

Communication Protocols and Internet Architectures

Harvard University

Lecture #12

Instructor: Len Evenchik
cs40@evenchik.com or evenchik@fas.harvard.edu

ALIGHSOD1701

© 1998 - 2017 L. Evenchik

Lecture Agenda

- Course Logistics
- Q&A and Topics from Last Week
- Digital Signatures
- Website Security, Certificates and TLS
- Firewalls and VPN
- VoIP
- Session Initiation Protocol (SIP)
- One Minute Wrap-Up

© 1998 - 2017 L. Evenchik

Course Logistics

© 1998 - 2017 L. Evenchik

Course Logistics

- Final Exam – Please check the weekly course information sheet for detailed information on the final.
- Upcoming Guest Lectures
- Homeworks #4 and #5 have been posted.
- Always check the weekly course information sheet for any updated schedule information for section meetings.
- **Please submit a one minute wrap-up each week.
Thank You!**

© 1998 - 2017 L. Evenchik

Q&A

Topics from Last Week

© 1998 - 2017 L. Evenchik

In Summary, Security Requires:

- Hardware
- Software
- Written Procedures and Processes
- People educated on what security means and how to properly do it.

Security is a system issue which requires all of the above, but without a doubt, people who understand and care about the issues are the most important element.

© 1998 - 2017 L. Evenchik

A More Technical Approach to Security

© 1998 - 2017 L. Evenchik

Structured way to Think about Security: Five Important Elements

- Privacy and confidentiality
- Authentication
- Authorization
- Integrity
- Nonrepudiation

© 1998 - 2017 L. Evenchik

Digital Signatures

© 1998 - 2017 L. Evenchik

Digital Signatures

- A digital signature should “prove” that a message came from a specific user (lets call them UserA) and that the message has not been changed.
- A digital signature does not encrypt the message.
- What is an example of why you might not want to encrypt the message or document, but still validate that it was from a specific user and that it had not changed
- One way to produce a digital signature
 - UserA computes a one-way hash function on the contents of the message...
 - .. *Lets work out the details, we will need to use public key encryption and hashing*

© 1998 - 2017 L. Evenchik

Digital Signatures

- A digital signature should “prove” that a message came from a specific user (lets call them UserA) and the message has not changed
- One way to produce a digital signature
 - UserA computes a one-way hash function on the contents of the message
 - UserA encrypts the hash code using their private key
 - The encrypted hash code is appended to the message and the combination is sent to UserB
 - UserB computes the same hash function on the contents of the message
 - UserB then decrypts the received hash code with UserA’s public key
 - If the hash codes match, the message came from UserA and the message was not changed in transit

© 1998 - 2017 L. Evenchik

Signing of “testfile1”

```
cmd (~): gpg --clearsign < testfile1
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

this is a test file to be used in the networks and
protocols class...
abcdefghijklmnopqrstuvwxyz1234567890
Hello World
This is line five (5) of this file.
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.0.7

iQCVAwUBP8u0S+nwDzqNmKQTAQIzEAQAIYTHo
PS4GZMUjFyzItigG2nWXuI3867oYyvPp/D9q+jTR6O
PapnwowXpgqJIZn0mluxMoTO0pSkygcC3ILqo0o4
W5z6BN8ykfdXoyDMCuuh4+n133OgjjYS/lYrq9org+
gEw9nn4Chyyq5LvbHwgo1B6fr1ml+HGi4P4PvwdCDM=
=ZQMq
-----END PGP SIGNATURE-----
cmd (~):
```

© 1998 - 2017 L. Evenchik

© 1998 - 2017 L. Evenchik

Web Site Security Certificates, CAs and SSL/TLS

© 1998 - 2017 L. Evenchik

What Do We Want to Accomplish?

- Authentication of the Server: How can you demonstrate that the server (or resource) you are communicating with is the server (or resource) you think it is? (Clients can also be authenticated given the proper infrastructure.)
- Confidentiality: How is the data encrypted between the client and the server such that others cannot eavesdrop on your conversation?
- Integrity: How do we ensure that the data on the webpage was not changed as it was sent.

© 1998 - 2017 L. Evenchik

Building Blocks for Web Site Security

- Public/Private Key Encryption
- Symmetric Key Encryption
- Hashing and Message Digests
- Digital Signatures
- Certificate Authorities (CAs)
- Certificates and the Chain of Trust
- SSL/TLS and a lot of other software

© 1998 - 2017 L. Evenchik

Sending a secret message: Option 1

Alice wants to send a secret message to Bob via the Internet

- Bob sends his Public key to Alice via the Internet
- Alice encrypts the message with Bob's Public key
- Alice sends the encrypted message to Bob via the Internet
- Bob decrypts the message

Problem NOT solved. What is wrong with this approach?

© 1998 - 2017 L. Evenchik

Sending a secret message: Option 2

Alice wants to send a secret message to Bob via the Internet

Assume that Bob has somehow already sent his Public key to Alice in a secure way, and that Alice has securely stored Bob's Public key on her computer.

- Alice encrypts the message with Bob's Public key
- Alice sends the encrypted message to Bob via the Internet
- Bob decrypts the message (*but note that the message has not been authenticated as coming from Alice*)

Problem solved for this very specific case, but what happens if Alice wants to send secret messages to a lot of people? How can we scale this?

© 1998 - 2017 L. Evenchik

Sending a secret message: Option 3

Step 0

- Assume that Alice and Bob both trust an organization called a CA, and that the Public key of the CA has been securely loaded onto both Alice's and Bob's computer.

Step 1

- Bob creates a Public/Private key pair and sends his Public key to the CA in a secure way.
- The CA confirms that the Public key is from Bob and then digitally signs a document, called a Certificate, that includes Bob's Public key and some other information.
- The CA sends this Certificate back to Bob.

© 1998 - 2017 L. Evenchik

Sending a secret message: Option 3

Step 2 - Alice wants to send a secret message to Bob

- Bob sends his certificate, which includes his Public key, and which has been digitally signed by the CA, to Alice via the Internet.
- Alice verifies the integrity of the Bob's Certificate using the Public key of the CA. The Public key of the CA has been securely stored on Alice's computer since Step 0
- Given that the integrity of Bob's Certificate has been proven, Alice extracts and trusts Bob's Public key.
- Alice encrypts the message with Bob's Public key
- Alice sends the encrypted message to Bob via the Internet
- Bob decrypts the message

Note that this is greatly simplified and would not be used on a website.

© 1998 - 2017 L. Evenchik

X.509 Public Key Infrastructure

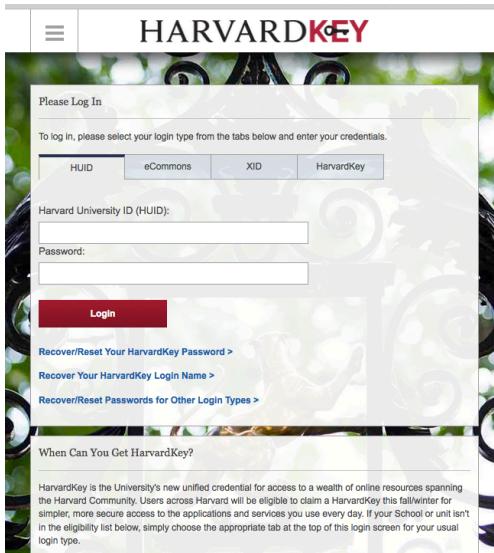
X.509 is NOT the same as TLS

- The PKI requires a trusted third party, called a Certificate Authority (CA), and the assumption is that the Public key of the CA has been securely loaded into browsers, clients, and phones.
- A website admin sends their Public key to a CA, and the CA validates the identity of the website and then digitally signs a document called a Certificate that includes the website's Public key.
- Given that the Public key of the CA has been previously and securely loaded into browsers, the validity of a website's Certificate, and hence the identity and Public key of a website, can be verified.
- A Certificate Chain, which is a hierarchical trust model, is common for CAs. The initial CA in the chain is called the root.
- There are 100s of CAs, some more trustworthy than others.
- Websites use a combination of Certificates and TLS to secure traffic. A Certificate is not the same as TLS.

© 1998 - 2017 L. Evenchik

© 1998 - 2017 L. Evenchik

Providing Secure Web Access: How do we know we are connected to a server at Harvard, and how is this web page encrypted?

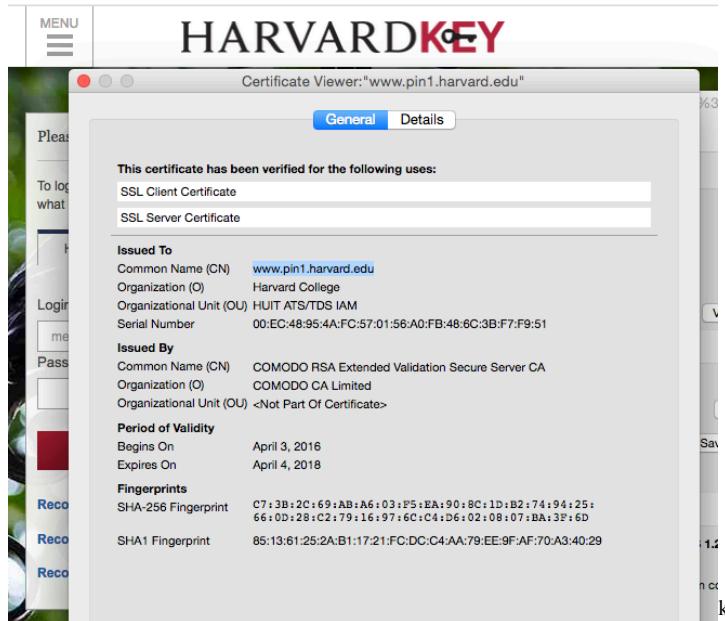


L. Evenchik

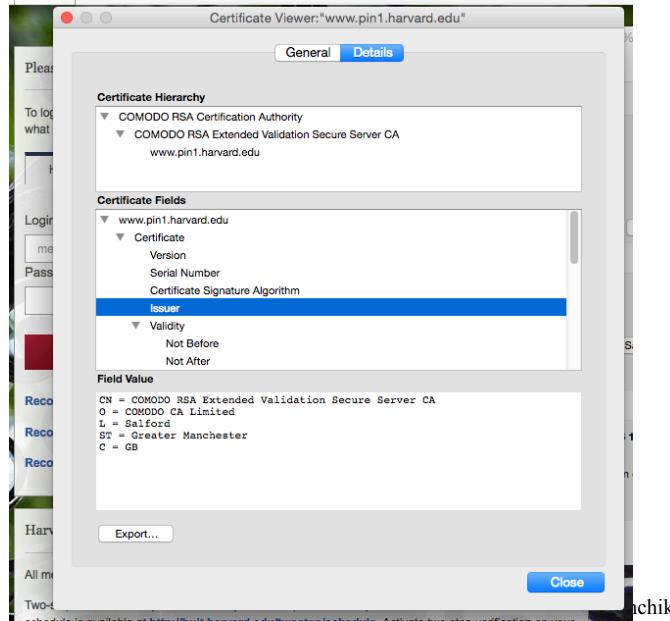
Web Page Security Information

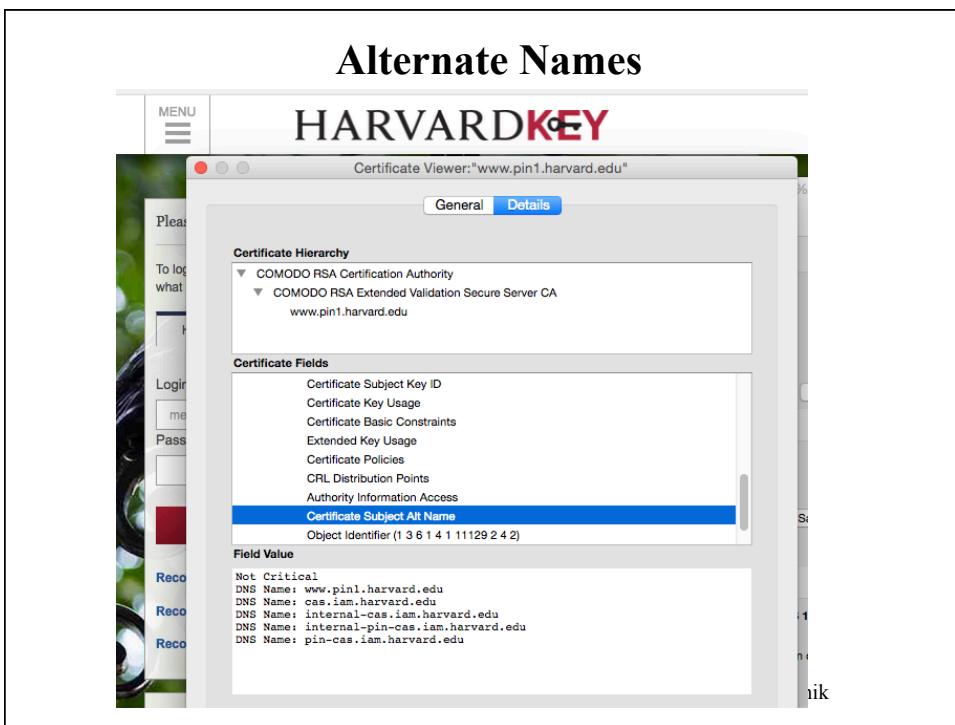
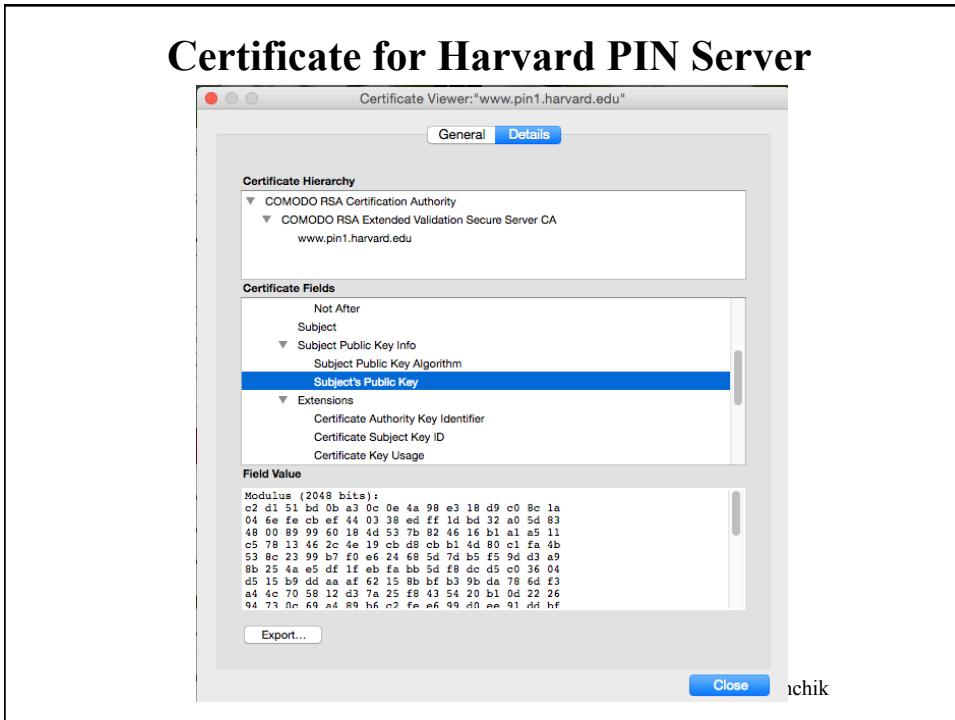
A screenshot of a browser window displaying security information for a website. The title bar shows "Page Info - https://www.pin1.harvard.edu/cas/login?service=https%3A%2F%2Fcan...". The browser tabs are "General", "Media", "Permissions", and "Security" (which is selected). The "Website Identity" section shows the website as "www.pin1.harvard.edu", owned by "Harvard College", and verified by "COMODO CA Limited". The "Privacy & History" section shows statistics: "Have I visited this website prior to today? Yes, 379 times" and "Is this website storing information (cookies) on my computer? Yes". The "Technical Details" section indicates "Connection Encrypted (TLS_DHE_RSA_WITH_AES_256_CBC_SHA, 256 bit keys, TLS 1.2)". It also states that the page is encrypted before being transmitted over the Internet, making it difficult for unauthorized people to view information traveling between computers.

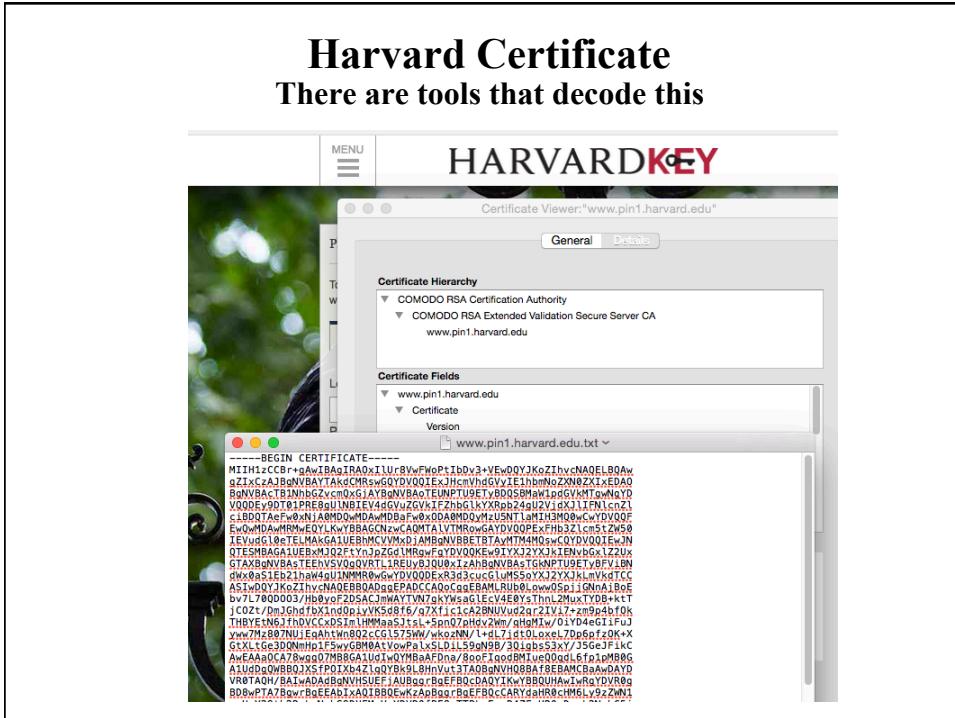
Certificate Information (1)



Certificate Issuer







Cert Decoding Tool

<https://www.sslshopper.com/certificate-decoder.html>

The Cert Decoder tool allows you to paste a PEM encoded SSL certificate into a text input field. It will then decode the certificate and display its details. The tool also includes links to the SSL Shopper homepage, SSL Wizard, SSL FAQ, and SSL Reviews.

Use this Certificate Decoder to decode your PEM encoded SSL certificate and verify that it contains all of the certificate key. Another simple way to view the information in a certificate on a Windows machine is to simply pasting the text of your certificate into the Certificate Decoder will do the rest. Your certificate should start with "-----BEGIN CERTIFICATE-----". Once you do the [SSL install](#) on your server, you can check to make sure it is using the [SSL Checker](#).

If you want to decode certificates on your own computer, run this OpenSSL command:

```
openssl x509 -in certificate.crt -text -noout
```

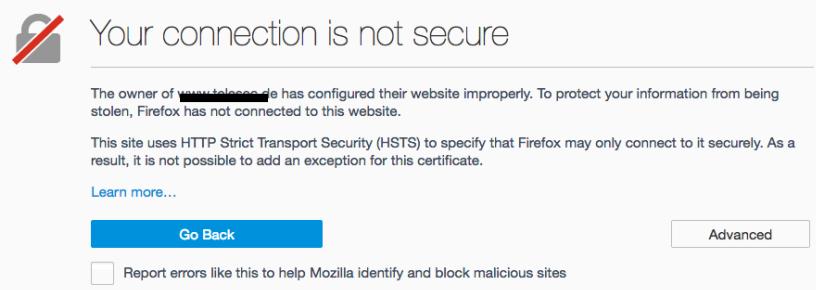
Paste Certificate Text

```
-----BEGIN CERTIFICATE-----
MIIEChCBiQIBAQUIBAQRADoxIUrBVwFwoPtbvW3+VEw0QYJkxZjhcnAQELBQAw
gZIxCAJBjNVAjYTAkCfHrsWgQYDVQ0Q1ExJHcmVhGvV1E1hbNzXnZkxEdAO
BdNVBaCTBjNvbGzvcmOxjAYBnVBAoTEUNPTU9ETyB0D5BMw1pdGVhMTowNaYD
VuVDEyDTB1PjRE8gUjNBEV46VjZGVh1Zb6LkXpp24u2vJ1JfNgnZl
-----END CERTIFICATE-----
```

L. Evenchik

Revoked Certificate

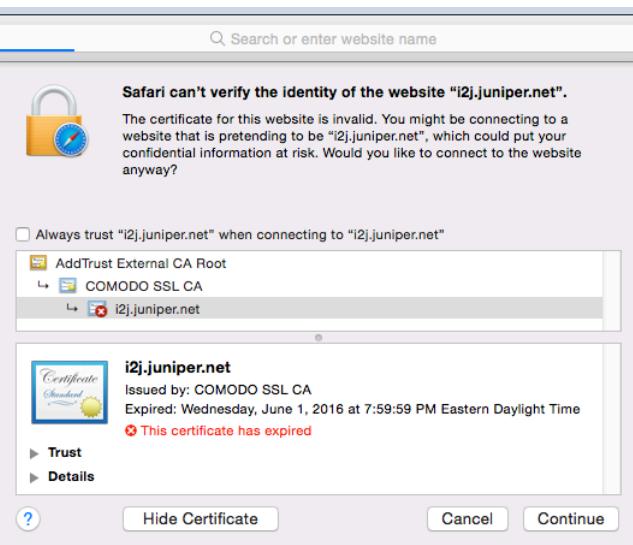
Example from Firefox, other Browsers Handle it Differently



© 1998 - 2017 L. Evenchik

Expired Certificate

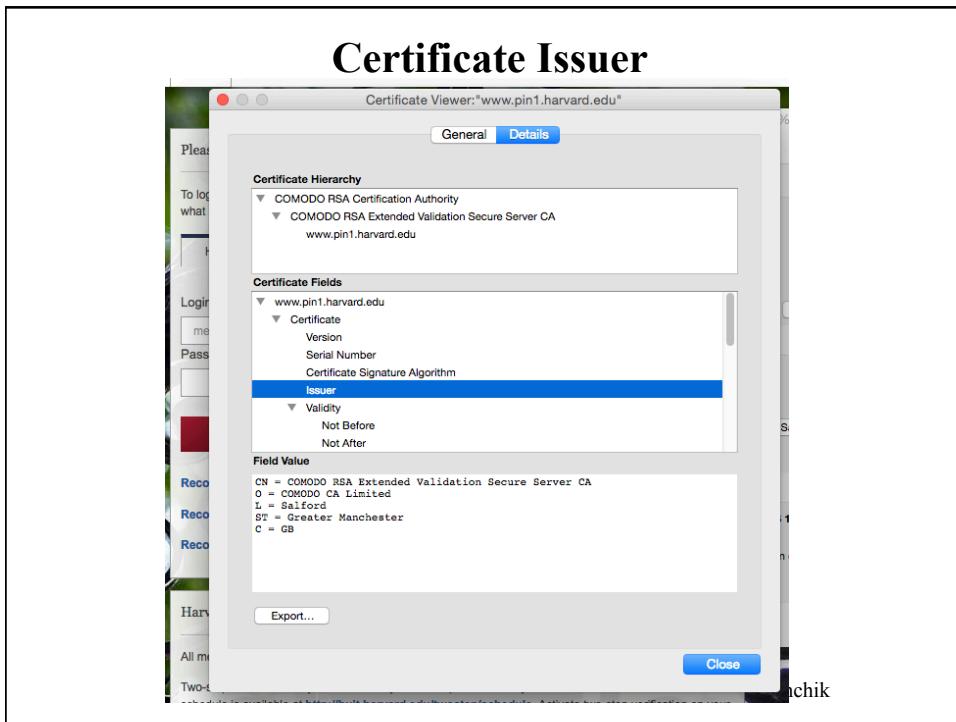
Example from Safari, other Browsers Handle it Differently



© 1998 - 2017 L. Evenchik

Web Site Security Certificate Chain of Trust

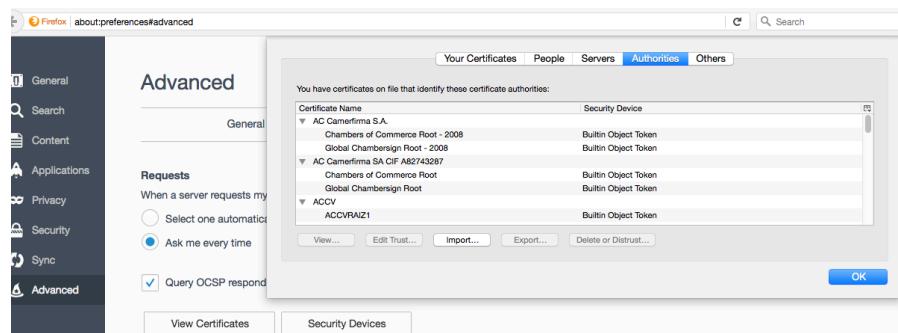
© 1998 - 2017 L. Evenchik



Browser Certificates in Firefox (1)

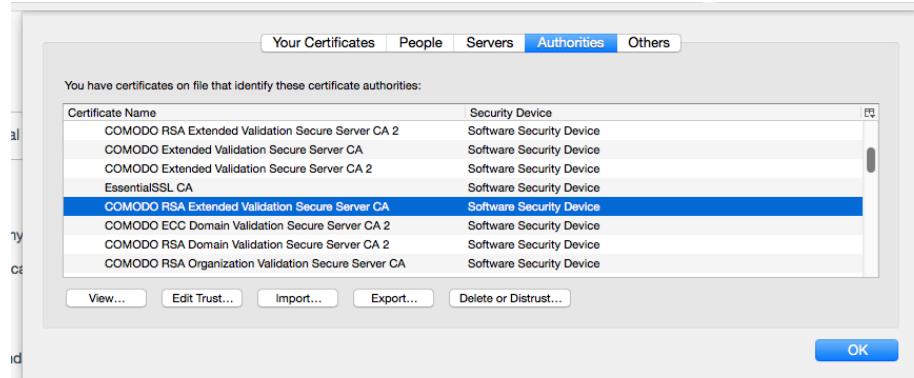
The screenshot shows the 'Advanced' section of the Firefox preferences. The left sidebar has icons for General, Search, Content, Applications, Privacy, Security, Sync, and Advanced (which is selected). The main area shows the 'Certificates' tab is active. Under 'Requests', it says 'When a server requests my personal certificate:' and has two options: 'Select one automatically' (radio button) and 'Ask me every time' (radio button, which is selected). There is also a checked checkbox for 'Query OCSP responder servers to confirm the current validity of certificates'. At the bottom are 'View Certificates' and 'Security Devices' buttons. The footer of the page says '© 1998 - 2017 L. Evenchik'.

Browser Certificates (2)



© 1998 - 2017 L. Evenchik

Browser Certificates (3)



© 1998 - 2017 L. Evenchik

Browser Certificates (4)

This certificate has been verified for the following uses:
SSL Certificate Authority

Issued To

Common Name (CN)	COMODO RSA Extended Validation Secure Server CA
Organization (O)	COMODO CA Limited
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	06:A7:43:80:D4:EB:FE:D4:35:B5:A3:F7:E1:6A:BD:D8

Issued By

Common Name (CN)	COMODO RSA Certification Authority
Organization (O)	COMODO CA Limited
Organizational Unit (OU)	<Not Part Of Certificate>

Period of Validity

Begins On	February 11, 2012
Expires On	February 11, 2027

Fingerprints

SHA-256 Fingerprint	7E:0E:16:C0:05:6F:41:A9:F4:C6:1F:57:15:03:C3:BC: F0:79:E2:BD:DB:22:8B:F2:21:9A:C3:12:00:49:6B:5C
SHA1 Fingerprint	1F:36:5C:20:E5:2A:D2:A6:B0:90:20:A0:E5:53:97:59:C9:8D:F8:D0

© 1998 - 2017 L. Evenchik

Browser Certificates (5)

Certificate Hierarchy

- ▼ COMODO RSA Certification Authority
 COMODO RSA Extended Validation Secure Server CA

Certificate Fields

- ▼ COMODO RSA Extended Validation Secure Server CA
 - ▼ Certificate
 - Version
 - Serial Number
 - Certificate Signature Algorithm
 - Issuer
 - ▼ Validity
 - Not Before
 - Not After

Field Value

CN = COMODO RSA Certification Authority
O = COMODO CA Limited
L = Salford
ST = Greater Manchester
C = GB

Export...

Browser Certificates (6)

The screenshot shows a detailed view of a certificate. At the top, there are tabs for 'General' and 'Details'. Below that is a 'Certificate Hierarchy' section showing 'COMODO RSA Certification Authority' and 'COMODO RSA Extended Validation Secure Server CA'. The main area is titled 'Certificate Fields' and includes sections for 'Issuer', 'Validity' (with 'Not Before' and 'Not After' fields), 'Subject', 'Subject Public Key Info' (with 'Subject Public Key Algorithm' and 'Subject's Public Key' which is highlighted with a blue bar), and 'Extensions'. Below this is a 'Field Value' section containing the modulus of the public key in hex format: 95 56 de 54 b4 df d5 02 49 7b d1 5b 5c a2 b2 1e 8f 9c 2b 62 4c 2b 8d 12 28 f3 1a 95 a3 c6 10 fd 29 de e1 9f 0b 38 40 93 d1 ef 6e 95 10 fc e1 90 17 77 2c ee 75 3e 7b 63 ec 61 92 6e 4f 3b ab 80 49 6b df 00 ea 03 00 7e 2f 75 d5 28 2f ec 56 67 8f 80 83 a3 bd dc 03 99 93 8b 94 91 56 5b a1 b8 6a 3a 3f 06 bd 0e 92 cc 60 9c fd b5 0f 66 30 5f db e6 94 f0 95 6a af c8 8a af 80 d9 e6 88 39 01 7a 1c c0 c5 2a f7 7b 95 a0 f2 76 ah fd 9b 72

Certificate Checking and Validation

There are a number of tools for this.

<https://www.google.com/transparencyreport/https/ct/?hl=en>

<https://www.sslshopper.com/ssl-checker.html>

Certificate Transparency

<https://www.google.com/transparencyreport/https/ct/?hl=en>

Look Up Certificates by Hostname

Look up all certificates present in public Certificate Transparency logs that have been issued for a given hostname.

Look up

Include certificates that are expired
 Include subdomains

Issuing Certificate Authorities

Issuer	# issued	Filter
C=US, O=Internet2, OU=InCommon, L=Ann Arbor, ST=MI, CN=InCommon RSA Server CA	3	Filter
C=GB, O=COMODO CA Limited, L=Salford, ST=Greater Manchester, CN=COMODO RSA Extended Validation Secure Server CA	4	Filter

Subject	Issuer	# DNS names	Valid from	Valid to	# CT logs	
www.pin1.harvard.edu	InCommon RSA Server CA	1	Feb 9, 2015	Feb 9, 2018	4	See details
www.pin1.harvard.edu	COMODO RSA Extended Validation Secure Server CA	1	Mar 22, 2016	Mar 23, 2018	3	See details
www.pin1.harvard.edu	COMODO RSA Extended Validation Secure Server CA	1	Mar 22, 2016	Mar 23, 2018	2	See details
www.pin1.harvard.edu	COMODO RSA Extended Validation Secure Server CA	5	Apr 3, 2016	Apr 4, 2018	6	See details
www.pin1.harvard.edu	COMODO RSA Extended Validation Secure Server CA	5	Apr 4, 2016	Apr 4, 2018	4	Show details

Certificate Chain of Trust

<https://www.sslshopper.com/ssl-checker.html>

SSL Checker

This SSL Checker will help you diagnose problems with your SSL certificate installation. You can verify the SSL certificate on your web server to make sure it is correctly installed, valid, trusted and doesn't give any errors to any of your users. To use the SSL Checker, simply enter your server's hostname (must be public) in the box below and click the Check SSL button. If you need an SSL certificate, check out the [SSL Wizard](#).

[More Information About the SSL Checker](#)

Check SSL

- ✓ www.pin1.harvard.edu resolves to 128.103.149.90
- ✓ The certificate should be trusted by all major web browsers (all the correct intermediate certificates are installed).
- ✓ The certificate was issued by Comodo. [Write review of Comodo](#)
- ✓ The certificate will expire in 130 days. [Remind me](#)
- ✓ The hostname (www.pin1.harvard.edu) is correctly listed in the certificate.

Certificate Chain of Trust

<https://www.sslshopper.com/ssl-checker.html>

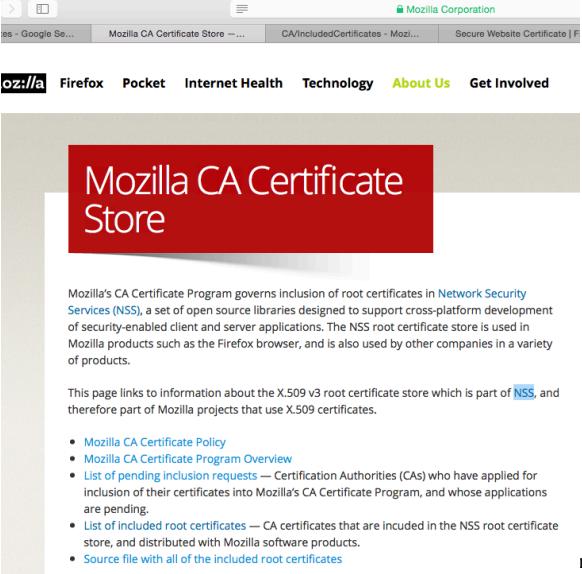


Evenchik

© 1998 - 2017 L. Evenchik

Browser Certificate List

Different Browsers and OS Store the Lists in Different Ways



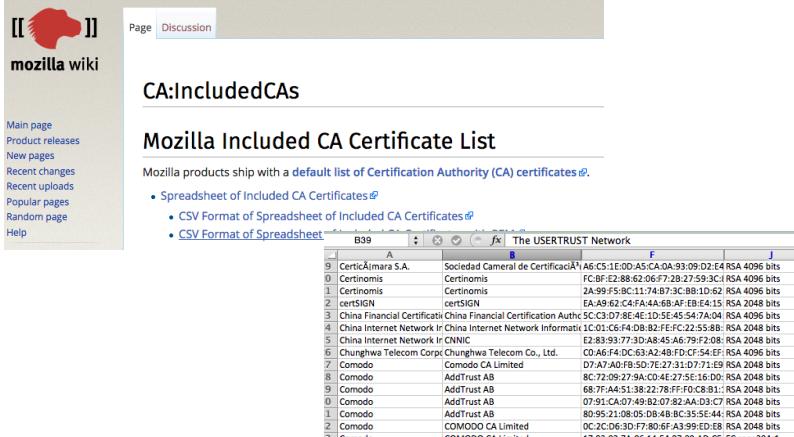
The screenshot shows a web browser window with the title "Mozilla CA Certificate Store". The page content includes a red header bar with the text "Mozilla CA Certificate Store". Below it, a paragraph explains the Mozilla CA Certificate Program's role in governing root certificate inclusion in Network Security Services (NSS). A bulleted list provides links to various program components and pending requests.

nchik

Browser Certificate List

Different Browsers and OS Store the Lists in Different Ways

https://wiki.mozilla.org/CA/Included_Certificates



The screenshot shows a Mozilla wiki page titled "CA:IncludedCAs". It features a table listing various Certification Authorities (CAs) and their corresponding certificate details, such as SHA-1 fingerprints and bit sizes. The table includes columns for the CA name, certificate type, and fingerprint.

© 1998 - 2017 L. Evenchik

CAs Do Make Errors and Have Other Problems

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

Chrome's Plan to Distrust Symantec Certificates

September 11, 2017

At the end of July, the Chrome team and the PKI community converged upon a [plan](#) to reduce, and ultimately remove, trust in Symantec's infrastructure in order to uphold users' security and privacy when browsing the web. This plan, arrived at after significant debate on the blink-dev forum, would allow reasonable time for a transition to new, independently-operated Managed Partner Infrastructure while Symantec modernizes and redesigns its infrastructure to adhere to industry standards. This post reiterates this plan and includes a timeline detailing when site operators may need to obtain new certificates.

<https://security.googleblog.com/2015/09/improved-digital-certificate-security.html>

Improved Digital Certificate Security

September 18, 2015

Posted by Stephan Somogyi, Security & Privacy PM, and Adam Eijdenberg, Certificate Transparency PM

On September 14, around 19:20 GMT, Symantec's Thawte-branded CA issued an Extended Validation (EV) pre-certificate for the domains google.com and www.google.com. This pre-certificate was neither requested nor authorized by Google.

We discovered this issuance via [Certificate Transparency](#) logs, which Chrome has required for EV certificates starting January 1st of this year. The issuance of this pre-certificate was recorded in both Google-operated and DigiCert-operated logs.

<https://wiki.mozilla.org/CA:IncludedCAs>

Mozilla Security Blog

OCT
24
2016

Distrusting New WoSign and StartCom Certificates

kwilson

Mozilla has discovered that a Certificate Authority (CA) called WoSign has had a number of technical and management failures. Most seriously, we discovered they were [backdating](#) SSL certificates in order to get around the [deadline](#) that CAs stop issuing SHA-1 SSL certificates by January 1, 2016. Additionally, Mozilla discovered that WoSign had acquired full ownership of another CA called StartCom and failed to disclose this, as required by Mozilla policy. The

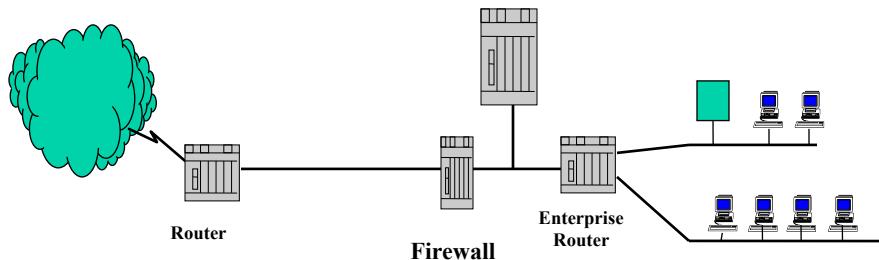
© 1998 - 2017 L. Evenchik

© 1998 - 2017 L. Evenchik

Can HTTPS Traffic Be Intercepted?

What can happen when a proxy or other device in the enterprise is trusted as a CA. Can it intercept the traffic? Could it change the contents of the traffic?

You have a homework question about this.



© 1998 - 2017 L. Evenchik

© 1998 - 2017 L. Evenchik

Virtual Private Networks and IPsec

© 1998 - 2017 L. Evenchik

Virtual Private Networks (1)

- VPNs provide an encrypted channel over an insecure link. VPNs can be implemented as part of a firewall or as stand-alone software or hardware.
- Two generic types of VPNs: tunnel mode and transport mode.
- Tunnel mode provides a secure encrypted tunnel between different sites for all the users at that site. Tunnel mode is typically done between firewalls (or other edge devices.)
- Sites can be connected together via the public internet or private network circuits (leased or owned.)
- A single access circuit can provide both VPN service and access to the public Internet at the same time.

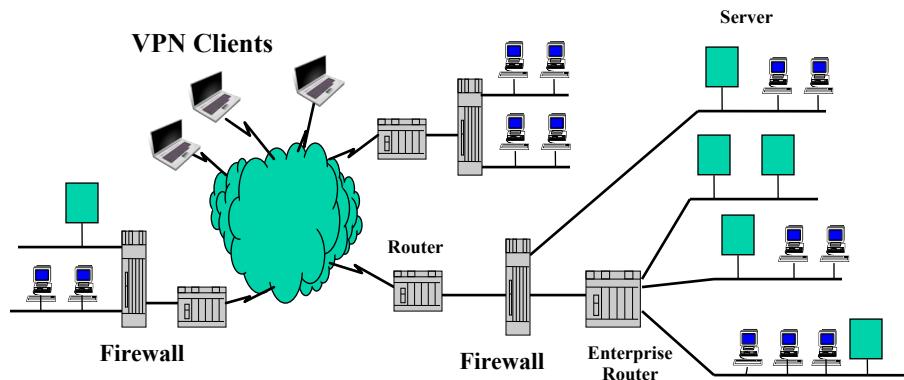
© 1998 - 2017 L. Evenchik

Virtual Private Networks (2)

- Transport mode is typically used to connect an individual user to a specific host. The encapsulated payload can be any application layer protocol (using UDP or TCP.)
- Individual users can be supported by VPN software for access via cable or xDSL and even dial-up. VPN software runs on clients and mobile devices.
- Tunnel mode can also be done between a client on a laptop or mobile devices and a firewall for access to all devices behind the firewall.
- Multiple protocols are available to set up VPNs, both proprietary and IETF protocols such as IPSec.

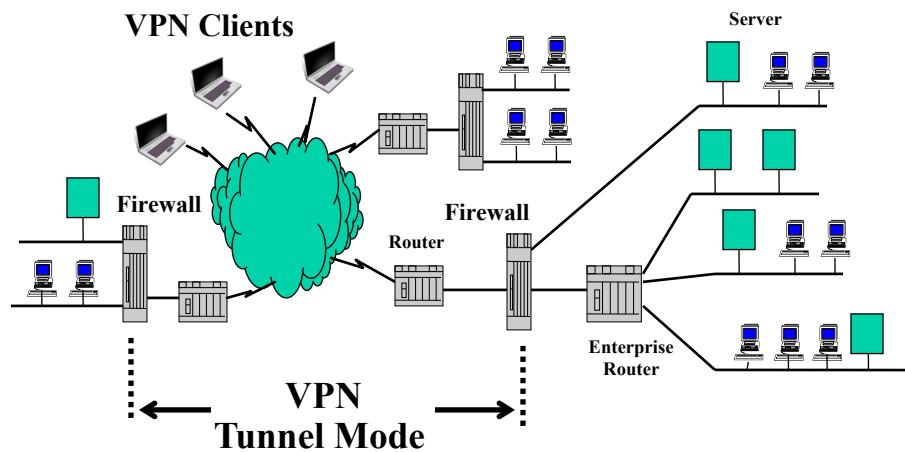
© 1998 - 2017 L. Evenchik

VPN Architecture



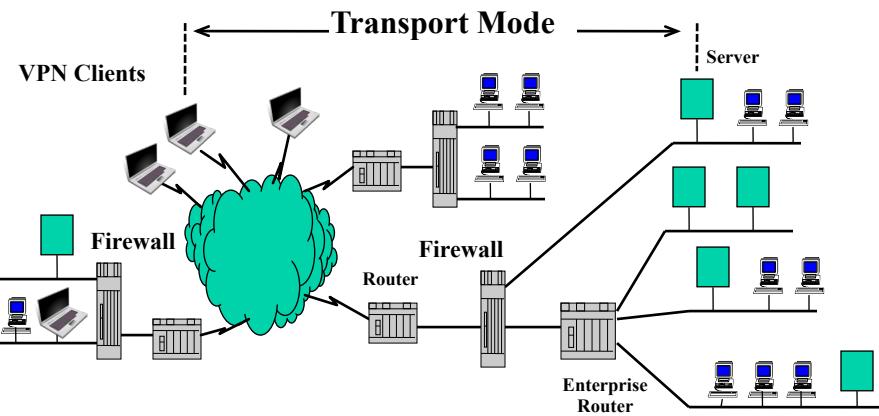
© 1998 - 2017 L. Evenchik

VPN Architecture - Tunnel Mode



© 1998 - 2017 L. Evenchik

VPN Architecture - Transport Mode



© 1998 - 2017 L. Evenchik

IPsec

- Developed by a working group of the IETF
- Provides confidentiality, integrity and authentication for the entire IP packet, or just UDP/TCP (and payload)
- Both Tunnel and Transport Mode supported, In tunnel mode, the payload encapsulates IP datagrams. In transport mode, the payload typically encapsulates TCP or UDP.
- Authentication Header (AH) - crypto checksum of contents
- Encapsulated Security Protocol (ESP) - provides confidentiality of contents plus authentication. **This is the common approach today.**
- Key management and exchange is separate (ISAKMP)
- IPv6 requires that IPSec be supported
- See RFC 4301 as your starting point.

© 1998 - 2017 L. Evenchik

AH Header

TRANSPORT MODE
AH is inserted after IP header and before any upper layer protocol

BEFORE APPLYING AH

```
-----  
IPv4 |orig IP hdr| | | |  
| (any options)| TCP | Data |  
-----
```

AFTER APPLYING AH

```
-----  
IPv4 |orig IP hdr| | | |  
| (any options)| AH | TCP | Data |  
-----  
|<---- authenticated ----->|  
except for mutable fields
```

TUNNEL MODE

Use of AH in either hosts or security gateways

```
-----  
IPv4 | new IP hdr* | | orig IP hdr* | | | |  
| (any options)| AH | (any options) |TCP | Data |  
-----  
|<- authenticated except for mutable fields -->|  
| in the new IP hdr
```

Source RFC 2402

© 1998 - 2017 L. Evenchik

ESP Header

TRANSPORT MODE

ESP is inserted after IP header and before any upper layer protocol

BEFORE APPLYING ESP

IPv4	orig IP hdr					
	(any options)	TCP		Data		

AFTER APPLYING ESP

IPv4	orig IP hdr	ESP				ESP		ESP
	(any options)	Hdr		TCP		Data		Trailer Auth

|<----- encrypted ----->|
|<----- authenticated ----->|

TUNNEL MODE

Use of AH in either hosts or security gateways

IPv4	new IP hdr*		orig IP hdr*				ESP		ESP
	(any options)	ESP		(any options)		TCP	Data	Trailer Auth	

|<----- encrypted ----->|
|<----- authenticated ----->|

Source RFC 2406

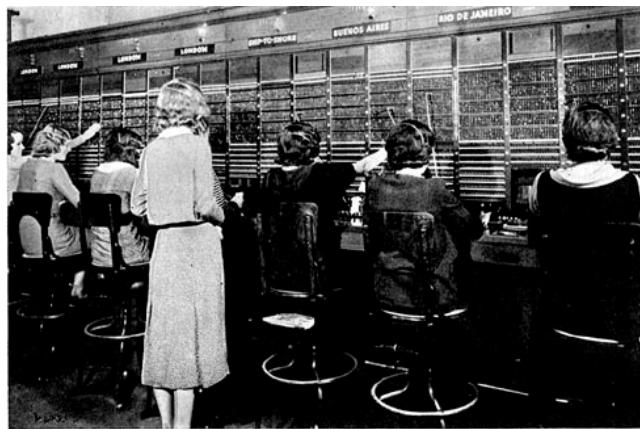
© 1998 - 2017 L. Evenchik

© 1998 - 2017 L. Evenchik

Voice and Video Over IP

© 1998 - 2017 L. Evenchik

This is Not VoIP



By courtesy of

[The American Telephone and Telegraph Co.

A LONG-DISTANCE TELEPHONE EXCHANGE.
Radio-telephone switchboard circa 1930. From the left the first four stations are
to London, the next Ship to Shore, Buenos Aires, and Rio de Janeiro.

AT&T Photo

© 1998 - 2017 L. Evenchik

Introduction

- Voice and video are both analog signals and must be converted to a digital signal (compressed and coded) for transport over packet switched networks.
- The video (or voice) transport can be one-way or full-duplex, and it can be real-time, or not real-time (i.e., streaming video.)
- Today, IP is the obvious protocol for carrying video and voice, but many other proprietary protocols have been used over the years.
- SIP is the predominant choice today for Voice over IP (VoIP) and newer videoconferencing systems. H.323 is still used for video in some corporate networks, but this is changing rapidly. Skype, now Microsoft, is a proprietary protocol. WebRTC is a browser-focused protocol approach.
- These are all protocol suites, not a single protocol.
- The transport of video or voice over packet based networks requires:
 - protocols for setting up the connection (called signaling)
 - protocols for establishing the capabilities of the end systems
 - protocols for actually sending the video and audio

© 1998 - 2017 L. Evenchik

Audio and Video Codecs (We'll talk about this more when we discuss QoS.)

- A codec converts an analog signal (either voice or video) to a digital signal (and vice versa)
- Audio Codecs
 - G.711 (8,000 samples per second, 64kbps, 30 msec sample)
 - G.722 (7Khz speech, 48kbps to 64 kbps)
 - G.723.1 (30 msec sample, 6.4kbps)
 - G.728 (16kbps, LD-CELP)
 - G.729 (8kbps, CELP)
 - Plus many proprietary and open source standards
- Video Codecs
 - H.261 (the first packet based video compression standard)
 - H.263, H.263+ and H.263++
 - H.264 (multiple versions and standards)
 - H.265 (most recent standard)
 - Plus many proprietary and open source standards

© 1998 - 2017 L. Evenchik

Introduction to RTP (1)

- Video and voice packets cannot be carried directly by UDP without additional functionality.
- The Real-time Transport Protocol is an IETF transport protocol for real time applications such as voice and video. It is standardized in RFCs 3550 and 3551.
- RTP uses UDP transport, with the inherent and limited functionality provided by UDP. This means error detection but not correction.
- RTP provides data sequencing, timing and synchronization
- RTP is augmented by a “control” protocol called RTCP (Real-Time Transport Control Protocol) which loosely monitors the flow

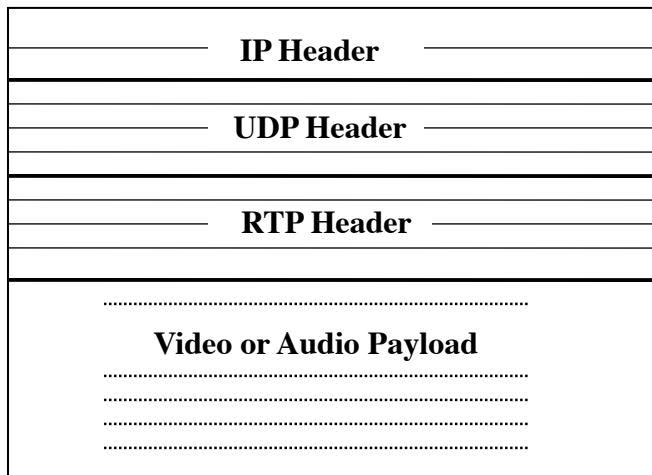
© 1998 - 2017 L. Evenchik

Introduction to RTP (2)

- RTP provides data sequencing, timing and synchronization
- RTCP provides media synchronization, feedback and forward status information
- RTP/RTCP flow uses a pair of UDP channels in each direction
- The Secure Real-time Transport Protocol (SRTP) RTP (RFC 3711) has also been defined and is used. There are other encrypted VoIP protocols such as ZRTP.

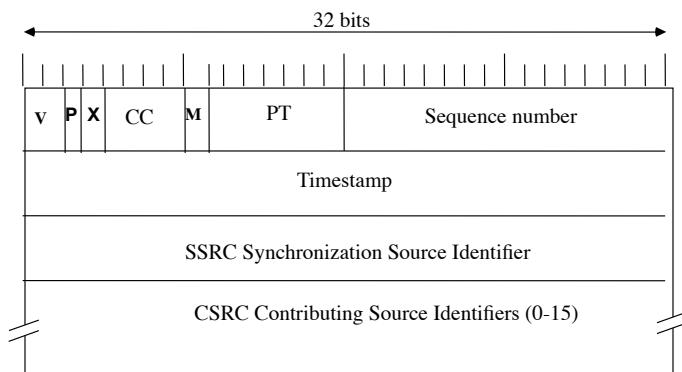
© 1998 - 2017 L. Evenchik

Combined IP/UDP/RTP Packet



© 1998 - 2017 L. Evenchik

RTP Header Format



© 1998 - 2017 L. Evenchik

RTP Header Fields

- V: version number
- P: flag to indicate padding bytes are present
- X: header extension flag
- PT: Payload Type
- CC: CSRC count
- M: marker (media dependent, defined in RTP profile)
- timestamp: sampling instant of the first byte, from media encoding clock
- SSRC: Synchronization source, the source of a single stream
- CSRC: Contributing source, a source that contributes to the combined stream produced by an RTP mixer

© 1998 - 2017 L. Evenchik

RTP Packet Flow for Video Call

© 1998 - 2017 L. Evenchik

© 1998 - 2017 L. Evenchik

SIP: Session Initiation Protocol

© 1998 - 2017 L. Evenchik

Introduction to SIP

- SIP stands for Session Initiation Protocol: This is the IETF protocol for session initiation and management. SIP does not carry voice or video packets; this is done by RTP.)
- SIP is the dominant protocol for Voice over IP (VoIP) signaling, but sessions can be many different things:
 - Telephone calls (business telephone systems)
 - Video calls
 - IM and chat traffic
 - Multimedia sessions with multiple parties
- SIP leverages other IETF and web protocols. SIP should be considered a suite of protocols.
- SIP's initial goals were simplicity and modularity, but today, it is anything but simple.

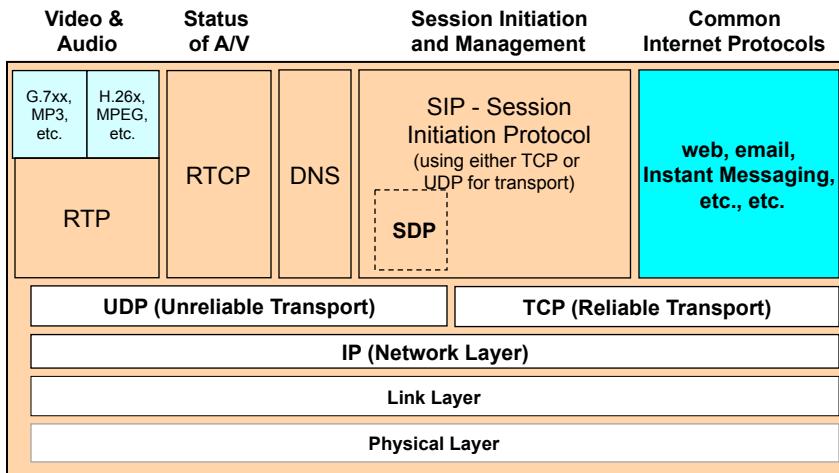
© 1998 - 2017 L. Evenchik

Protocols Required for a VoIP Call

- The transport of video or voice over packet switched networks requires protocols to support three different functions:
 - protocols for setting up the connection or session (signaling)
 - protocols for determining and deciding upon the specific voice and video capabilities and parameters that will be used by the end systems during the session (call)
 - protocols for actually sending the video and audio
- We describe these as the **three phases of the call** and it is a great way to understand voice and video protocols. However, it is a simplification in some cases given that protocols that set the capabilities can be piggybacked within the signaling phase.

© 1998 - 2017 L. Evenchik

SIP Protocol Architecture



© 1998 - 2017 L. Evenchik

Introduction to SIP (2)

- SIP is an application layer signaling protocol that looks a lot like a combination of HTTP (web) and SMTP (email).
- SIP messages are text based, comparable to email (no ASN.1 encoding is used as was done in H.323)
- SIP messages are formatted somewhat like email messages
- SIP users are addressed by a SIP URI (`sip:alice@harvard.edu`)
- Telephone numbers can also be defined and used

© 1998 - 2017 L. Evenchik

Introduction to SIP (3)

- SIP sessions and media capabilities are described by SDP, Session Description Protocol
- SIP protocol uses a Request / Response approach
- SIP message start with a Method and are followed by multiple headers
 - Methods are the actions to be performed
 - Headers contain the needed parameters and details
- SIP can use TCP, UDP or TLS as the transport protocol
- If you know H.323, it is helpful to compare SIP to H.323 (but we will not do this here.)

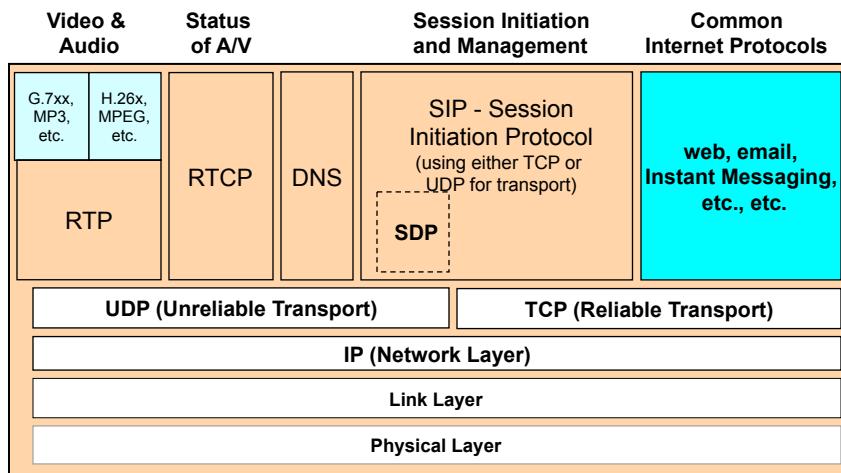
© 1998 - 2017 L. Evenchik

Primary SIP and SIP Related Protocols

- SIP: RFC 3261, core SIP protocol RFC
- SDP: RFC 4566, Session Description Protocol, describes multimedia sessions
- RFC 3263, Locating SIP Servers
- RFC 3264, an Offer/Answer model for SDP
- RTP: RFC 3550, A Transport Protocol for Real-Time Applications
- *Plus dozens of others.* We should have a hitchhikers guide for this journey, so take a look at RFC 5411.

© 1998 - 2017 L. Evenchik

SIP Protocol Architecture



© 1998 - 2017 L. Evenchik

SIP Related IETF Working Groups

- SIPCORE: Session Initiation Protocol
- CLUE: ControLling mUltiple streams for tElepresence
- Plus others including: enum, avt and drinks

© 1998 - 2017 L. Evenchik

SIP Addressing (1)

- SIP uses URI style addressing; the common way that this is explained is to say that SIP uses email style addressing, such as alice@atlanta.com
- URI stands for Uniform Resource Identifier and this approach provides a simple and extensible means for identifying a specific resource. It was introduced for use with the web.
- A URI begins with a scheme (such as http or sip), schemes are defined at:
 - <http://www.iana.org/assignments/uri-schemes.html>
- Example URIs:
 - <http://www.ietf.org/rfc/rfc2396.txt>
 - mailto:John.Doe@example.com
 - tel:+1-816-555-1212
 - sip:lensip@harvard.edu

sources rfc3986, rfc4395, rfc3261

© 1998 - 2017 L. Evenchik

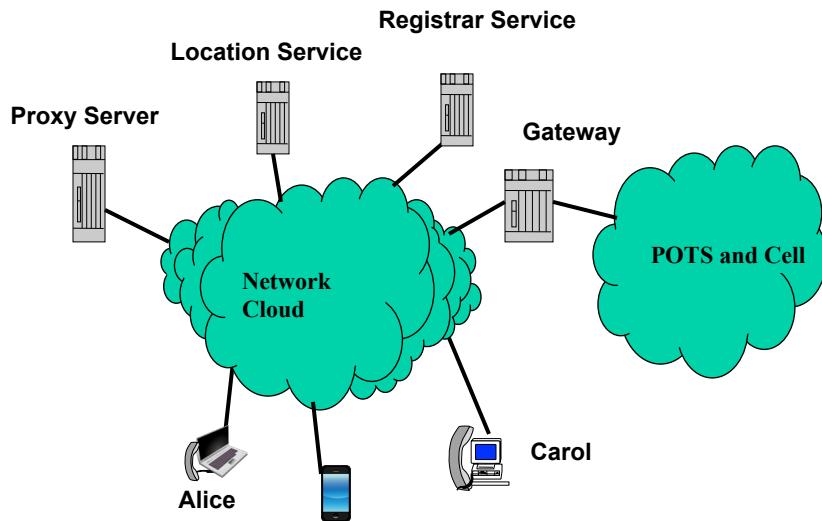
SIP Addressing (2)

- SIP or SIPS addresses identify a user, phone, computer, telephone number, service or resource.
- The general form of a SIP URI is,
`sip:user:password@host:port;uri-parameters?headers`
 - host can be an IP address, a FQDN (pc1.atlanta.com) or domain (atlanta.com)
 - uri-parameters take the form parameter-name "=" parameter-value
 - a password used in this way would not be secure
- Common form of SIP addresses:
 - sip:alice@atlanta.com
 - sip:bob@biloxi.com;transport=udp
 - sip:+1-212-555-1212;1234@gateway.com:10100;user=phone
 - sips:6100@siplearn.com:5060
 - sip:alice@18.0.2.4
- SIPS specifies a secure channel, but there are problems with this approach. There are other options such as ZRTP.
- AOR means Address of Record and it is intended to be a public SIP user address.

sources rfc3986, rfc4395, rfc3261

© 1998 - 2017 L. Evenchik

SIP Building Blocks



© 1998 - 2017 L. Evenchik

SIP Network Building Blocks

- User Agents (UA)
- Proxies
- Registrar Server
- Redirect Services/Servers
- Location Services/Servers
- Gateways
- Application Services/Servers
- Media Servers
- DNS
- Back-2-Back User Agents (B2BUA)
- Firewalls, Session Border Controllers

© 1998 - 2017 L. Evenchik

SIP User Agent

- User Agent (UA): A logical entity that can act as both a user agent client and user agent server.
- User Agent Client (UAC): A user agent client is a logical entity that creates a new session. The role of UAC lasts only for the duration of that transaction.
- User Agent Server (UAS): A user agent server is a logical entity that generates a response to a SIP request. The response accepts, rejects, or redirects the request. This role lasts only for the duration of that transaction.
- The above is a rather formal way to say that a UA such as phone can create or accept calls (sessions.)
- Consider how this relates to web clients and servers.

(source RFC 3261)

© 1998 - 2017 L. Evenchik

SIP Softphones and Hardware-based Phones (1,000s of Options Today)

Hardware based SIP Phones
and VoIP adapters



SIP Softphones are
available for all OS



We will use JITSI for some of our demos but there
are many other good options

SIP Softphones for
Android, IOS, etc



© 1998 - 2017 L. Evenchik

SIP Server and SIP Proxy

- Server: A server is a network element that receives SIP requests in order to service them and sends back SIP responses to those requests. Examples of servers are registrar servers, location servers, redirect servers, etc.
- Proxy, Proxy Server: An intermediary entity that acts as both a server and a client for the purpose of making requests on behalf of other clients.

(source RFC 3261)

© 1998 - 2017 L. Evenchik

SIP Session Support

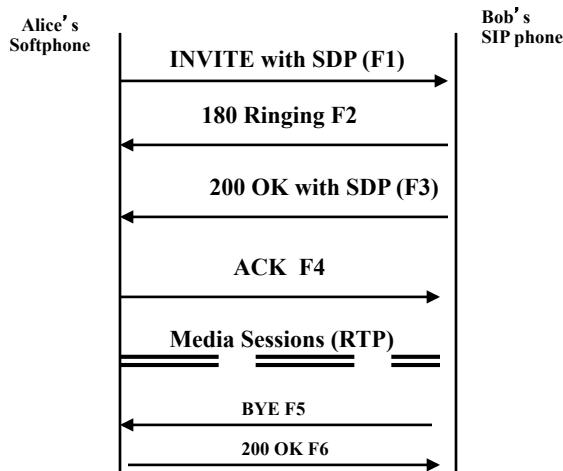
SIP supports establishing, managing and terminating multimedia sessions.

- Session setup: calling, ringing, setting session parameters
- Session management: transfer and termination, service invocation, modification of session parameters
- User location: determining the location of the end system
- User availability: willingness of the called party to engage in communications
- User capabilities: determination of the media and media parameters to be used

(source RFC 3261)

© 1998 - 2017 L. Evenchik

Point-to-Point SIP



(source RFC 3261)
F1, F2 etc are in the RFC

© 1998 - 2017 L. Evenchik

SIP Methods

- Six methods are defined in RFC 3261
 - REGISTER
 - INVITE
 - ACK
 - CANCEL
 - BYE
 - OPTIONS
- Additional methods have been defined for uses such as IM and are documented in standards track RFCs
- New methods continue to be defined and debated. The current list can be found at:
<http://www.iana.org/assignments/sip-parameters>

© 1998 - 2017 L. Evenchik

SIP INVITE Message (Simple form)

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;
      branch=z9hG4bK776asdhd
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
(Alice's SDP not shown)
```

(source RFC 3261)

© 1998 - 2017 L. Evenchik

Format of SIP Requests and Responses

- SIP is a text-based protocol and it uses a request / response approach
- SIP messages have three parts: first line (or start line), one or more header fields, and message body (optional)
- In a Request, the first line identifies the Method and includes a Request-URI, and then the protocol version
- In a Response, the first line includes the protocol version followed by a numeric status code, and then text that further explains the status code. (Remember SMTP used numeric status codes.)
- Header fields provide required information such as addresses and sequence numbers and have the form - field name : field value ; parameters (as needed)
- The message body includes additional information such as SDP data.

© 1998 - 2017 L. Evenchik

SIP Reply Codes

- 1xx: Provisional -- request received, continuing to process the request;
- 2xx: Success -- the action was successfully received, understood, and accepted;
- 3xx: Redirection -- further action needs to be taken in order to complete the request;
- 4xx: Client Error -- the request contains bad syntax or cannot be fulfilled at this server;
- 5xx: Server Error -- the server failed to fulfill an apparently valid request;
- 6xx: Global Failure -- the request cannot be fulfilled at any server.

(source RFC 3261)

© 1998 - 2017 L. Evenchik

Encapsulation of a SIP Invite

Frame (1007 bytes on wire, 1007 bytes captured)

Ethernet II, Src: 00:03:47:8f:ba:dd, Dst: 00:d0:00:db:23:fc

Internet Protocol, Source: 140.247.197.83 Destination: 195.37.77.99

User Datagram Protocol, Src Port: 5060, Dst Port: 5060

Session Initiation Protocol

Request-Line: INVITE sip:cbarkley2442@iptel.org SIP/2.0

Method: INVITE

MUCH MORE DETAIL TO FOLLOW

© 1998 - 2017 L. Evenchik

SIP Invite (part 1)

Session Initiation Protocol
Request-Line: INVITE sip:carkley2442@iptel.org SIP/2.0
Method: INVITE
Message Header
Via: SIP/2.0/UDP 140.247.197.83:5060;rport;branch=z9hG4bKDAED....
From: TestUser1 <sip:bkermitt2442@iptel.org>;tag=2672264672
 SIP Display info: TestUser1
 SIP from address: sip:bkermitt2442@iptel.org
 SIP tag: 2672264672
To: <sip:carkley2442@iptel.org>
 SIP to address: sip:carkley2442@iptel.org
Contact: <sip:bkermitt2442@140.247.197.83:5060
Call-ID: F68DE3D8-2245-4A16-A820-88752F2222AE@140.247.197.83
CSeq: 22610 INVITE
Proxy-Authorization: Digest username="bkermitt2442",realm="iptel.org",nonce="417ad0b1a61...",response="5595fd",uri="sip:carkley2442@iptel.org"
Max-Forwards: 70
Content-Type: application/sdp
User-Agent: X-Lite release 1103m
Content-Length: 302

© 1998 - 2017 L. Evenchik

SIP Invite (part 2)

Message body
Session Description Protocol
Session Description Protocol Version (v): 0
 Owner, Session Id (o): bkermitt2442 1803743 1804204
 IN IP4 140.247.197.83
 Owner Username: bkermitt2442
 Session ID: 1803743
 Session Version: 1804204
 Session Name (s): X-Lite
 Connection Information (c): IN IP4 140.247.197.83
 Connection Network Type: IN
 Connection Address Type: IP4
 Connection Address: 140.247.197.83
 Media Description, (m): audio 8000 RTP/AVP 0 8 3 98 97 101
 Media Type: audio
 Media Port: 8000
 Media Proto: RTP/AVP
 Media Format: ITU-T G.711 PCMU
 Media Format: ITU-T G.711 PCMA
 Media Format: GSM 06.10
 PLUS Others not shown here

© 1998 - 2017 L. Evenchik

SIP Invite: the bits and nothing but the bits

```
0000 00 d0 00 db 23 fc 00 03 47 8f ba dd 08 00 45 00 ....#....G.....E.
0010 03 e1 0b 53 00 00 80 11 00 00 8c f7 c5 53 c3 25 ...S.....S.% 
0020 4d 63 13 c4 13 c4 03 cd 0f e3 49 4e 56 49 54 45 Mc.....INVITE
0030 20 73 69 70 3a 63 62 61 72 6b 6c 65 79 32 34 34 sip:carkley244
0040 32 40 69 70 74 65 6c 2e 6f 72 67 20 53 49 50 2f 2@iptel.org SIP/
0050 32 2e 30 0d 0a 56 69 61 3a 20 53 49 50 2f 32 2e 2.0..Via: SIP/2.
0060 30 2f 55 44 50 20 31 34 30 2e 32 34 37 2e 31 39 0/UDP 140.247.19
0070 37 2e 38 33 3a 35 30 36 30 3b 72 70 6f 72 74 3b 7.83:5060;rport;
0080 62 72 61 6e 63 68 3d 7a 39 68 47 34 62 4b 44 41 branch=z9hG4bKDA
0090 45 44 38 35 37 46 34 45 39 44 34 34 36 36 38 44 ED857F4E9D44668D
00a0 42 46 37 42 42 44 44 34 46 31 36 33 34 31 0d 0a BF7BBDD4F16341..
00b0 46 72 6f 6d 3a 20 74 65 73 74 6f 6e 6c 61 70 74 From: testonlapt
00c0 6f 70 20 3c 73 69 70 3a 62 6b 65 72 6d 69 74 32 op <sip:bkermit2
00d0 34 34 32 40 69 70 74 65 6c 2e 6f 72 67 3e 3b 74 442@iptel.org>;t
00e0 61 67 3d 32 36 37 32 32 36 34 36 37 32 0d 0a 54 ag=2672264672..T
```

© 1998 - 2017 L. Evenchik

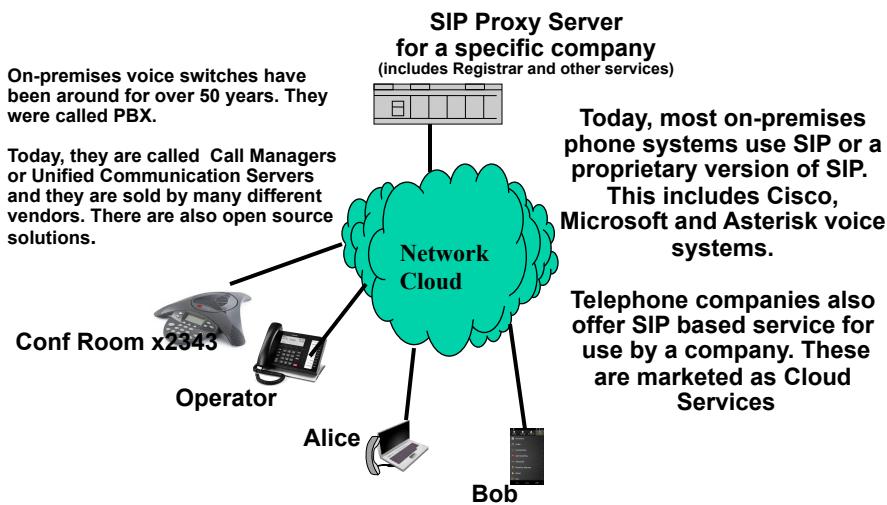
© 1998 - 2017 L. Evenchik

SIP Proxies and SIP Trapezoid

Building Real World SIP-based Networks

© 1998 - 2017 L. Evenchik

SIP Proxy Server supporting a single company



© 1998 - 2017 L. Evenchik

Asterisk is a Very Popular Open Source Solution

The screenshot shows the official Asterisk website homepage. At the top, there's a navigation bar with links for DOCS, BLOG, FORUMS, TRAINING, JOIN, and a Google Custom Search bar. Below the header, the Asterisk logo is displayed. The main content area features a large image of a man wearing glasses and a beard, holding a telephone receiver to his ear. To the left of the image, the text "Ready To Get Started With Asterisk?" is prominently displayed. Below this, a brief description states: "Asterisk is a free and open source framework for building communications applications and is sponsored by Digium." A "Watch the Video" button is located below the description. The page is divided into several sections: "Next-generation IP phones for Asterisk" (with a "See the IP Phones" button), "Asterisk is the #1 open source communications toolkit" (with a "Download Asterisk" button), and "Need a Phone System?" (with a "Get the Guide" button). The overall design is professional and user-friendly.

© 1998 - 2017 L. Evenchik

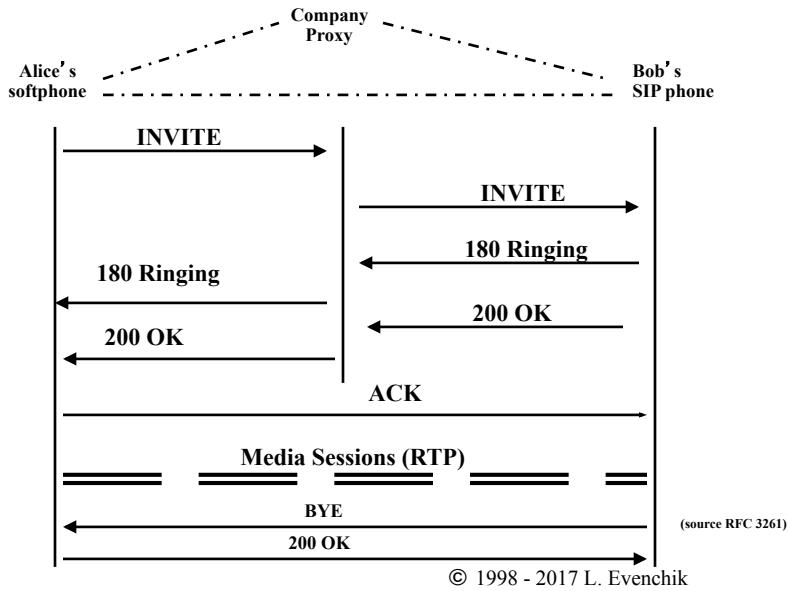
The Trend in Unified Communications (UC) for many years has been a move to Open Source and generic hardware from Proprietary hardware, software and protocols. In addition, voice services are moving to the Cloud versus premises-based solutions.

The screenshot shows the Asterisk website homepage with a focus on the trend towards unified communications. The main headline reads: "The Trend in Unified Communications (UC) for many years has been a move to Open Source and generic hardware from Proprietary hardware, software and protocols. In addition, voice services are moving to the Cloud versus premises-based solutions." To the right of the text, there are two large images of server racks: a "DEFINITY Single Carrier Cabinet" and a "DEFINITY Multi Carrier Cabinet". Below these images, the text "Old Style Proprietary Systems, 1980s" is displayed. On the left side of the page, there's a sidebar with icons for SIP Trunking, IP Phones, Telephony Cards, Software, IP PBX, and VoIP Gateway, each with a "Read More" button. The overall layout is clean and modern, contrasting with the "old style" hardware shown on the right.

Asterisk

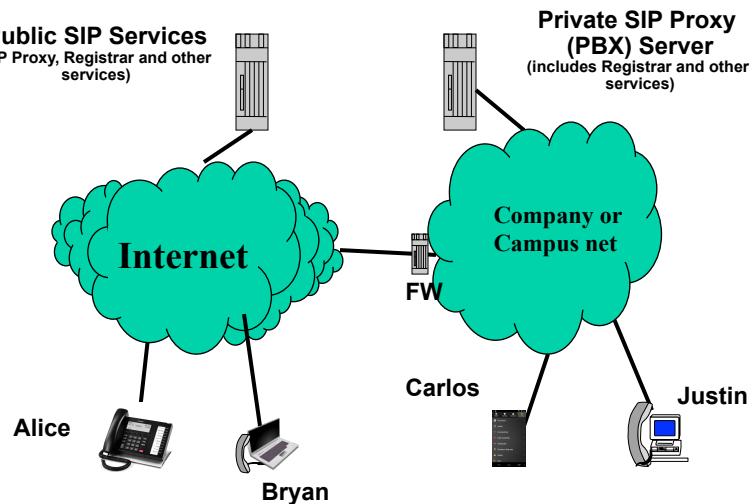
© 1998 - 2017 L. Evenchik

SIP Call via a Single Proxy

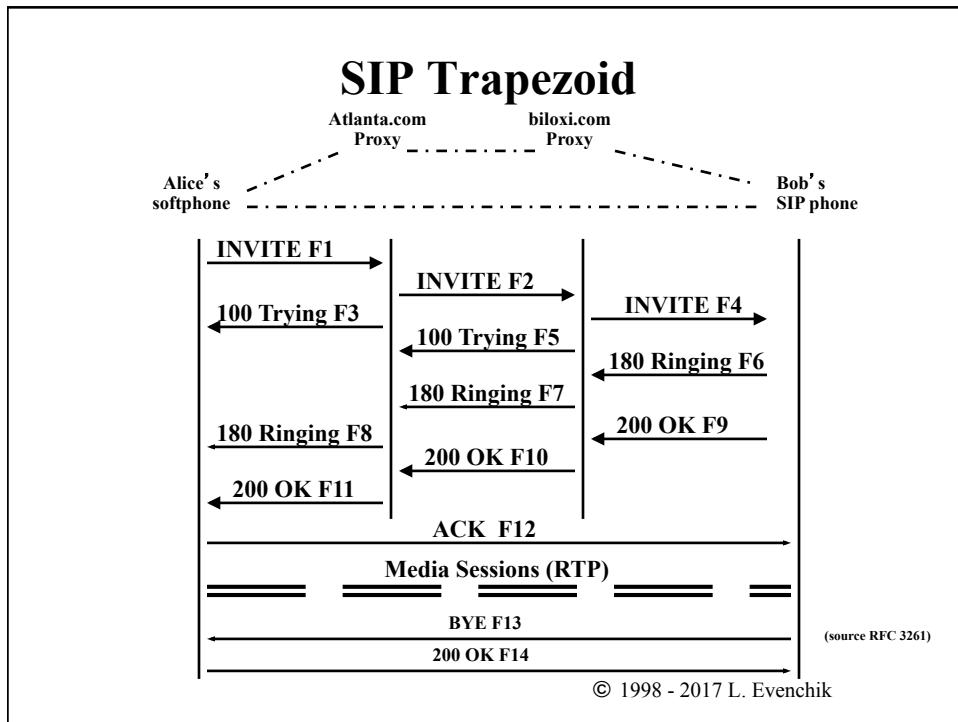


Public and Private SIP-based Networks

There are 100s of public VoIP service providers that use SIP



© 1998 - 2017 L. Evenchik



SIP Session INVITE 200 OK Response

SIP/2.0 200 OK

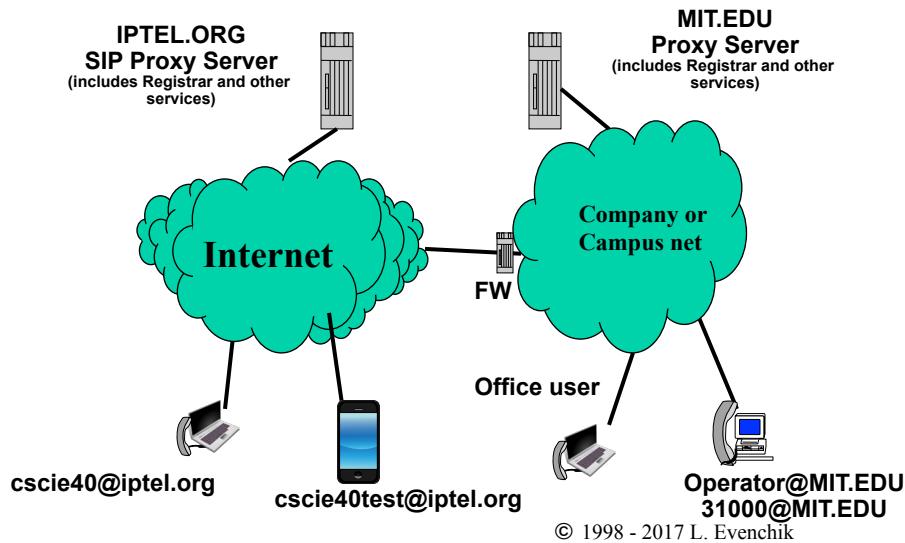
Via: SIP/2.0/UDP server10.biloxi.com
;branch=z9hG4bKnashds8;received=192.0.2.3
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com
;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP pc33.atlanta.com
;branch=z9hG4bK776asdhd ;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131
(Bob's SDP not shown)
(source RFC 3261)

© 1998 - 2017 L. Evenchik

SIP Call Demonstration

© 1998 - 2017 L. Evenchik

SIP Call Demonstration



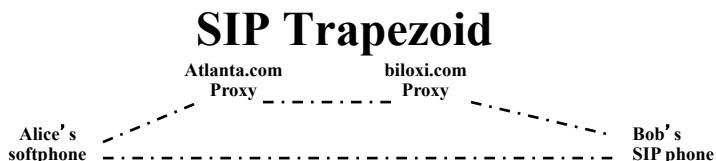
WWW.IPTEL.ORG Public SIP Service

A screenshot of a web browser showing the 'iptel.org user management' login page. The URL in the address bar is 'serweb.iptel.org/user/index.php'. The page title is 'iptel.org Userlogin'. It asks for a username and password. There is a 'language: English' dropdown and a 'Save' button. Below the form, there are links for 'Forgot Password?', 'Subscribe!', and 'Have-my-domain!'. At the bottom right of the page, there is a copyright notice '© 1998 - 2017 L. Evenchik'.

SIP Registration and User Authentication

Authentication uses the Hashing and Message Digests we learned about in the previous lecture on security.

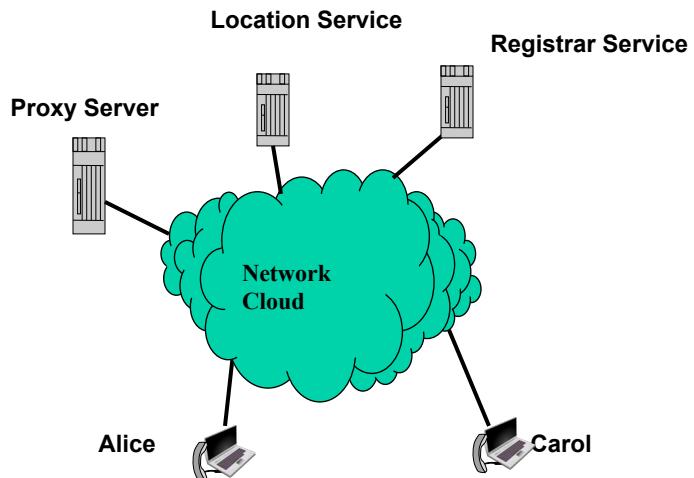
© 1998 - 2017 L. Evenchik



(source RFC 3261)

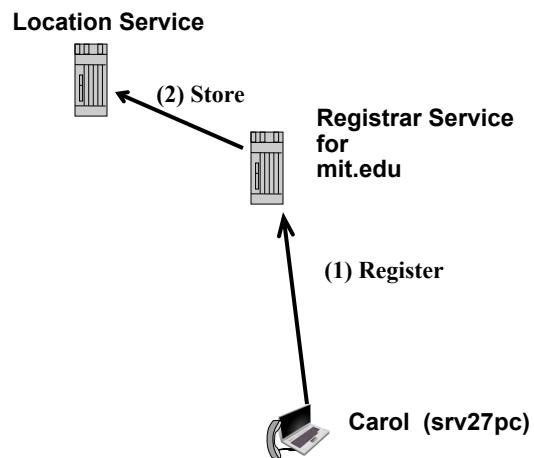
© 1998 - 2017 L. Evenchik

SIP Registration Building Blocks



© 1998 - 2017 L. Evenchik

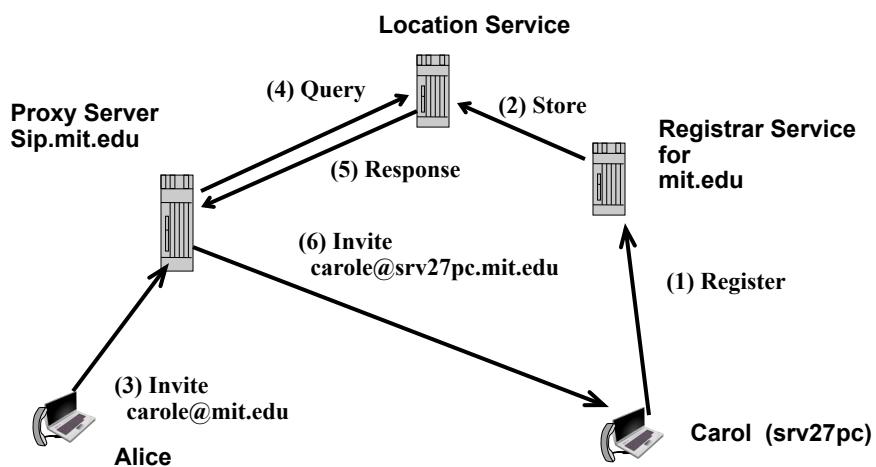
SIP Registration Carol registers in mit.edu



© 1998 - 2017 L. Evenchik

© 1998 - 2017 L. Evenchik

INVITE after SIP Registration



© 1998 - 2017 L. Evenchik

© 1998 - 2017 L. Evenchik

Domain Name System (DNS) and SIP

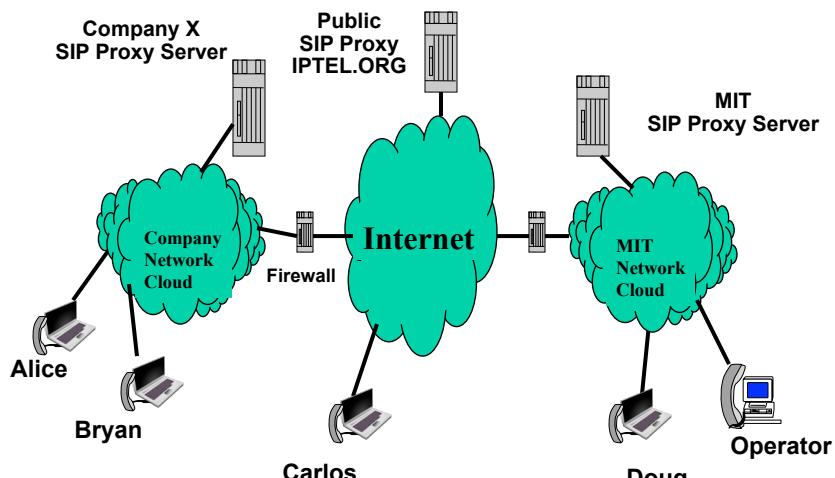
© 1998 - 2017 L. Evenchik

Finding the Proxy Server for a Remote Domain (not your own domain)

- Imagine you want to call bill@siplearn.com. How does your SIP proxy server find the IP address of the proxy that supports the siplearn.com domain.
- The approach is comparable to finding the foreign mail server for a domain.
- In other words, this will be done with DNS.

© 1998 - 2017 L. Evenchik

SIP Services Today How does the SIP proxy at Company X find the Proxy for MIT?



Justin

© 1998 - 2017 L. Evenchik

DNS Resource Records (partial listing)

- A - specifies 32 bit IPv4 address
- AAAA – IPv6 address record
- MX - mail exchange record
- NS - specifies authoritative name server for a domain
- CNAME - canonical name, provides alias functionality
- HINFO - specifies limited host information
- SRV – identifies a specific service, such as SIP Server
- NAPTR – Name Authority Pointer, points to more info

© 1998 - 2017 L. Evenchik

Finding a Network Resource via DNS

- “A” records provide a mapping between names and addresses. This is what you would expect the DNS to handle.
- But how do you find a resource such as a mail server when you don’t know the name of the server?
- For example, email to webmaster@harvard.edu must be delivered to the mail server for Harvard, even though you do not know the name (or IP address) of the server that handles incoming mail.

© 1998 - 2017 L. Evenchik

Harvard DNS MX Query

```
fas% dig harvard.edu mx
;; QUESTION SECTION:
;harvard.edu.      IN  MX

;; ANSWER SECTION:
harvard.edu.    10800  IN  MX  20 mail.br.harvard.edu.
harvard.edu.    10800  IN  MX  10 netopc.harvard.edu.
harvard.edu.    10800  IN  MX  0 netop3.harvard.edu.

;; ADDITIONAL SECTION:
mail.br.harvard.edu. 10066  IN  A   128.119.3.169
netopc.harvard.edu. 10800  IN  A   128.103.1.37
netop3.harvard.edu. 10800  IN  A   128.103.208.29

ns1.harvard.edu.    10800  IN  A   128.103.200.101
ns.harvard.edu.     10800  IN  A   128.103.201.100
ns2.harvard.edu.    10800  IN  A   128.103.1.1
```

© 1998 - 2017 L. Evenchik

Finding a SIP Proxy Server

- Finding a SIP proxy server for a specific domain is comparable to finding a mail server for a specific domain.
- SRV records configured by the administrator of the domain are used by other proxy servers on the Internet to locate the domains SIP proxy server.
- NAPTR records are also used and they provide added flexibility to the type of SRV record
- For example, a SIP call to bill@siplearn.com must be sent to the proxy server for siplearn.com domain, even though the user (or the user's proxy server) does not know the name (or IP address) of the proxy server. DNS provides this needed address information.

© 1998 - 2017 L. Evenchik

DNS SRV Query for domain SIPLEARN.COM

```
cmd% dig _sip._udp.siplearn.com SRV
;; QUESTION SECTION:
;_sip._udp.siplearn.com.      IN      SRV

;; ANSWER SECTION:
_sip._udp.siplearn.com. 3600  IN      SRV  1 1 5060 asterisk.siplearn.com.

*** Then Another DNS lookup

cmd% dig asterisk.siplearn.com
;; QUESTION SECTION:
;asterisk.siplearn.com.  IN      A

;; ANSWER SECTION:
asterisk.siplearn.com. 3600 IN      A      aa.bb.cc.dd
```

© 1998 - 2017 L. Evenchik

DNS SRV Query (Use DIG is Available)

```
nslookup
> set type=srv
> _sip._udp.siplearn.com

Answer is asterisk.siplearn.com
>
```

© 1998 - 2017 L. Evenchik

© 1998 - 2017 L. Evenchik

Session Description Protocol (SDP)

© 1998 - 2017 L. Evenchik

Example of SDP

Message body
Session Description Protocol
Session Description Protocol Version (v): 0
Owner, Session Id (o): bkermit2442 1803743 1804204
IN IP4 140.247.197.83
Owner Username: bkermit2442
Session ID: 1803743
Session Version: 1804204
Session Name (s): X-Lite
Connection Information (c): IN IP4 140.247.197.83
Connection Network Type: IN
Connection Address Type: IP4
Connection Address: 140.247.197.83
Media Description, (m): audio 12312 RTP/AVP 0 8 3 98 97 101
Media Type: audio
Media Port: 12312
Media Proto: RTP/AVP
Media Format: ITU-T G.711 PCMU
Media Format: ITU-T G.711 PCMA
Media Format: GSM 06.10
PLUS Others not shown here

© 1998 - 2017 L. Evenchik

Session Description Protocol (SDP)

- SDP is a general purpose protocol used to describe multimedia sessions. It is defined in RFC 4566.
- SDP is a format for session description, it is not a transport protocol and hence it must be carried by SIP messages.
- SDP is used to describe:
 - Session name and purpose
 - Contact and user information
 - Time information
 - Types of media to be used
 - Specific details for each media stream, including the Port #
- RFC 3264 defines offer / answer model for agreeing on media parameters.
- SDP was originally designed for describing multimedia session on the MBone test network. Some of the parameters you see in the spec relate to this history.
- SDP is being used for new protocols such as WebRTC.

© 1998 - 2017 L. Evenchik

SDP (2)

- As with SIP, SDP is text based (not ASN.1 encoded.)
- Each line of an SDP message is of the form
 $\langle\text{type}\rangle = \langle\text{value}\rangle$
 - $\langle\text{type}\rangle$ MUST be exactly one case-significant character
 - $\langle\text{value}\rangle$ is structured text whose format depends on $\langle\text{type}\rangle$
- Common Session description lines
 - v= (protocol version)
 - o= (originator and session identifier)
 - s= (session name)
 - C= (connection/address information -- not required if included in all media)
- Common Media description lines
 - m= (media name and transport address)
 - c=*(connection/address information -- optional if included at session level for all streams)
 - b=*(zero or more bandwidth information lines)
 - a=*(zero or more media attribute lines)

Plus many more

* means optional

(source RFC 4566)

© 1998 - 2017 L. Evenchik

SDP Attributes and Media lines (3)

- Attributes ("a=")
 $a=\langle\text{attribute}\rangle$
 $a=\langle\text{attribute}\rangle:\langle\text{value}\rangle$
 - Attributes are the primary means for extending SDP
- Media Descriptions ("m=")
 $m=\langle\text{media}\rangle \langle\text{port}\rangle \langle\text{proto}\rangle \langle\text{fmt}\rangle \dots$
 - $\langle\text{media}\rangle$ is the media type
 - Media includes "audio", "video", "text", "application", and "message"
 - $\langle\text{port}\rangle$ is the transport port to which the media stream is sent
 - $\langle\text{fmt}\rangle$ is a media format description.
- We also need to look at MIME types

(source RFC 4566)

© 1998 - 2017 L. Evenchik

SDP for H.264 per Internet Draft (abridged)

- SDP for codecs such as H.264 can have a large number of attributes

Offerer -> Answerer SDP message:

```
m=video 49170 RTP/AVP 100 99 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; packetization-mode=0;
  sprop-parameter-sets=<parameter sets data#0>
a=rtpmap:99 H264/90000
a=fmtp:99 profile-level-id=42A01E; packetization-mode=1;
  sprop-parameter-sets=<parameter sets data#1>
a=rtpmap:100 H264/90000
a=fmtp:100 profile-level-id=42A01E; packetization-mode=2;
  sprop-parameter-sets=<parameter sets data#2>;
  sprop-interleaving-depth=45; sprop-deint-buf-req=64000;
  sprop-init-buf-time=102478; deint-buf-cap=128000
```

© 1998 - 2017 L. Evenchik

© 1998 - 2017 L. Evenchik

SIP Offer/Answer

© 1998 - 2017 L. Evenchik

Offer/Answer Example 1 (What is the outcome?)

OFFER

```
v=0
o=alice 2890844526 2890844526
IN IP4 host.atlanta.example.com
c=IN IP4 host.atlanta.example.com
m=audio 49170 RTP/AVP 0 8 97
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
m=video 51372 RTP/AVP 31 132
a=rtpmap:31 H261/90000
a=rtpmap:132 H264/90000
```

ANSWER

```
v=0
o=bob 2808844564 2808844564
IN IP4 host.biloxi.example.com
c=IN IP4 host.biloxi.example.com
m=audio 49174 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
m=video 53456 RTP/AVP 132
a=rtpmap:132 H264/90000
```

(source RFC 4317)
© 1998 - 2017 L. Evenchik

Offer/Answer Example 2 (What is the outcome?)

OFFER

```
v=0
o=alice 2890844526 2890844526 IN
    IP4 host.atlanta.example.com
c=IN IP4 host.atlanta.example.com
m=audio 49170 RTP/AVP 0 8 97
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 ilBC/8000
m=video 51372 RTP/AVP 31 32
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
```

ANSWER

```
v=0
o=bob 2808844564 2808844564
    IN IP4 host.biloxi.example.com
c=IN IP4 host.biloxi.example.com
m=audio 49172 RTP/AVP 0 8
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
m=video 0 RTP/AVP 31
a=rtpmap:31 H261/90000
```

(source RFC 4317)

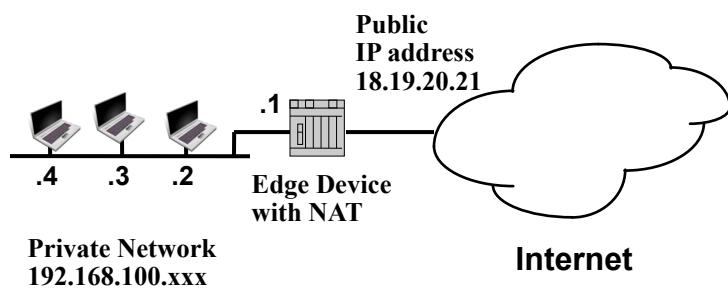
© 1998 - 2017 L. Evenchik

© 1998 - 2017 L. Evenchik

SIP and NAT and Firewalls

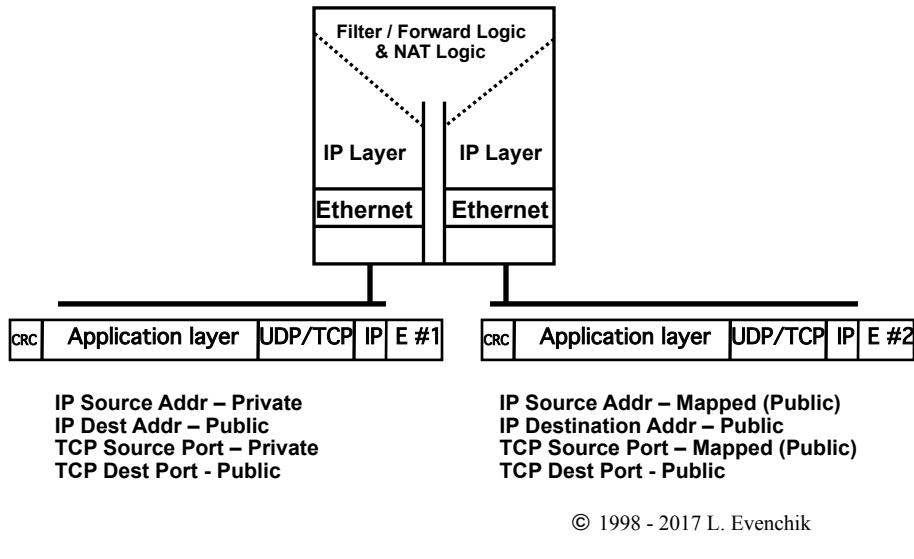
© 1998 - 2017 L. Evenchik

Network Address Port Translation (NAPT) Block Diagram

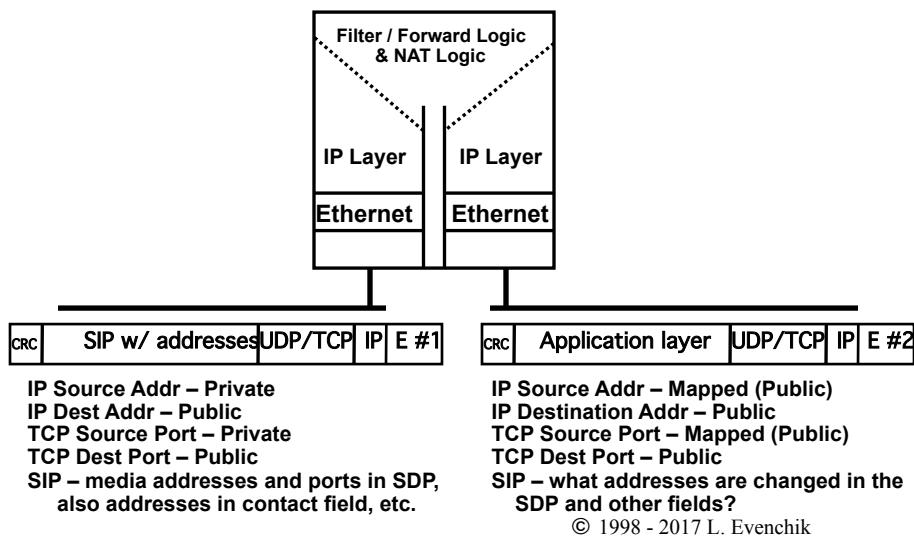


© 1998 - 2017 L. Evenchik

NAT and Port-Mapping Functionality Implemented by a Router or Firewall



NAT and Port-Mapping with Embedded Addresses in Application Layer



SIP Use of STUN, TURN and ICE

- SIP clients and proxies use STUN, TURN and ICE to overcome the problems caused by NAT. Proprietary protocols are also used by many systems.
- STUN – protocol used by a client to determine the presence and type of NAT
- TURN – protocol for working with a media relay located on the Internet. A TURN relay replaces the need for an inbound call through at NAT. All the clients place outbound calls to the TURN server instead.
- ICE – complex protocol for managing NAT traversal in protocols such as SIP (for VoIP) that use the offer/answer model.

© 1998 - 2017 L. Evenchik

One Minute Wrap-Up

- Please do this Wrap-Up at the end of each lecture.
- Please fill out the form on the website.
- The form is anonymous (but you can include your name if you want.)
- Please answer three questions:
 - What is your grand “Aha” for today’s class?
 - What concept did you find most confusing in today’s class?
 - What questions should I address next time
- **Thank you!**

© 1998 - 2017 L. Evenchik