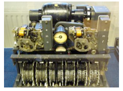


Network Security Protocols

Mike Freedman

COS 461: Computer Networks

Lectures: MW 10-10:50am in Architecture N101

http://www.cs.princeton.edu/courses/archive/spr13/cos461/

Network Security

- Application layer
  - E-mail: PGP, using a web-of-trust
  - Web: HTTP-S, using a certificate hierarchy
- Transport layer
  - Transport Layer Security/ Secure Socket Layer
- Network layer
  - IP Sec
- Network infrastructure
  - DNS-Sec and BGP-Sec

Basic Security Properties

- **Confidentiality:**
- **Authenticity:**
- **Integrity:**
- **Availability:**
- **Non-repudiation:**
- **Access control:**

Basic Security Properties

- **Confidentiality:** Concealment of information or resources
- **Authenticity:** Identification and assurance of origin of info
- **Integrity:** Trustworthiness of data or resources in terms of preventing improper and unauthorized changes
- **Availability:** Ability to use desired information or resource
- **Non-repudiation:** Offer of evidence that a party indeed is sender or a receiver of certain information
- **Access control:** Facilities to determine and enforce who is allowed access to what resources (host, software, network, …)

Encryption and MAC/Signatures

Confidentiality (Encryption)

Sender:

- Compute C = $\text{Enc}_K(M)$
- Send C

Receiver:

- Recover M = $\text{Dec}_K(C)$

Auth/Integrity (MAC / Signature)

Sender:

- Compute s = $\text{Sig}_K(\text{Hash }(M))$
- Send <M, s>

Receiver:

- Compute s' = $\text{Ver}_K(\text{Hash }(M))$
- Check s' == s

These are simplified forms of the actual algorithms

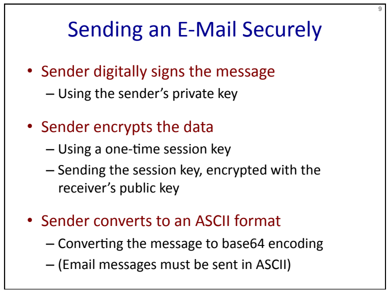
Email Security:
Pretty Good Privacy (PGP)

E-Mail Security

- Security goals
  - Confidentiality: only intended recipient sees data
  - Integrity: data cannot be modified en route
  - Authenticity: sender and recipient are who they say
- Security non-goals
  - Timely or successful message delivery
  - Avoiding duplicate (replayed) message
  - (Since e-mail doesn't provide this anyway!)

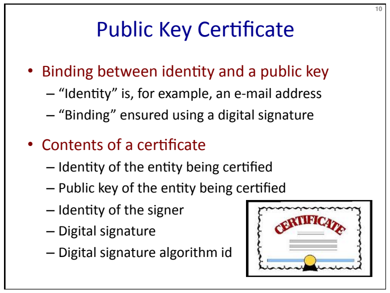
Sender and Receiver Keys

- If the sender knows the receiver's public key
  - Confidentiality
  - Receiver authentication
- If the receiver knows the sender's public key
  - Sender authentication
  - Sender non-repudiation



Sending an E-Mail Securely

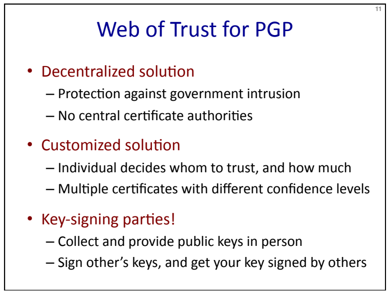
- Sender digitally signs the message
  - Using the sender’s private key
- Sender encrypts the data
  - Using a one-time session key
  - Sending the session key, encrypted with the receiver’s public key
- Sender converts to an ASCII format
  - Converting the message to base64 encoding
  - (Email messages must be sent in ASCII)



Public Key Certificate

- Binding between identity and a public key
  - “Identity” is, for example, an e-mail address
  - “Binding” ensured using a digital signature
- Contents of a certificate
  - Identity of the entity being certified
  - Public key of the entity being certified
  - Identity of the signer
  - Digital signature
  - Digital signature algorithm id

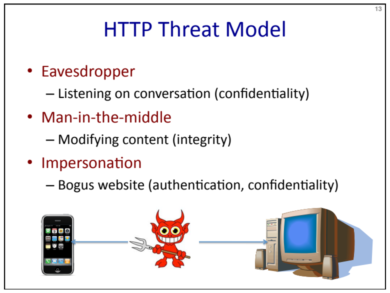
CERTIFICATE



Web of Trust for PGP

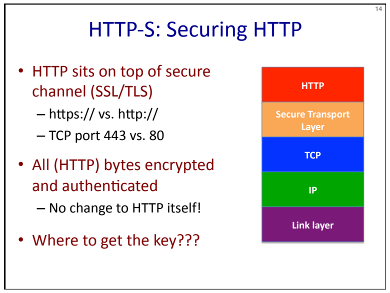
- Decentralized solution
  - Protection against government intrusion
  - No central certificate authorities
- Customized solution
  - Individual decides whom to trust, and how much
  - Multiple certificates with different confidence levels
- Key-signing parties!
  - Collect and provide public keys in person
  - Sign other’s keys, and get your key signed by others

HTTP Security



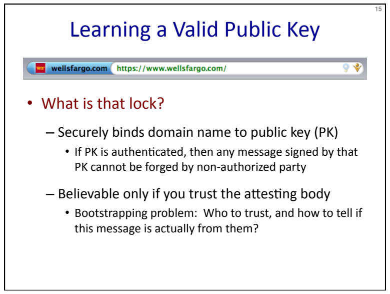
HTTP Threat Model

- Eavesdropper
  - Listening on conversation (confidentiality)
- Man-in-the-middle
  - Modifying content (integrity)
- Impersonation
  - Bogus website (authentication, confidentiality)



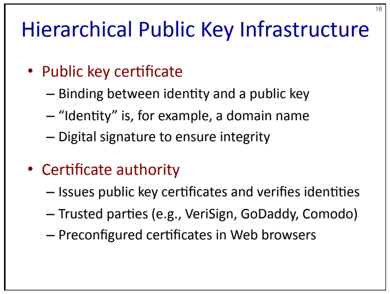
HTTP-S: Securing HTTP

- HTTP sits on top of secure channel (SSL/TLS)
  - https:// vs. http://
  - TCP port 443 vs. 80
- All (HTTP) bytes encrypted and authenticated
  - No change to HTTP itself!
- Where to get the key???



Learning a Valid Public Key

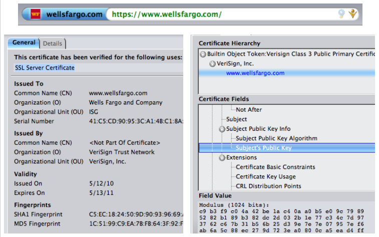
- What is that lock?
  - Securely binds domain name to public key (PK)
    - If PK is authenticated, then any message signed by that PK cannot be forged by non-authorized party
  - Believable only if you trust the attesting body
    - Bootstrapping problem: Who to trust, and how to tell if this message is actually from them?



Hierarchical Public Key Infrastructure

- Public key certificate
  - Binding between identity and a public key
  - “Identity” is, for example, a domain name
  - Digital signature to ensure integrity
- Certificate authority
  - Issues public key certificates and verifies identities
  - Trusted parties (e.g., VeriSign, GoDaddy, Comodo)
  - Preconfigured certificates in Web browsers

Public Key Certificate



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

Issued To
Common Name (CN) www.wellsfargo.com
Organization (O) Wells Fargo and Company
Organizational Unit (OU) ISG
Serial Number 41:C5:CD:90:95:3C:A1:4B:C1:8A:

Issued By
Common Name (CN) <Not Part Of Certificate>
Organization (O) VeriSign Trust Network
Organizational Unit (OU) VeriSign, Inc.

Validity
Issued On 5/12/10
Expires On 5/13/11

Fingerprints
SHA1 Fingerprint C5:EC:18:24:50:9D:90:93:96:69:
MD5 Fingerprint 1C:51:99:C9:EA:7B:FB:64:3F:92:F

Certificate Hierarchy
Builtin Object Token:Verisign Class 3 Public Primary Certific
VeriSign, Inc.
www.wellsfargo.com

Certificate Fields
Not After
Subject
Subject Public Key Info
Subject Public Key Algorithm
Subject's Public Key
Extensions
Certificate Basic Constraints
Certificate Key Usage
CRL Distribution Points

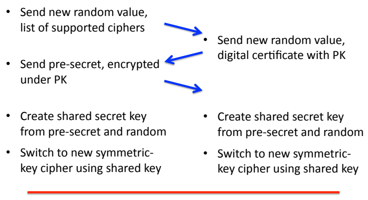
Field Value

```
Modulus (1024 bits):
c9 b3 f9 c0 4a 42 be 1a c4 0a a0 b5 e0 9c 79 89
52 82 b1 89 b3 82 dc 2d 03 2b 1e 77 c3 4c 7d 97
37 62 c6 7b 31 b5 6b 25 d3 9e 7e 7e 07 95 7e f6
ab 6a 5c 88 ec 27 9d 72 3e a0 80 0c a5 ea d4 ff
```

Transport Layer Security (TLS)

Based on the earlier Secure Socket Layer (SSL) originally developed by Netscape

TLS Handshake Protocol



- Send new random value, list of supported ciphers
- Send new random value, digital certificate with PK
- Send pre-secret, encrypted under PK
- Create shared secret key from pre-secret and random
- Create shared secret key from pre-secret and random
- Switch to new symmetric-key cipher using shared key
- Switch to new symmetric-key cipher using shared key

TLS Record Protocol

- Messages from application layer are:
  - Fragmented or coalesced into blocks
  - Optionally compressed
  - Integrity-protected using an HMAC
  - Encrypted using symmetric-key cipher
  - Passed to the transport layer (usually TCP)
- Sequence #s on record-protocol messages
  - Prevents replays and reorderings of messages

Comments on HTTPS

- HTTPS authenticates server, not content
  - If CDN (Akamai) serves content over HTTPS, customer must trust Akamai not to change content
- Symmetric-key crypto after public-key ops
  - Handshake protocol using public key crypto
  - Symmetric-key crypto much faster (100-1000x)
- HTTPS on top of TCP, so reliable byte stream
  - Can leverage fact that transmission is reliable to ensure: each data segment received exactly once
  - Adversary can't successfully drop or replay packets

IP Security

IP Security

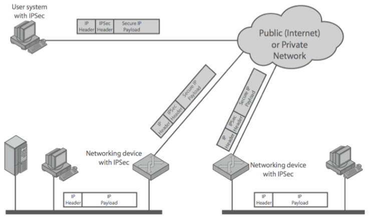
- There are range of app-specific security mechanisms
  - eg. TLS/HTTPS, S/MIME, PGP, Kerberos, …
- But security concerns that cut across protocol layers
- Implement by the network for all applications?

Enter IPSec!

IPSec

- General IP Security framework
- Allows one to provide
  - Access control, integrity, authentication, originality, and confidentiality
- Applicable to different settings
  - Narrow streams: Specific TCP connections
  - Wide streams: All packets between two gateways

IPSec Uses


User system with IPSec
IP Header
IPSec Header
Secure IP Payload
Public (Internet) or Private Network
Networking device with IPSec
IP Header
IP Payload
Networking device with IPSec

Benefits of IPSec

- If in a firewall/router:
  - Strong security to all traffic crossing perimeter
  - Resistant to bypass
- Below transport layer
  - Transparent to applications
  - Can be transparent to end users
- Can provide security for individual users

IP Security Architecture

- Specification quite complex
  - Mandatory in IPv6, optional in IPv4
- Two security header extensions:
  - Authentication Header (AH)
    - Connectionless integrity, origin authentication
      - MAC over most header fields and packet body
    - Anti-replay protection
  - Encapsulating Security Payload (ESP)
    - These properties, plus confidentiality

Encapsulating Security Payload (ESP)

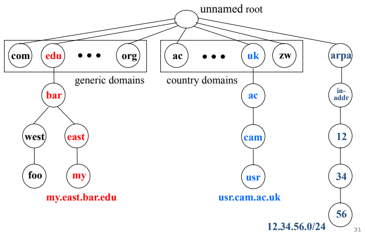
- Transport mode: Data encrypted, but not header
  - After all, network headers needed for routing!
  - Can still do traffic analysis, but is efficient
  - Good for host-to-host traffic
- Tunnel mode: Encrypts entire IP packet
  - Add new header for next hop
  - Good for VPNs, gateway-to-gateway security

Replay Protection is Hard

- Goal: Eavesdropper can't capture encrypted packet and duplicate later
  - Easy with TLS/HTTP on TCP: Reliable byte stream
  - But IP Sec at packet layer; transport may not be reliable
- IP Sec solution: Sliding window on sequence #'s
  - All IPSec packets have a 64-bit monotonic sequence number
  - Receiver keeps track of which seqno's seen before
    - [lastest – windowsize + 1 , latest] ; windowsize typically 64 packets
  - Accept packet if
    - seqno > latest (and update latest)
    - Within window but has not been seen before
  - If reliable, could just remember last, and accept iff last + 1

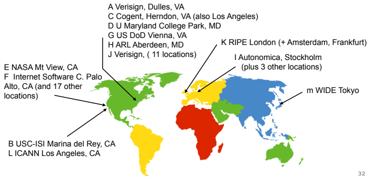
DNS Security

Hierarchical Naming in DNS



DNS Root Servers

- 13 root servers (see http://www.root-servers.org/)
- Labeled A through M



A Verisign, Dulles, VA
C Cogent, Herndon, VA (also Los Angeles)
D U Maryland College Park, MD
G US DoD Vienna, VA
H ARL Aberdeen, MD
J Verisign, (11 locations)

K RIPE London (+ Amsterdam, Frankfurt)

I Autonomica, Stockholm
(plus 3 other locations)

E NASA Mt View, CA
F Internet Software C. Palo Alto, CA (and 17 other locations)

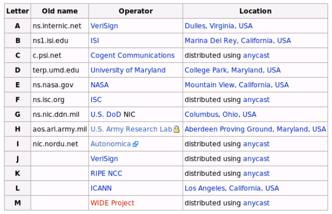
m WIDE Tokyo

B USC-ISI Marina del Rey, CA
L ICANN Los Angeles, CA

DoS attacks on DNS Availability

- Feb. 6, 2007
  - Botnet attack on the 13 Internet DNS root servers
  - Lasted 2.5 hours
  - None crashed, but two performed badly:
    - g-root (DoD), l-root (ICANN)
    - Most other root servers use anycast

Defense: Replication and Caching

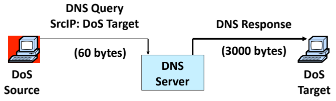


Letter	Old name	Operator	Location
A	ns.internic.net	VeriSign	Dulles, Virginia, USA
B	ns1.isi.edu	ISI	Marina Del Rey, California, USA
C	c.psi.net	Cogent Communications	distributed using anycast
D	terp.umd.edu	University of Maryland	College Park, Maryland, USA
E	ns.nasa.gov	NASA	Mountain View, California, USA
F	ns.isc.org	ISC	distributed using anycast
G	ns.nic.ddn.mil	U.S. DoD NIC	Columbus, Ohio, USA
H	aos.arl.army.mil	U.S. Army Research Lab	Aberdeen Proving Ground, Maryland, USA
I	nic.nordu.net	Autonomica	distributed using anycast
J		VeriSign	distributed using anycast
K		RIPE NCC	distributed using anycast
L		ICANN	Los Angeles, California, USA
M		WIDE Project	distributed using anycast

source: wikipedia

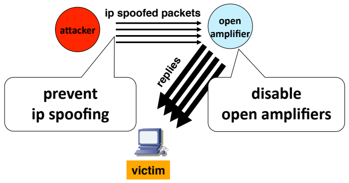
Denial-of-Service Attacks on Hosts

×40 amplification



580,000 open resolvers on Internet (Kaminsky-Shiffman'06)

Preventing Amplification Attacks



DNS Integrity and the TLD Operators

- If domain name doesn't exist, DNS should return NXDOMAIN (non-existant domain) msg
- Verisign instead creates wildcard records for all .com and .net names not yet registered
  - September 15 – October 4, 2003
- Redirection for these domain names to Verisign web portal: "to help you search"
  - And serve you ads…and get "sponsored" search
  - Verisign and online advertising companies make $$

DNS Integrity: Cache Poisoning

- Was answer from an authoritative server?
  - Or from somebody else?
- DNS cache poisoning
  - Client asks for www.evil.com
  - Nameserver authoritative for www.evil.com returns additional section for (www.cnn.com, 1.2.3.4, A)
  - Thanks! I won't bother check what I asked for

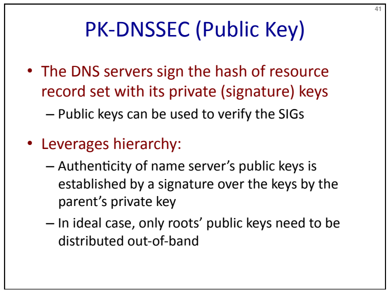
DNS Integrity: DNS Hijacking

- To prevent cache poisoning, client remembers:
  - The domain name in the request
  - A 16-bit request ID (used to demux UDP response)
- DNS hijacking
  - 16 bits: 65K possible IDs
  - What rate to enumerate all in 1 sec? 64B/packet
  - 64*65536*8 / 1024 / 1024 = 32 Mbps
- Prevention: also randomize DNS source port
  - Kaminsky attack: this source port… wasn't random

http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html

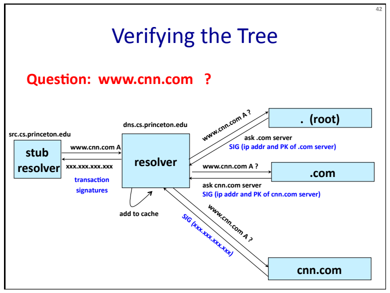
Let's strongly believe the answer! Enter DNSSEC

- DNSSEC protects against data spoofing and corruption
- DNSSEC also provides mechanisms to authenticate servers and requests
- DNSSEC provides mechanisms to establish authenticity and integrity



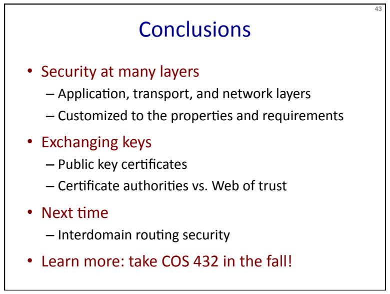
PK-DNSSEC (Public Key)

- The DNS servers sign the hash of resource record set with its private (signature) keys
  - Public keys can be used to verify the SIGs
- Leverages hierarchy:
  - Authenticity of name server's public keys is established by a signature over the keys by the parent's private key
  - In ideal case, only roots' public keys need to be distributed out-of-band



Verifying the Tree

Question: www.cnn.com ?



Conclusions

- Security at many layers
  - Application, transport, and network layers
  - Customized to the properties and requirements
- Exchanging keys
  - Public key certificates
  - Certificate authorities vs. Web of trust
- Next time
  - Interdomain routing security
- Learn more: take COS 432 in the fall!