in-order delivery of packets, recombined to long message
  - error checking
  - retransmission of lost packets
  - congestion control

UDP (User Datagram Protocol)

- **connectionless** protocol
  - send a packet, that’s it
- **unreliable**
  - simple error checking
  - no retransmission of lost packets
  - used for streaming
    - audio, video, VOIP

Application Layer

Application layer protocols are used by applications to provide user services over a network.

Each application protocol has unique message formats that are sent and received to achieve their tasks.

- HTTP (web)
- FTP (file transfer)
- SSH, Telnet (login)
- SMTP, POP3, IMAP (email)
- XMPP (chat)
- BitTorrent (I'm sure you know what this is used for)

Each application protocol requires the lower network layers (TCP, IP, Network Access) to communicate on the network.

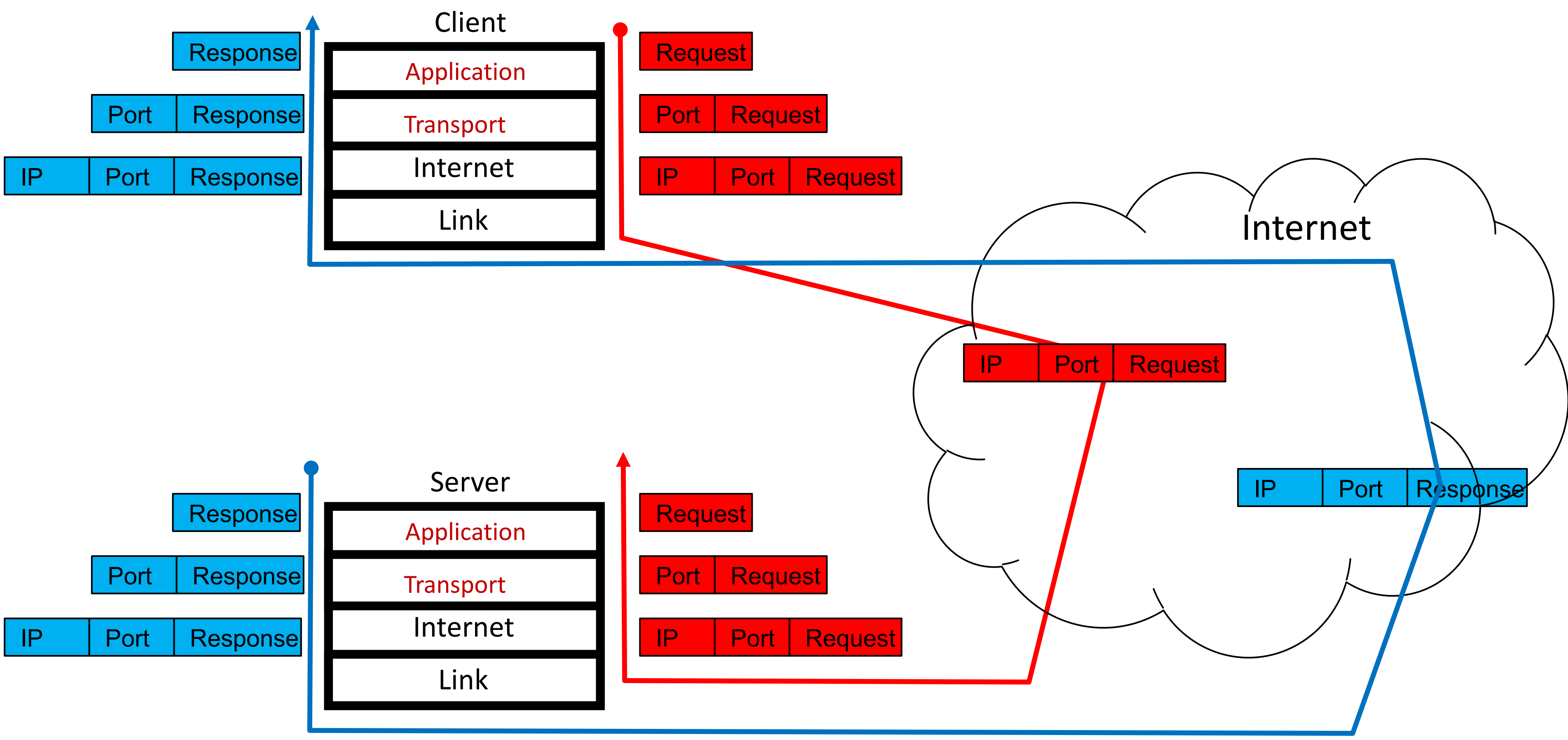
Many use an intermediate protocol called SSL/TLS for encryption and authentication.

Client-server on the Internet

- Each **application server** listens for messages on a particular port number. Common ports:
  - web servers: port 80 (HTTP), 443 (HTTPS)
  - login: port 22 (SSH), 23 (Telnet)
  - file transfer: port 20/21 (FTP), 22 (SFTP/SCP)
  - email servers: port 25 (SMTP), 220/993 (IMAP), 110 (POP)

- Clients identify the machine they want to connect to using an IP address.
- Clients identify the program they want to use using a port number.

Client-server communication



Example: requesting a webpage

Application Layer – Web browser

- Constructs the request in a specific format – HTTP request.
- Includes address information of the server (domain name)

Transport Layer

- Breaks HTTP request into TCP packets (each with address info – domain name and port)

Internet Layer

- Looks up IP address for domain name
- Routes TCP packets to destination IP address (packet switching)

Link Layer

- Packets are transmitted across wire/wireless to Internet Service Provider
- ISP relays packets across various hops in the network

Example: receiving a webpage request

Application Layer – Web server

- Processes the HTTP request, maybe prepares a response

Transport Layer (TCP)

- Collects packets for a specified port
- Assembles packets of request in order.
- Determines if there are any errors, and if so requests retransmission.
- Sends complete HTTP request to application for this port

Internet Layer

- Collects packets for this IP address

Link Layer

- Receives packets across “wire”

Network security protocols

- Network-related security protocols in common use include:
  - **Secure Shell (SSH):**
    - Used for remote login, file transfer, and limited VPN service.
  - **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.
  - **WiFi security (WEP, WPA, WPA2):**
    - Provides security services at the link layer for wireless communication

IETF Internet Protocol suite

Layer	Examples
Application	web (HTTP, HTTPS) email (SMTP, POP3, IMAP) login (SSH, Telnet)
Transport	connection-oriented (TCP) connectionless (UDP)
Internet	addressing and routing: • IPv4, IPv6 control (ICMP) security (IPsec)
Link	packet framing (Ethernet) physical connection • WLAN (WEP, WPA) • ADSL • GSM/3G

TLS

TRANSPORT LAYER SECURITY (TLS)

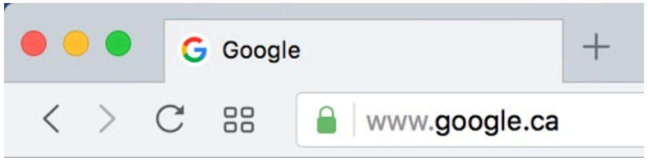
A.K.A. SECURE SOCKETS LAYER (SSL)

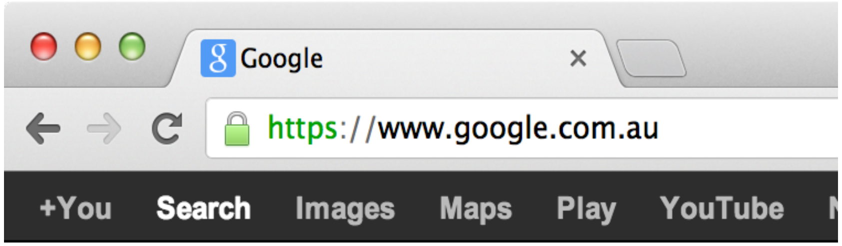
Terminology

- SSL: Secure Sockets Layer
- Proposed by Netscape
  - SSLv2: 1995
  - SSLv3: 1996

- TLS: Transport Layer Security
- IETF Standardization of SSL
  - TLSv1.0 = SSLv3: 1999
  - TLSv1.1: 2006
  - TLSv1.2: 2008
  - TLSv1.3: 2018

- HTTPS: HTTP (Hypertext Transport Protocol) over SSL





TLS

- Transport Layer Security (TLS) is a cryptographic protocol that operates above the transport layer to provide security services to applications
  - TLS runs over TCP
  - Datagram TLS (DTLS) runs over UDP
- Consists of a variety of modes and has many options
- Usually relies on a public key infrastructure

IETF Internet Protocol suite

TLS adds encryption to many application level protocols

Layer	Examples
Application	web (HTTP, HTTPS) email (SMTP, POP3, IMAP) login (SSH, Telnet)
Transport	connection-oriented (TCP) connectionless (UDP)
Internet	addressing and routing: • IPv4, IPv6 control (ICMP) security (IPsec)
Link	packet framing (Ethernet) physical connection • WLAN • ADSL • GSM/3G

TLS

Security goals of TLS

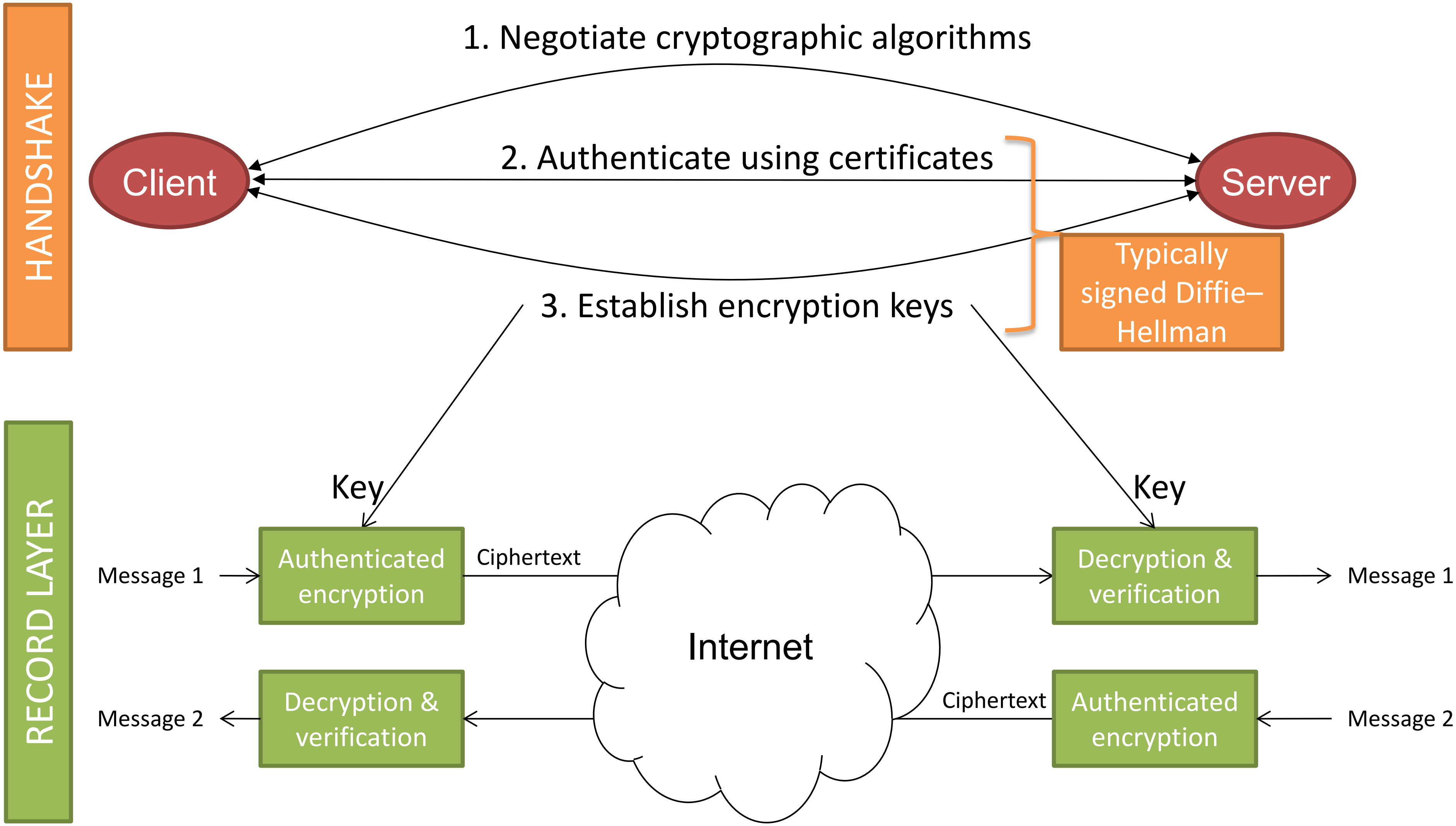
- Provides **authentication** based on public key certificates
  - server-to-client (always)
  - client-to-server (optional)
- Provides **confidentiality** and integrity of message transmission

- But only protects confidentiality if authentication is correct.

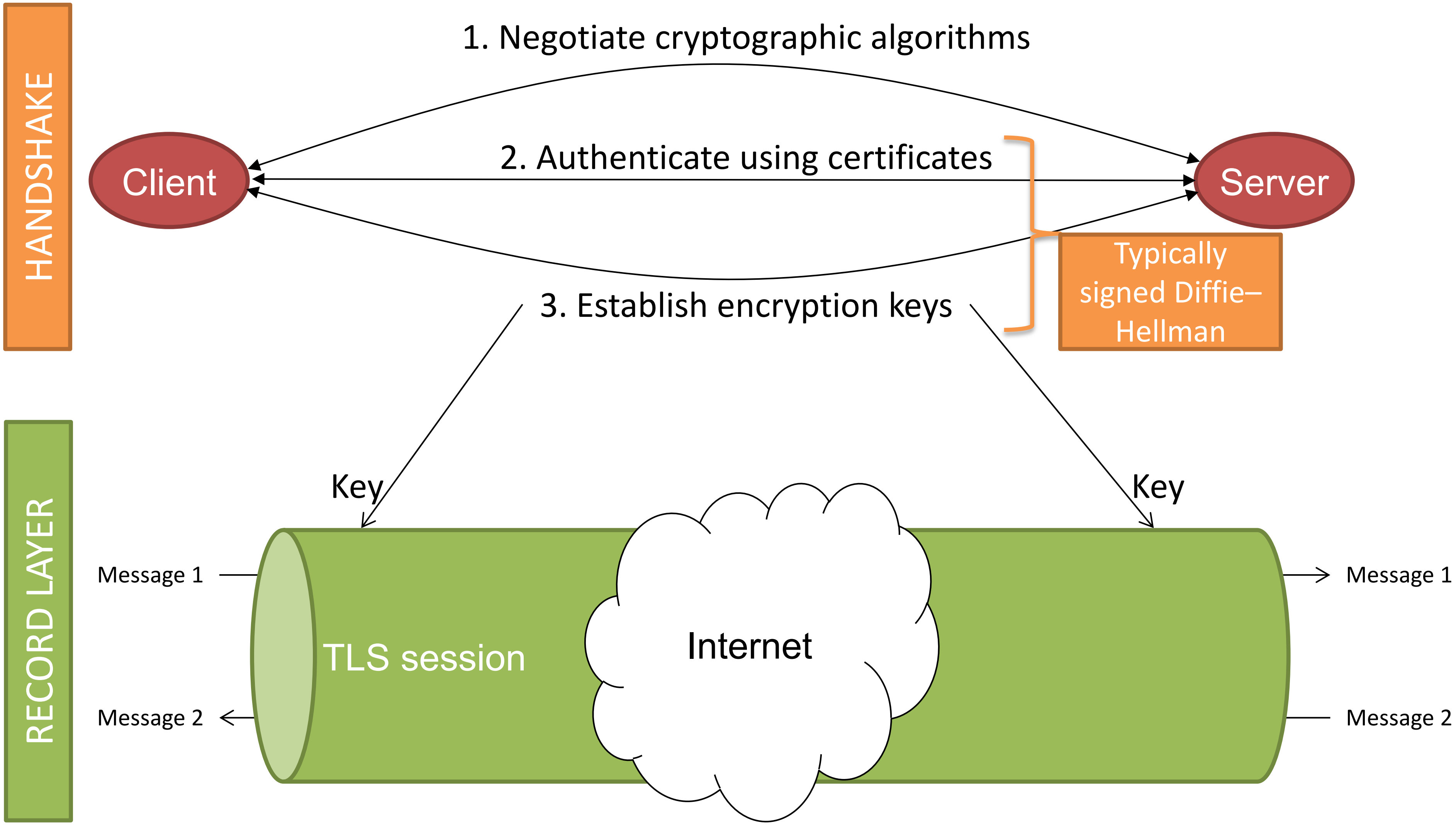
TLS and HTTP

- TLS can be used to provide protection for HTTP communications:
  - Port 443 is reserved for HTTP over TLS
  - HTTPS is the name of the URL scheme used with this port.
  - http://www.example.com/ implies the use of standard HTTP using port 80
  - https://www.example.com/ implies the use of HTTP over TLS using port 443.

SSL/TLS Protocol



SSL/TLS Protocol



Structure of TLS

HANDSHAKE PROTOCOL

Negotiation of cryptographic parameters

Authentication (one-way or mutual) using public key certificates

Establishment of a master secret key

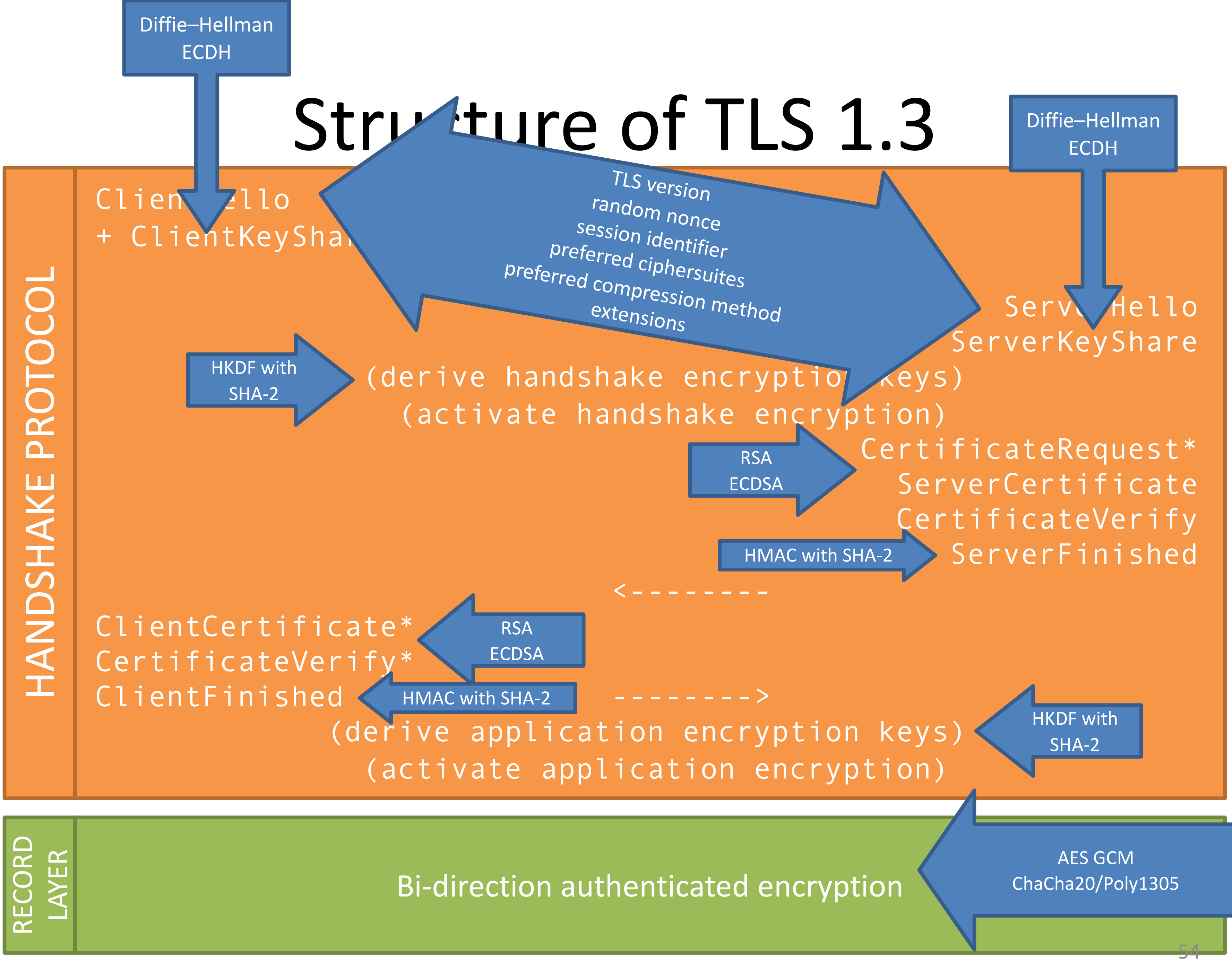
Derivation of encryption and authentication keys

Key confirmation

RECORD LAYER

Bi-direction authenticated encryption

Structure of TLS 1.3



TLS: Handshake Protocol

- Provides one service for TLS connections:
  - **Authentication** (server-to-client)
    - Ensures that the connection really is with the server with the given domain name
    - Typically uses X.509 certificates
    - Optionally can do client-to-server authentication

- Handshake protocol also establishes keys that will be used in the record protocol for additional security services.

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 is used to establish a symmetric key to be used to encrypt SSL/TLS payloads in the record protocol.
  - **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 Message Authentication Code.
- Supplied by an **authenticated encryption** (with associated data) scheme (AEAD)

Is TLS secure?

What should TLS do?

- Server-to-client authentication
- Client-to-server authentication (optional)
- Confidential communication with integrity protection

What doesn't TLS do?

- (Trusted creation of certificates)
- Password-based authentication
- Stop denial of service attacks
- Prevent web application vulnerabilities

TLS security considerations

Trust and digital certificates

- TLS uses public keys – provided in digital certificates
- Certificates should be verified – requires tracing certificate pathways
- Web browsers come with pre-configured lists of root certificates but users can add or remove root CAs

One-way or mutual authentication?

- Authentication is usually of server to client only, not mutual
- Users usually do not have client certificates
- Typically, authentication of users is not performed in handshake
- Instead, password authentication over server-authenticated HTTPS channel

58

Many attacks on TLS

Target	Attack Name	Year	Reference
Core cryptography			
RSA PKCS#1v1.5 decryption	Side channel – Bleichenbacher	1998*, 2014	[12]*, [37]
DES	Weakness – brute force	1998	[21]
MD5	Weakness – collisions	2005	[32]
RC4	Weakness – biases	2000*, 2013,15	[24, 34]*, [4, 48, 33]
RSA export keys	FREAK	2015	[8]
DH export keys	Logjam	2015	[2]
RSA-MD5 signatures	SLOTH	2016	[11]
Triple-DES	Sweet32	2011*, 2016	[44]*, [10]
Crypto usage in ciphersuites			
CBC mode encryption	BEAST	2002*, 2011	[38]*, [20]
Diffie–Hellman parameters	Cross-protocol attack	1996*, 2012	[50]*, [36]
MAC-encode-encrypt padding	Lucky 13, Lucky microseconds	2013,15	[5, 3]
CBC mode encryption + padding	POODLE, ZombiePoodle, GoldenDoodle	2014,19	[39, 52]
TLS protocol functionality			
Support for old versions	Jager et al., DROWN	2015, 2016	[27, 6]
Negotiation	Downgrade to weak crypto	1996, 2015	[50, 8, 2]
Termination	Truncation, Cookie Cutter	2007,13,14	[7, 45, 9]
Renegotiation	Renegotiation attack	2009	[42]
Compression	CRIME, BREACH, HEIST	2002*, 2012,16	[28]*, [43, 41, 47]
Session resumption	Triple-handshake attack	2014	[9]
Pre-shared keys	Selfie†	2019	[19]
Implementation – libraries			
OpenSSL – RSA	Side-channel	2005, 2007	[40, 1]
Debian OpenSSL	Weak RNG	2008	[46]
OpenSSL – elliptic curve	Side-channel	2011–14	[15, 14, 51]
Apple – certificate validation	goto fail;	2014	[31]
OpenSSL – Heartbeat extension	Heartbleed	2014	[16, 17]
Multiple – certificate validation	Frankencerts	2014	[13]
NSS – RSA PKCS#1v1.5 signatures	BERserk (Bleichenbacher)	2006*, 2014	[23]*, [30]
Multiple – state machine	CCS injection, SMACK	2014, 2015	[29, 8]
Implementation – HTTP-based applications			
Netscape	Weak RNG	1996	[26]
Multiple – certificate validation	"Most dangerous code", MalloDroid	2012	[25, 22]
Application-level protocols			
HTTP	SSL stripping	2009	[35]
HTTP server virtual hosts	Virtual host confusion	2014	[18]
IMAP/POP/FTP	STARTTLS command injection	2011	[49]

* denotes theoretical basis for a later practical attack; † denotes TLS 1.3-specific attack.

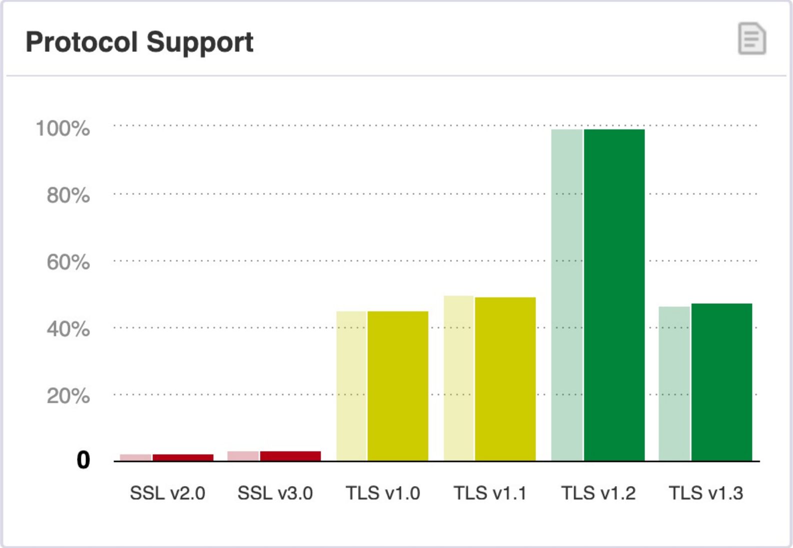
https://www.douglas.stebila.ca/research/presentations/tls-attacks/

(Perfect) Forward secrecy

- An adversary who later learns the server's long-term private key shouldn't be able to read previous transmissions
- Signed Diffie–Hellman key exchange provides forward secrecy
- TLS ≤1.2 supported RSA public key encryption for key exchange which does not provide forward secrecy

TLSv1.3: The Next Generation

- Multi-year process involving good interaction between academics and industry
- Standardized in August 2018



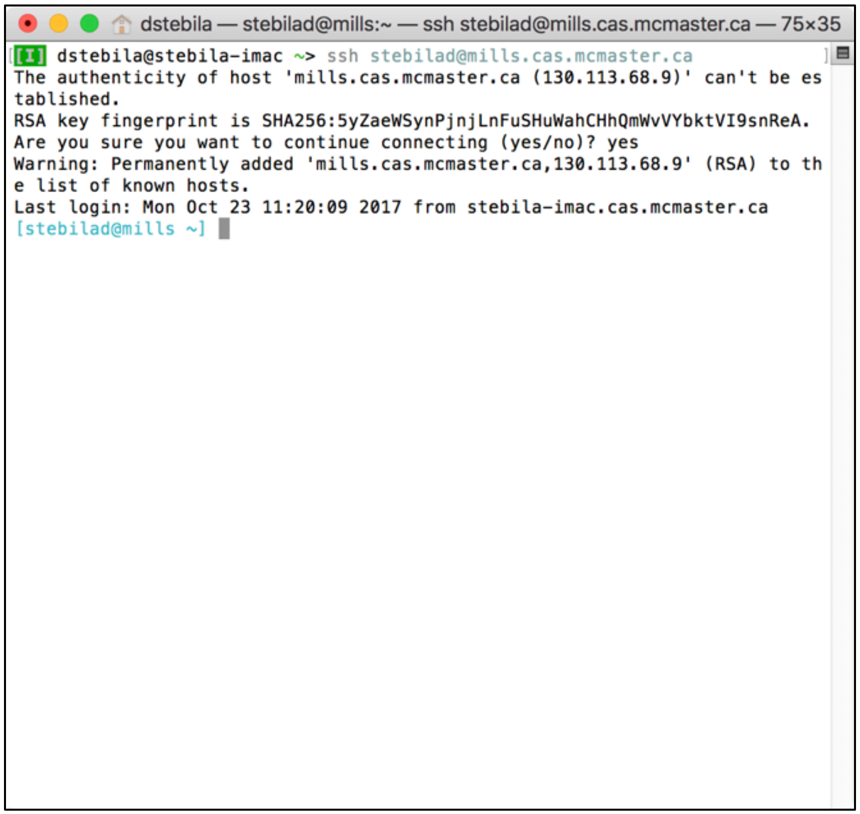
- Primary goals:
  - remove ciphersuites without forward secrecy
  - remove obsolete / deprecated algorithms
  - provide low-latency mode with fewer round trips
  - encrypt more of the handshake to improve privacy

https://www.ssllabs.com/ssl-pulse/ July 11, 2021

SSH, IPsec

OTHER PROTOCOLS

SSH (Secure Shell) protocol



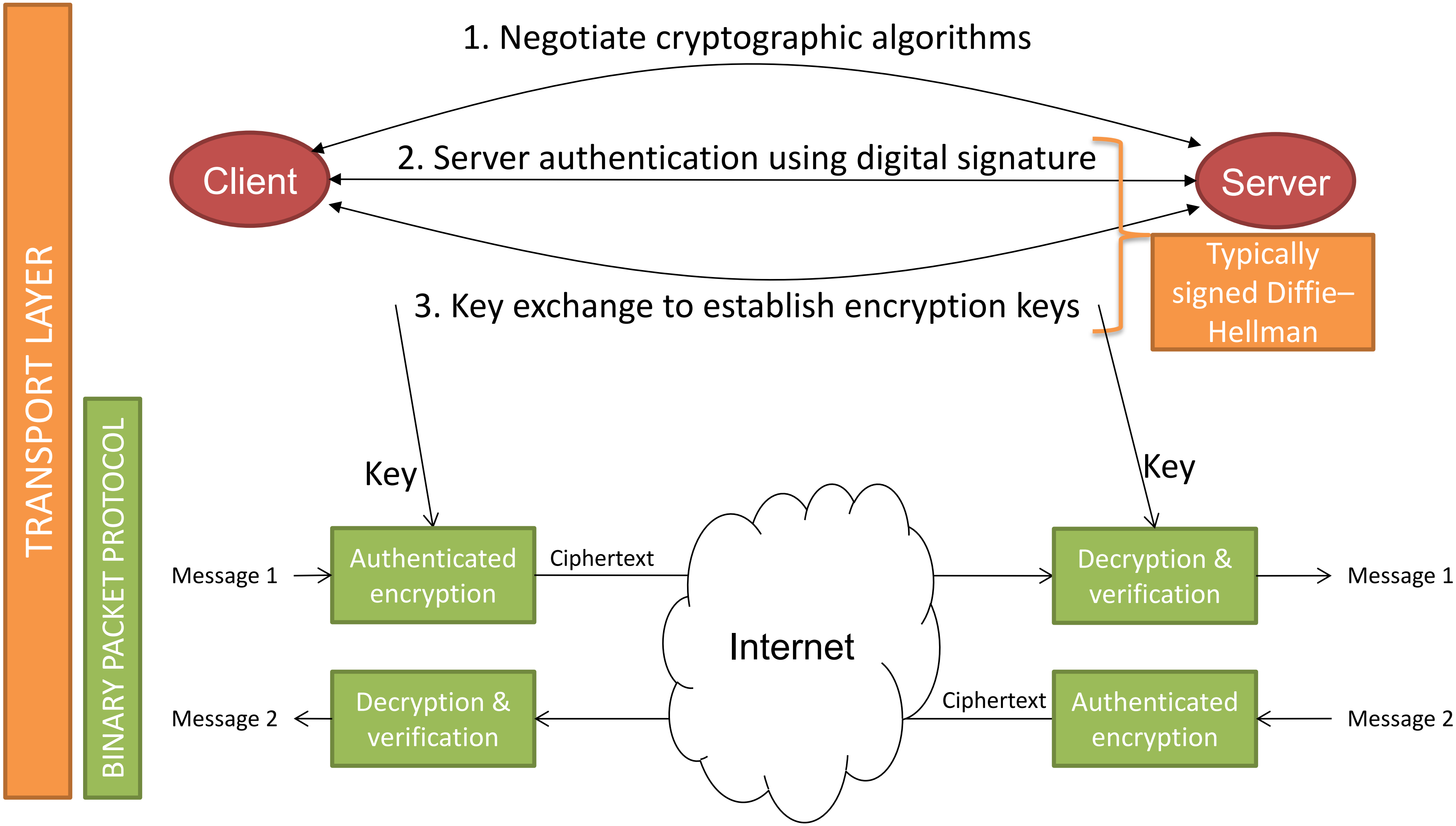
- SSH used for secure remote access (like telnet, but secure)
  - Occasionally used as a "poor man's VPN"
- Run over TCP, typically on port 22
- Provides public key authentication of servers and clients and encrypted communication
- Specified in RFCs by the IETF

Use of SSH

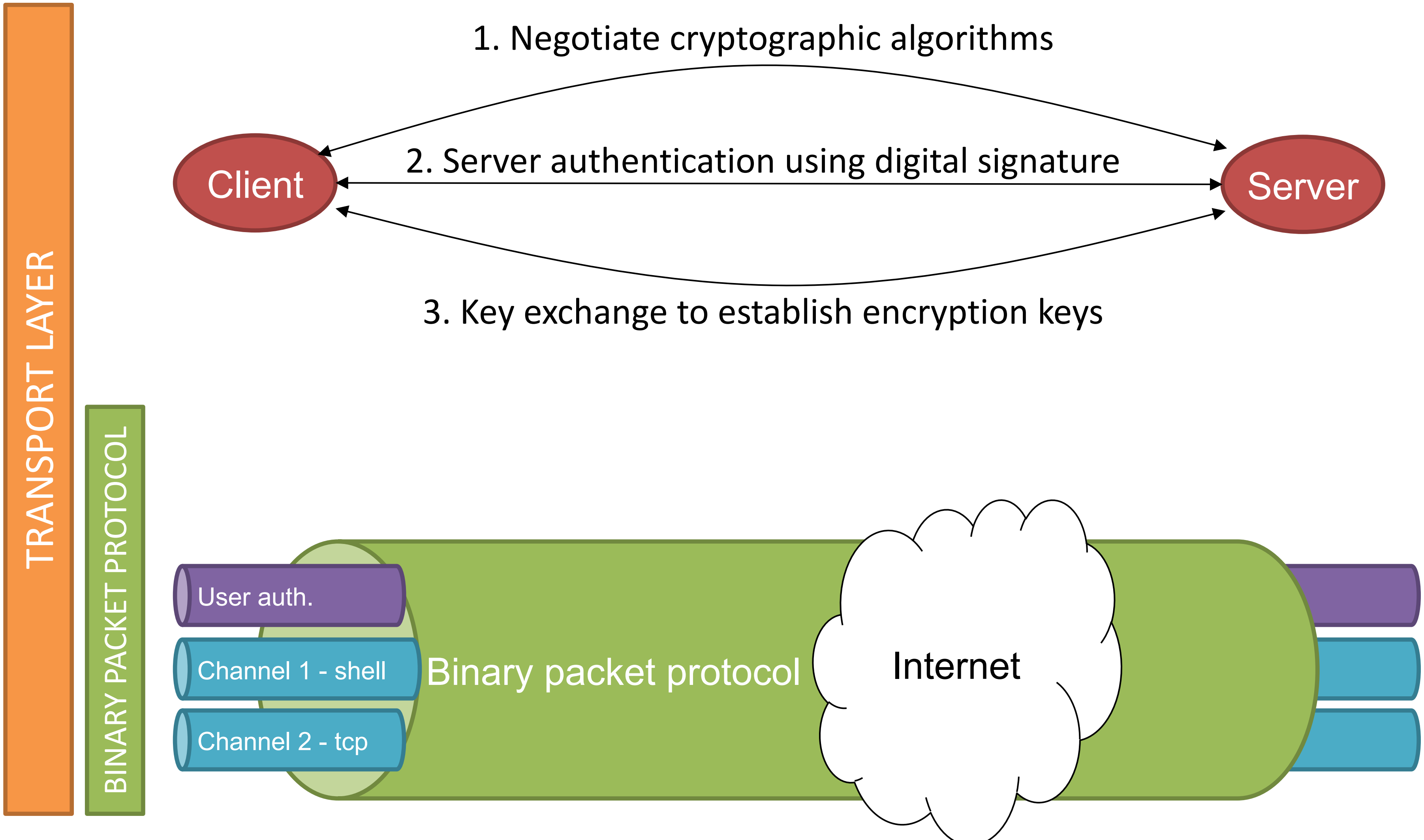
- Primarily used as an application itself (remote login)
- Occasionally used as a "poor man's VPN"

Layer	Examples
Application	web (HTTP, HTTPS) email (SMTP, POP3, IMAP) login (SSH, Telnet)
Transport	connection-oriented (TCP) connectionless (UDP)
Internet	addressing and routing: • IPv4, IPv6 control (ICMP) security (IPsec)
Link	packet framing (Ethernet) physical connection • WLAN (WEP, WPA) • ADSL • GSM/3G

SSH protocol



SSH protocol

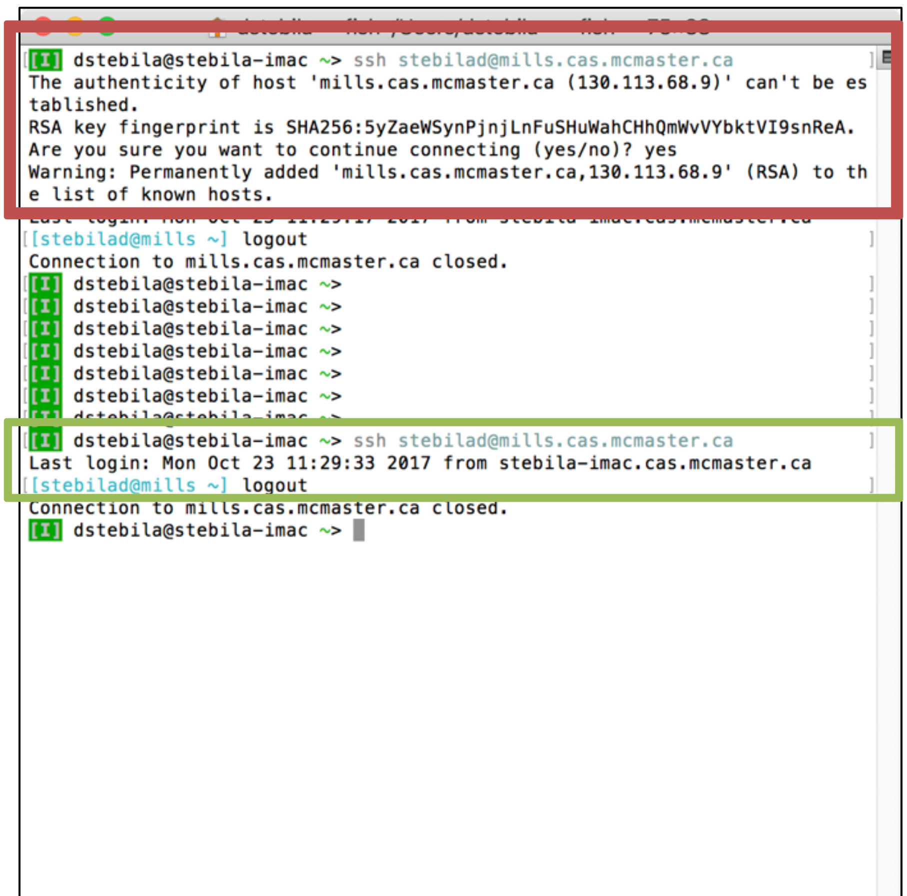


Security goals of SSH

- **Message Confidentiality**
  - Protects against unauthorised data disclosure
  - Achieved using encryption
- **Message Integrity**
  - Protects against unauthorised changes to data during transmission (intentional or unintentional)
  - Achieved using message authentication code
- **Message Replay Protection**
  - The same data is not delivered multiple times
  - Achieved using counters and integrity protection
- **Peer Authentication**
  - Ensures that traffic is being sent from the expected party
  - Server-to-client auth:
    - based on public keys
  - Client-to-server auth:
    - based on passwords or public keys

Server authentication in SSH

- Based on public key digital signatures



- Unlike TLS, (typically) does not use X.509 certificates – just a raw public key
- No systematic solution for authentic distribution of public keys
  - Console displays public key fingerprint (hash) on first login
  - User should check hash through some out-of-band method
    - E.g. phone call to sysadmin
  - SSH client saves hash for future logins and raises alert if changed

Client authentication in SSH

- Based on passwords or public key digital signatures

- Security-conscious installation would disable password-based authentication and only support public key authentication

Public key client authentication in SSH

- "Many-to-many mapping"

- **Each account can have multiple associated public keys**
  - Multiple users can login to a single account without having to be told the password for that account. Easy to revoke one user’s access to that account
  - One user could have a different key from each local computer (laptop, desktop, ...); if one of local computer is lost/compromised, easy to revoke its access

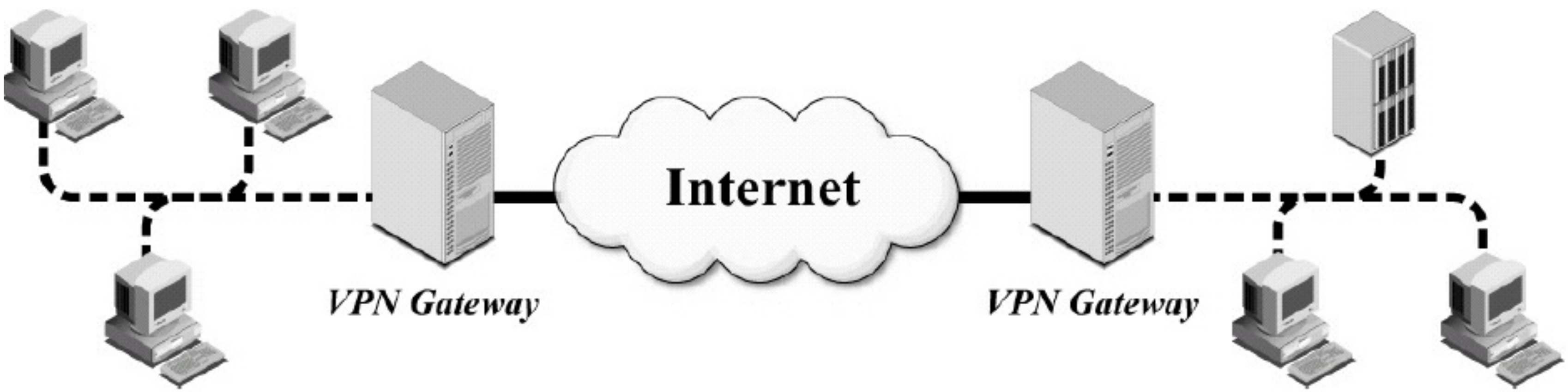
- **Each user can associate the same public key with multiple accounts on multiple servers**
  - Yields a form of single sign-on
  - Users can & should protect their private key using a password

IPsec (Internet Protocol Security)

- Provides confidentiality and authentication for Internet communications
- Works at the IP layer of the protocol stack
  - TLS works at higher levels, so applications have to be designed to use TLS
  - IPsec can be used transparently with any application
- Often used for Virtual Private Networks (VPNs)

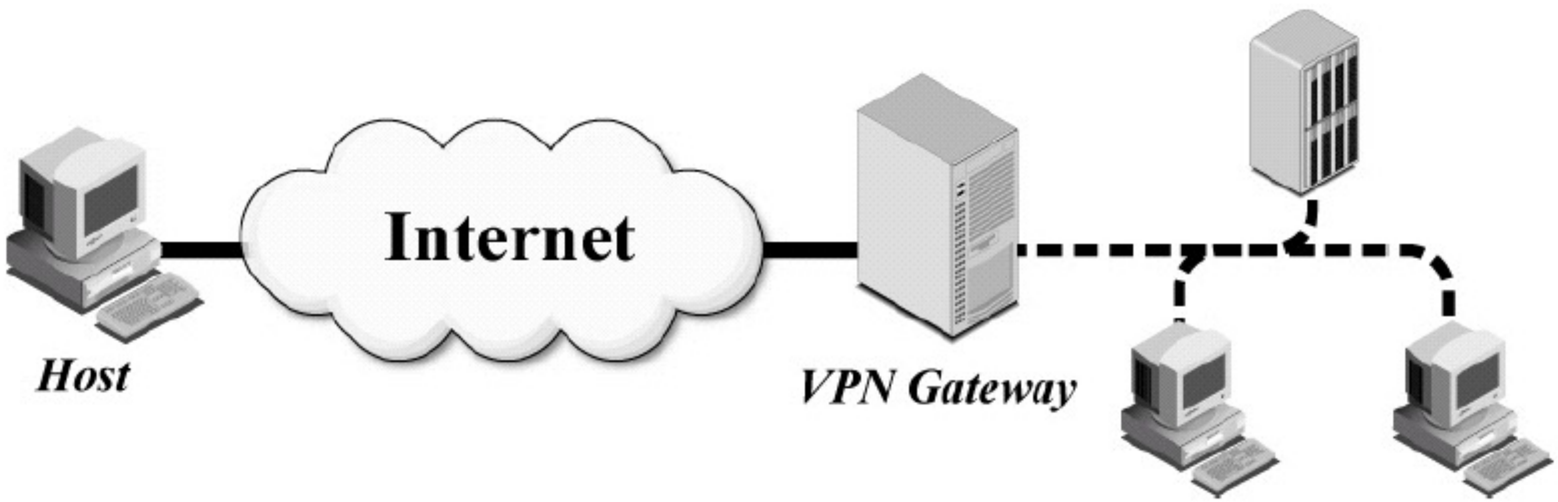
Layer	Examples
Application	web (HTTP, HTTPS) email (SMTP, POP3, IMAP) login (SSH, Telnet)
Transport	connection-oriented (TCP) connectionless (UDP)
Internet	addressing and routing: • IPv4, IPv6 control (ICMP) security (IPsec)
Link	packet framing (Ethernet) physical connection • WLAN (WEP, WPA) • ADSL • GSM/3G

IPsec: Common Architectures
Gateway-to-gateway



Source: NIST Special Publication 800-77

IPsec: Common Architectures
Host-to-gateway



Source: NIST Special Publication 800-77

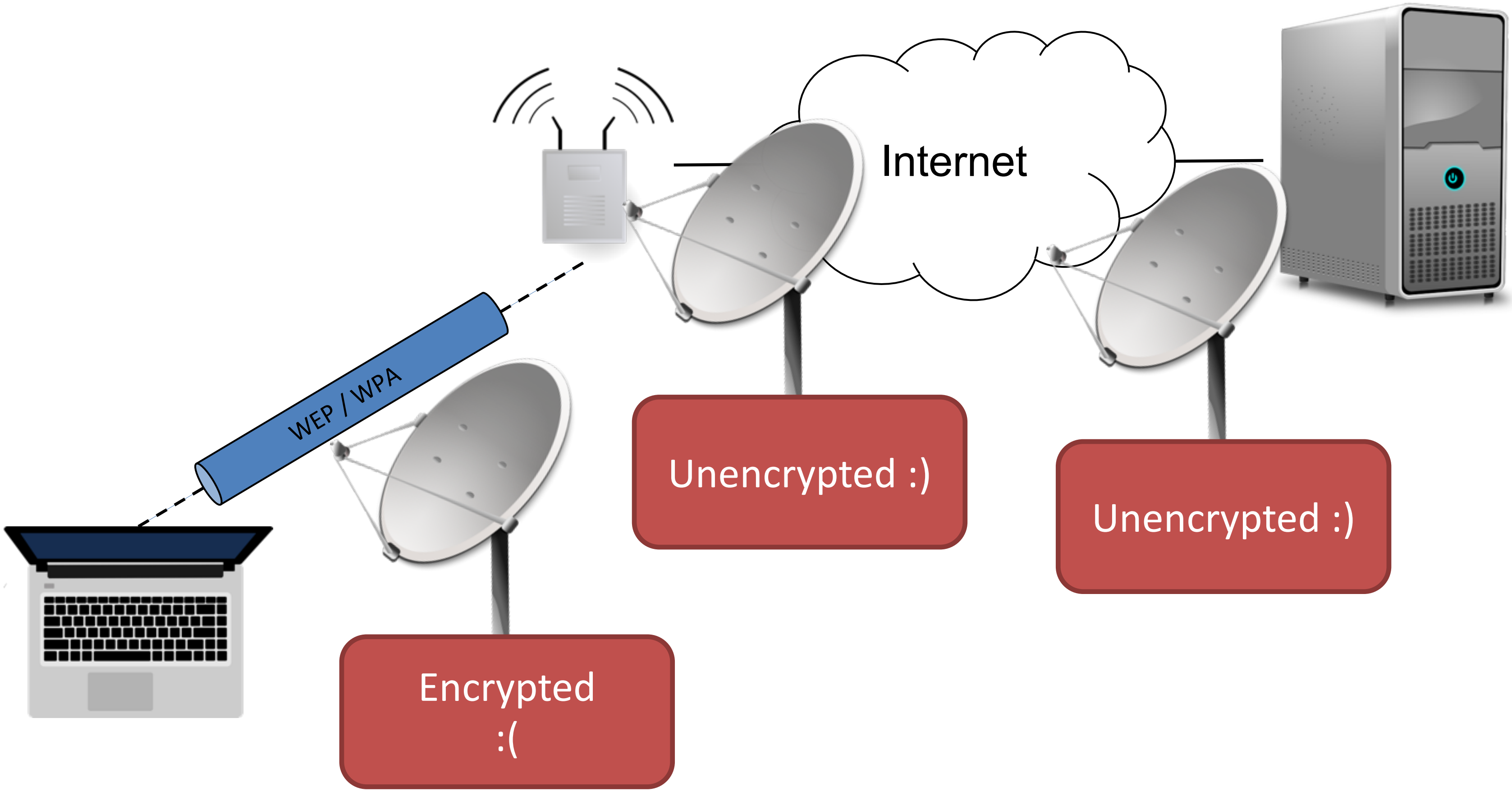
WIRED EQUIVALENT PRIVACY (WEP)

IETF Internet Protocol suite

Layer	Examples
Application	web (HTTP, HTTPS) email (SMTP, POP3, IMAP) login (SSH, Telnet)
Transport	connection-oriented (TCP) connectionless (UDP)
Internet	addressing and routing: • IPv4, IPv6 control (ICMP) security (IPsec)
Link	packet framing (Ethern physical connection • WLAN (WEP, WPA) • ADSL • GSM/3G

WEP and WPA are to packets before they are transmitted over WiFi

WiFi Security



Wireless LAN

IEEE 802.11

- Working group of the IEEE (Institute of Electrical and Electronic Engineers)
- Various standards for WLAN protocols
  - Core OSI layer 1 and layer 2 WLAN standards: 802.11, 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, …
  - Security standards: 802.11i, 802.1X, …

Wi-Fi Alliance

- Trade group that owns the Wi-Fi trademark and licenses it to products that comply with a certain subset of (rebranded) IEEE standards for interoperability
  - ≥ 1 of 802.11 a/b/g/n
  - Wi-Fi Protected Access II (WPA2) ≈ 802.11i

Wireless LAN security protocols

	IEEE	Wi-Fi Alliance	
1997	Wired Equivalent Privacy (WEP)	N/A	Included in original 802.11 standard
2003	802.11i draft	Wi-Fi Protected Access (WPA)	
2004	802.11i	Wi-Fi Protected Access II (WPA2)	
2001–2004	802.1X	WPA-Enterprise WPA2-Enterprise	
2006	N/A	Wi-Fi Protected Setup (WPS)	
2018	802.11-2016	WPA3-Personal WPA3-Enterprise	

Wired Equivalent Privacy (WEP)

- Part of the original 802.11 standard in 1997
- **Entity Authentication**:
  - Open System authentication:
    - Basically no authentication
    - Public WLAN with capture/splash screen
  - Ethernet MAC address – easily spoofed
  - Shared Key authentication:
    - Challenge-response protocol based on knowledge of pre-shared key
- **Confidentiality & Integrity**:
  - Encryption using RC4 with various key sizes
  - Integrity using CRC-32 checksum

Insecurity of WEP

- **Entity authentication**: completely insecure; attacker can impersonate after seeing a single packet
- **Message integrity**: completely insecure; attacker can undetectably modify any packet with 100% success rate
- **Message confidentiality:** completely insecure; attacker can recover secret key with high probability in just a minute using readily available tools

Wi-Fi Protected Access II (WPA2)

- Wi-Fi Alliance name for the IEEE 802.11i final standard of 2014
- **Goal:** improve security compared to WEP
- **Entity Authentication:**
  - WPA-Personal, WPA-Enterprise, Wi-Fi Protected Setup
- **Confidentiality & Integrity**:
  - Encryption: AES in Counter Mode
  - Integrity: AES-CBC-MAC
  - "CCMP": CTR mode with CBC-MAC Protocol

IETF Internet Protocol suite

Layer	Examples
Application	web (HTTP, HTTPS) email (SMTP, POP3, IMAP) login (SSH, Telnet)
Transport	connection-oriented (TCP) connectionless (UDP)
Internet	addressing and routing: • IPv4, IPv6 control (ICMP) security (IPsec)
Link	packet framing (Ethernet) physical connection • WLAN (WEP, WPA) • ADSL • GSM/3G

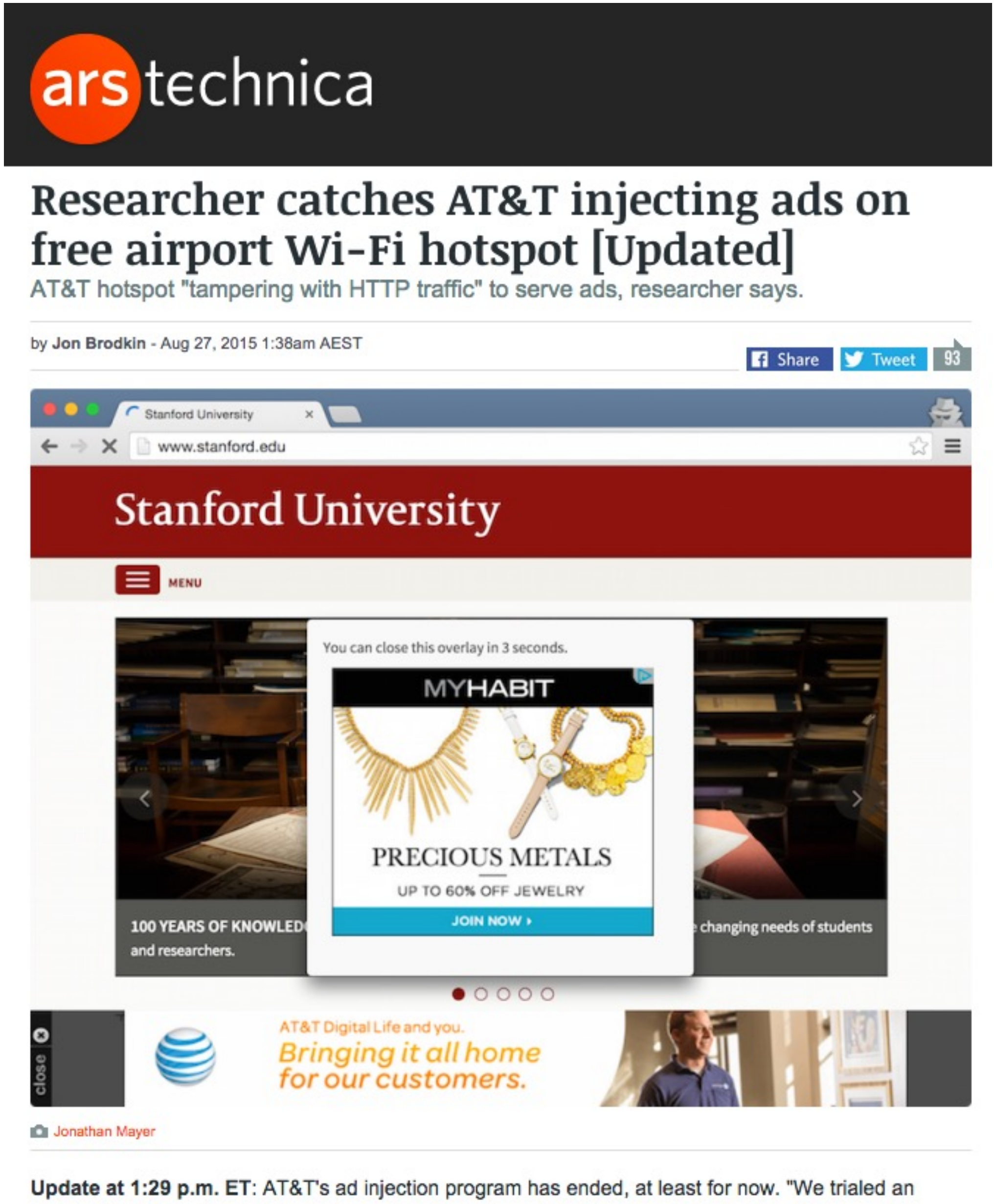
TLS

Case study: Injecting ads in Wi-Fi hotspot

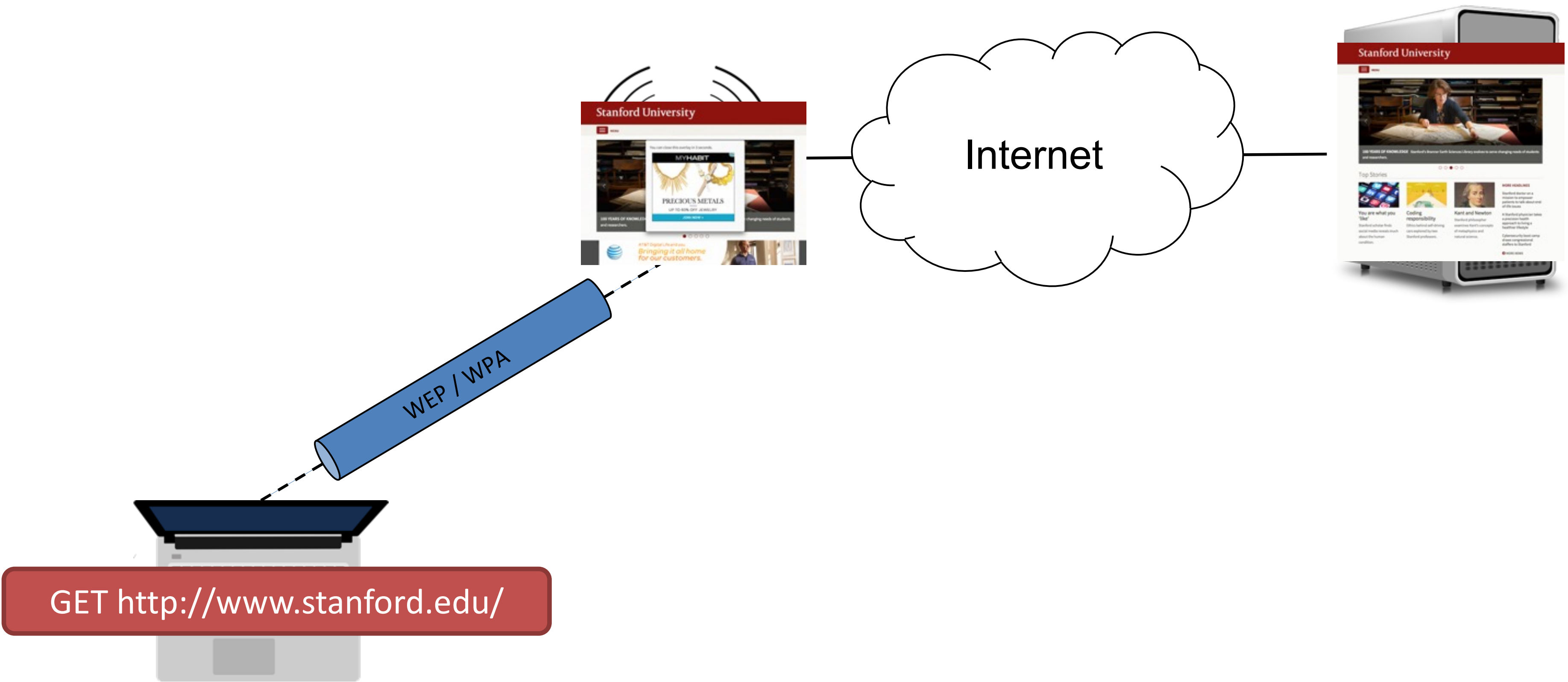
AT&T provides free Wi-Fi hotspots in airports.

In addition to making users view ads when they first connected to the hotspot, AT&T was also modifying HTTP responses from web servers to include their own ads on pages.

http://arstechnica.com/business/2015/08/atts-free-wi-fi-hotspot-injects-extra-ads-on-non-att-websites/



Case study:
Injecting ads in Wi-Fi hotspot



Case study:
Injecting ads in Wi-Fi hotspot

Layer	Examples
Application	web (HTTP, HTTPS) email (SMTP, POP3, IMAP) login (SSH, Telnet)
Transport	connection-oriented (TCP) connectionless (UDP)
Internet	addressing and routing: • IPv4, IPv6 control (ICMP) security (IPsec)
Link	packet framing (Ethernet) physical connection • WLAN (WEP, WPA) • ADSL • GSM/3G

TLS

- Link-layer security would **not** protect against this attack
  - WEP/WPA
- Internet-layer, transport-layer, and application-layer would protect against this attack
  - IPsec: Use a VPN to a trusted gateway.
  - TLS: Encryption/integrity protection for web page connections.
  - SSH: Encryption/integrity protection for remote login.

Assignment 1

1a) Secure email - PGP

- Generate a public key / private key pair
- Send me an encrypted email using PGP

Assignment 0

Downloading and installing VirtualBox and Kali Linux

1b) HTTPS connections

- Inspect X.509 certificates used in a browser
- Use Wireshark to examine the messages in a TLS connection
  - Can do in Kali Linux or in a local installation of Wireshark

1c) Choosing network security protocols

- Discuss the use of different protocols

https://www.douglas.stebila.ca/teaching/cryptoworks21/