KNC 401
COMPUTER SYSTEM SECURITY

Module 8.1
UNIT-IV-PartII

By
Dr. Alok Katiyar
Email:alok.katiyar@ipec.org.in

Computer System Security (KNC401)

Unit-IV-Part-II

Public Key Distribution
Real World Protocol

Alok Katiyar
CSE Department, IPEC

Public Key Distribution

Public Key Distribution

- Every user has his/her own public key and private key.
- Public keys are all published in a database.
- Sender and receiver agree on a cryptosystem.
- Sender gets receiver's public key from the db.
- Sender encrypts the message and sends it.
- Receiver decrypts it using his/her private key.
- What can be a problem?

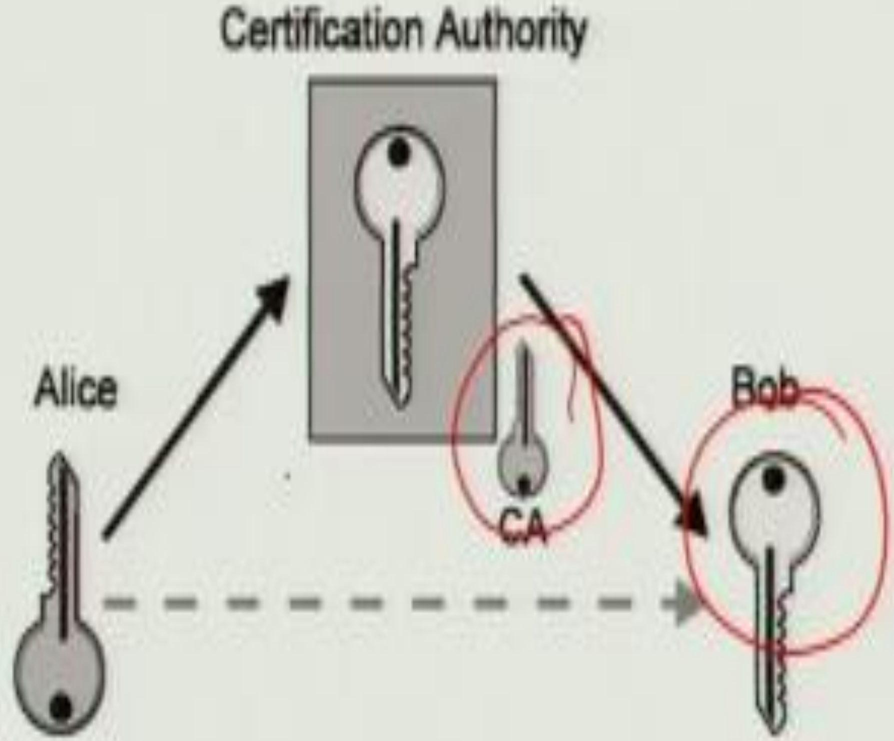
Matching keys to owners

- Insecurity of TCP/IP
  - No authentication
  - No privacy/confidentiality
  - Repudiation possible
- Public key cryptography not enough
- Need to match keys to owners
- Need *infrastructure* and *certificate authorities*

Public Key Infrastructure (PKI)

- As defined by Netscape:
  - *"Public-key infrastructure (PKI) is the combination of software, encryption technologies, and services that enables enterprises to protect the security of their communications and business transactions on the Internet."*
  - Integrates digital certificates, public key cryptography, and certification authorities
- Two major frameworks
  - X.509
  - PGP (Pretty Good Privacy)

Certification Authorities (CAs)



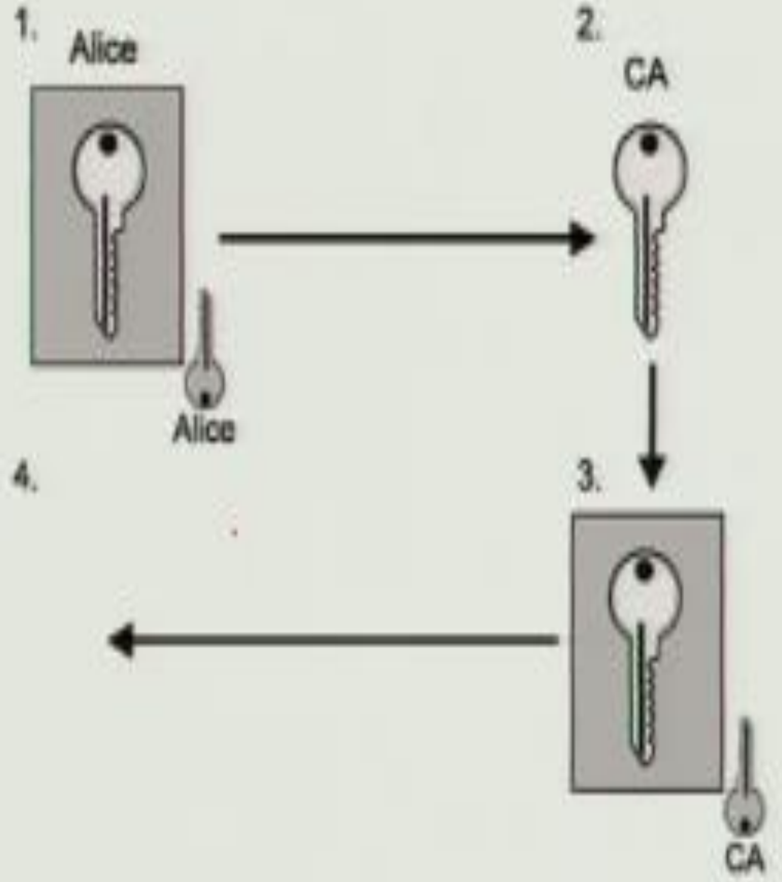
Certification Authorities (cont.)

- Guarantee connection between public key and end entity
  - Man-In-Middle no longer works undetected
  - Guarantee authentication and non-repudiation
  - Privacy/confidentiality not an issue here
    - Only concerned with linking key to owner
- Distribute responsibility
  - Hierarchical structure

Digital Certificates

- Introduced by IEEE-X.509 standard (1988)
- Originally intended for accessing IEEE-X.500 directories
  - Concerns over misuse and privacy violation gave rise to need for access control mechanisms
  - X.509 certificates addressed this need
- From X.500 comes the Distinguished Name (DN) standard
  - Common Name (CN)
  - Organizational Unit (OU)
  - Organization (O)
  - Country (C)
- Supposedly enough to give every entity on Earth a unique name

Obtaining Certificates



Obtaining Certificates

- 1. Alice generates A_{priv}, A_{pub} and A_{ID}; Signs $\{A_{pub}, A_{ID}\}$ with A_{priv}
  - Proves Alice holds corresponding A_{priv}
  - Protects $\{A_{pub}, A_{ID}\}$ en route to CA
- 2. CA verifies signature on $\{A_{pub}, A_{ID}\}$
  - Verifies A_{ID} offline (optional)
- 3. CA signs $\{A_{pub}, A_{ID}\}$ with CA_{priv}
  - Creates certificate
  - Certifies binding between A_{pub} and A_{ID}
  - Protects $\{A_{pub}, A_{ID}\}$ en route to Alice
- 4. Alice verifies $\{A_{pub}, A_{ID}\}$ and CA signature
  - Ensures CA didn't alter $\{A_{pub}, A_{ID}\}$
- 5. Alice and/or CA publishes certificate

PKI: Risks

- Certificates only as trustworthy as their CAs
  - Root CA is a single point of failure
- PKI only as secure as private signing keys
- DNS not necessarily unique
- Server certificates authenticate DNS addresses, not site contents
- CA may not be authority on certificate contents
  - i.e., DNS name in server certificates

Real World Protocol

- Secure Sockets Layer (SSL)
  - Client/server authentication, secure data exchange
- Secure Multipurpose Internet Mail Extensions Protocol (S/MIME), PGP
- Secure Electronic Transactions (SET)
- Internet Protocol Secure Standard (IPSec)
  - Authentication for networked devices

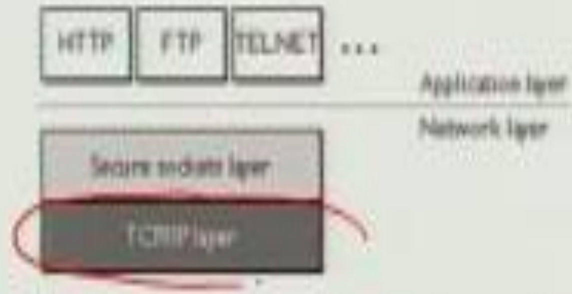
Basics Steps

- Authenticate (validate the other side)
- Key agreement/exchange (agree on or exchange a secret key)
- Confidentiality (exchange encrypted messages)
- Integrity (proof message not modified)
- Nonrepudiation (proof you got exactly what you want)

Secure Sockets Layer (SSL)

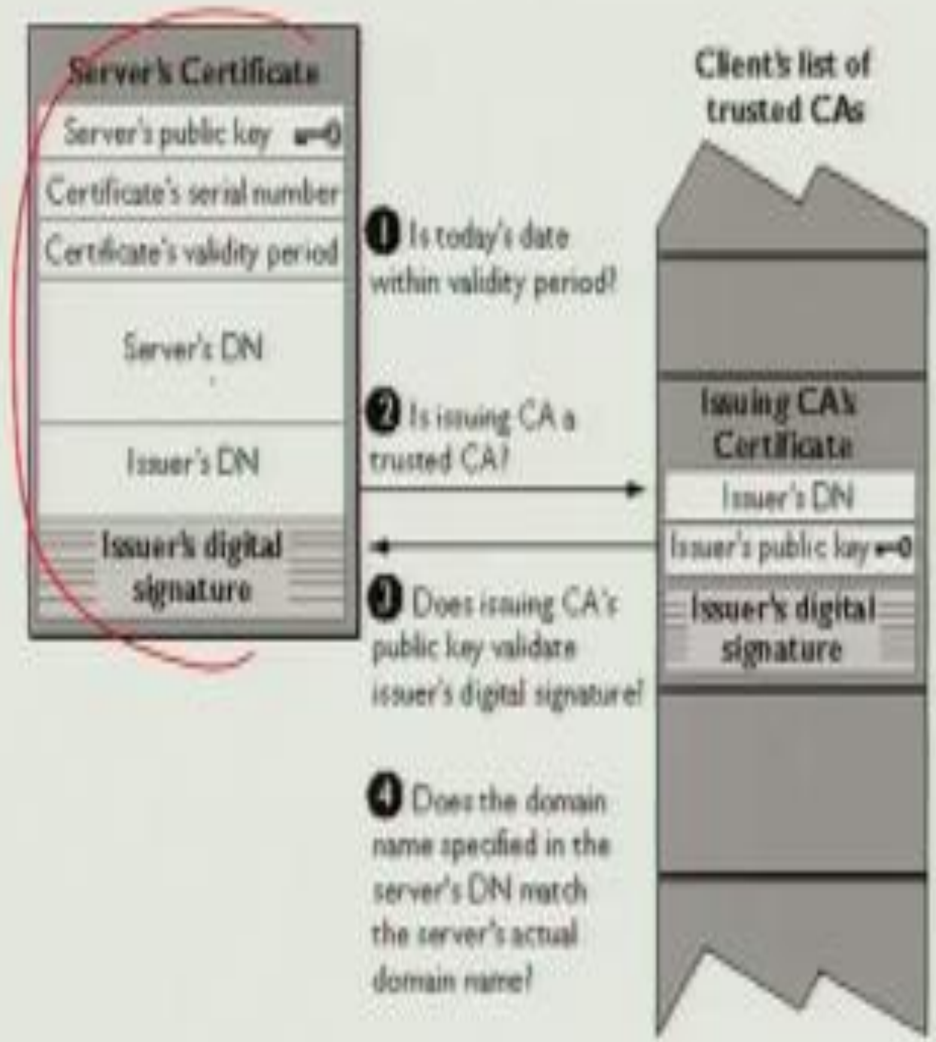
- Developed by Netscape
- Provides privacy
  - Encrypted connection
    - Confidentiality and tamper-detection
- Provides authentication
  - Authenticate server
  - Authenticate client optionally

Secure Sockets Layer (cont.)

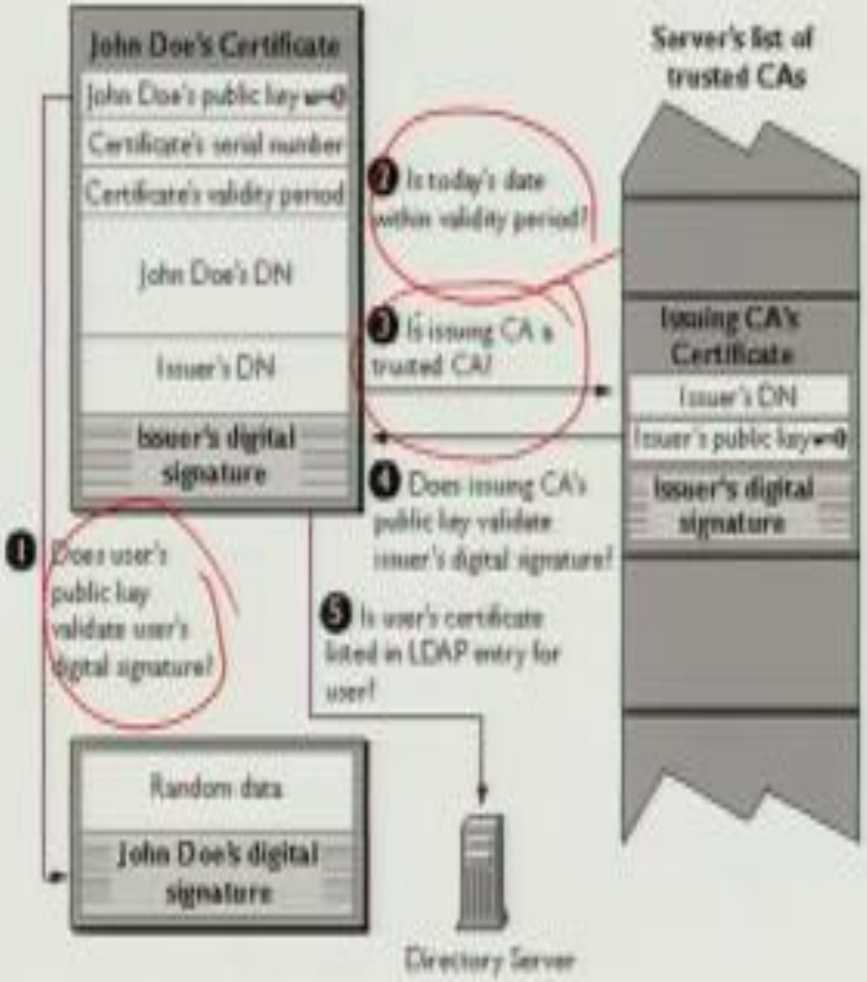


- Lies above transport layer, below application layer
  - Can lie atop any transport protocol, not just TCP/IP
  - Runs under application protocols like HTTP, FTP, and TELNET

SSL: Server Authentication



SSL: Client Authentication



Network Security -Terminology

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

<u>Confidentiality (Encryption)</u>

Sender:

- Compute $C = Enc_K(M)$
- Send C

Receiver:

- Recover $M = Dec_K(C)$

<u>Auth/Integrity (MAC / Signature)</u>

Sender:

- Compute $s = Sig_K(Hash(M))$
- Send <M, s>

Receiver:

- Compute $s' = Ver_K(Hash(M))$
- Check $s' == s$

These are simplified forms of the actual algorithms

Email Security, Certificates

Email Security:
Pretty Good Privacy (PGP)

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

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

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 Threat Model

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

Wireshark

HTTP-S: Securing HTTP

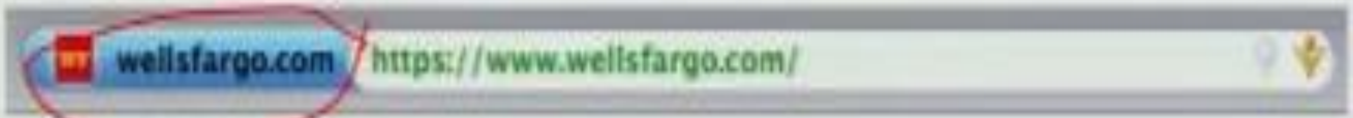
https://

- 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!

OpenSSL

2013
Heartbleed

Learning a Valid Public Key

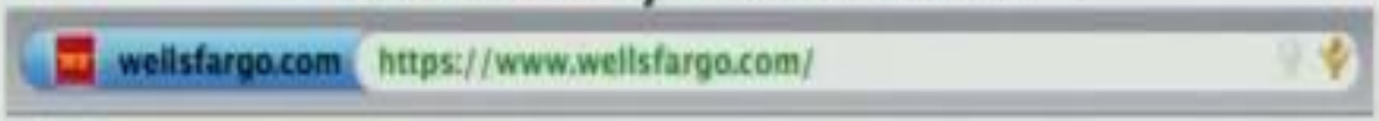


- What is that lock?
  - Securely binds domain name to public key (PK)
    - If PK is authenticated, then any message signed by that PK cannot be forged by non-authorized party
  - Believable only if you trust the attesting body
    - Bootstrapping problem: Who to trust, and how to tell if this message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



Security Protocol

TLS: Transport Layer Secuirty
SSL: Secure Security Layer
IP Seuirty
MAC Secuirty

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
- Switch to new symmetric-key cipher using shared key

- Create shared secret key from pre-secret and random
- Switch to new symmetric-key cipher using shared key

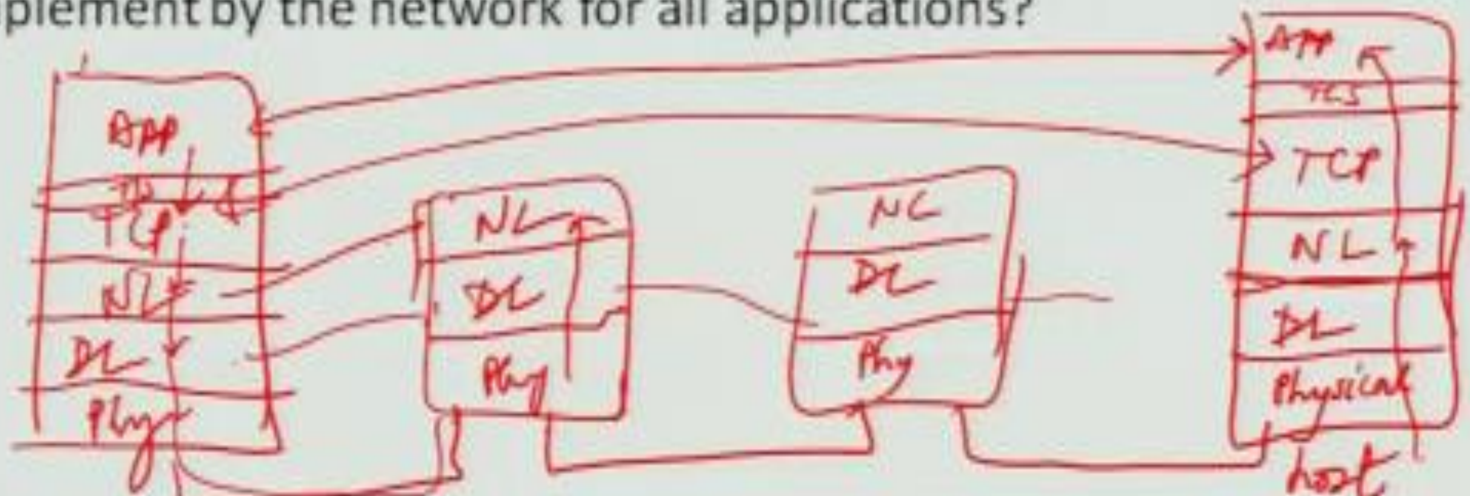
Comments on HTTPS

- 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

Module 7.10
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?

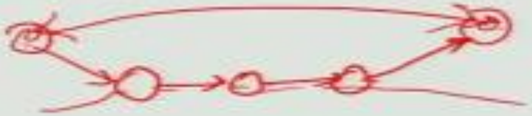


IPSec

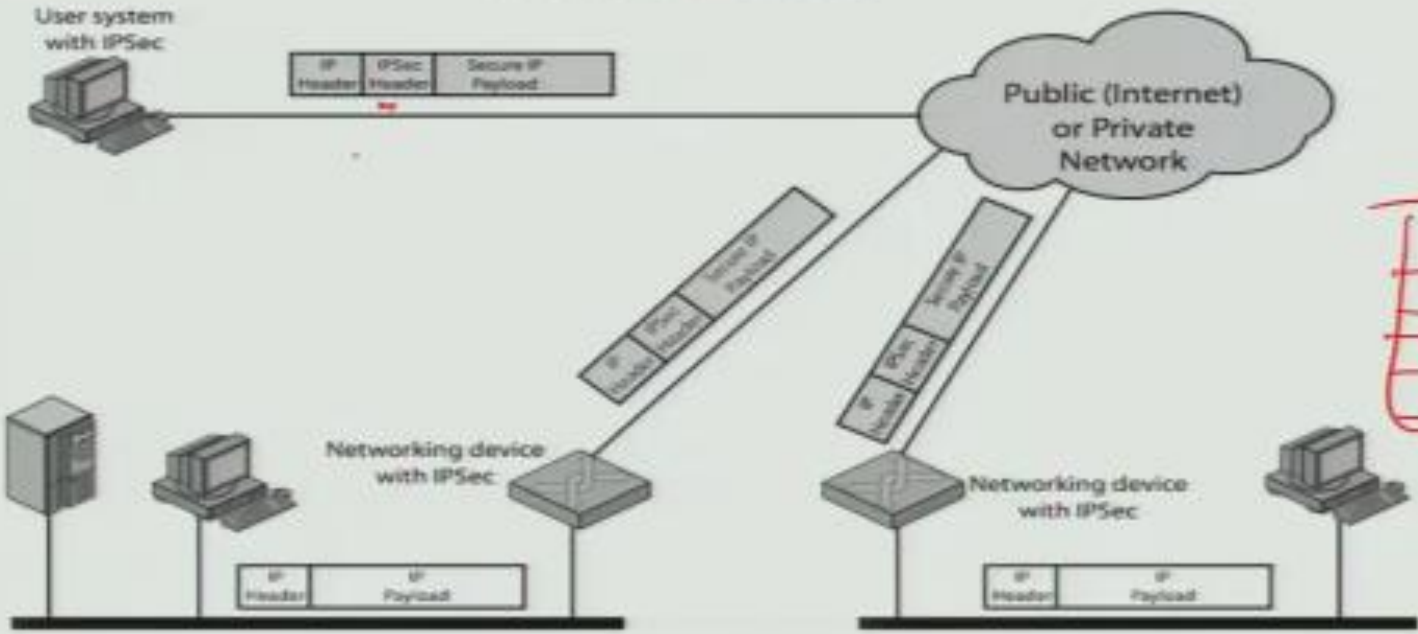
- General IP Security framework
- Allows one to provide
  - Access control, integrity, authentication, originality, and confidentiality
- Applicable to different settings
  - Narrow streams: Specific TCP connections
  - Wide streams: All packets between two gateways

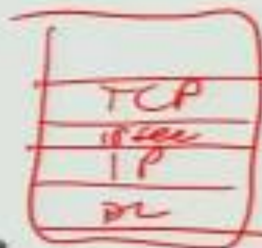
IPSec

- 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





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 Root Servers

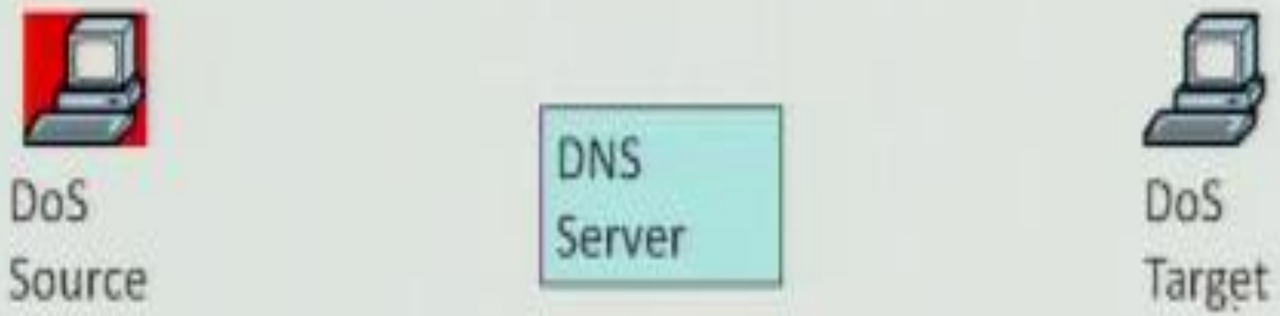
- 13 root servers (see http://www.root-servers.org/)
- Labeled A through M

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

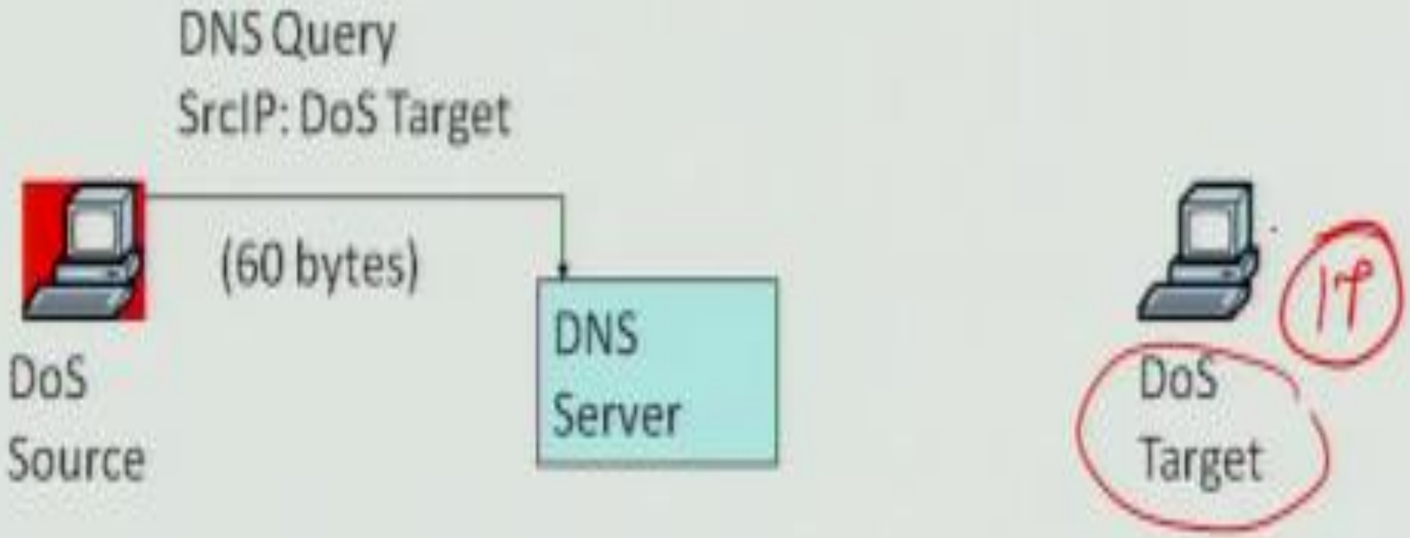
Denial-of-Service Attacks on Hosts

×40 amplification

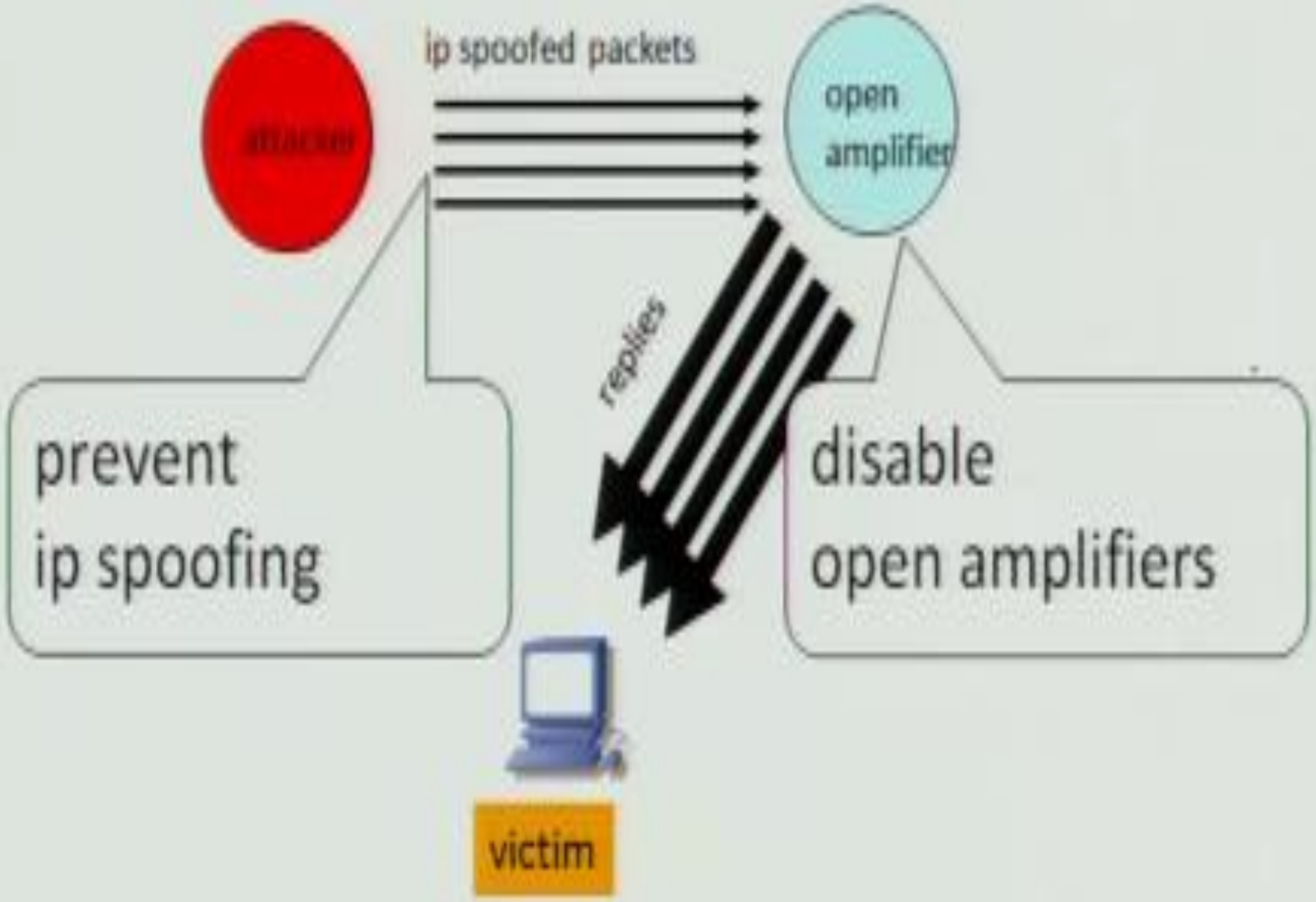


Denial-of-Service Attacks on Hosts

×40 amplification



Preventing Amplification Attacks



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)

2^{16}

- DNS hijacking
  - 16 bits: 65K possible IDs
  - What rate to enumerate all in 1 sec? 64B/packet
  - 64*65536*8 / 1024 / 1024 = 32 Mbps

Let's strongly believe the answer!
Enter DNSSEC

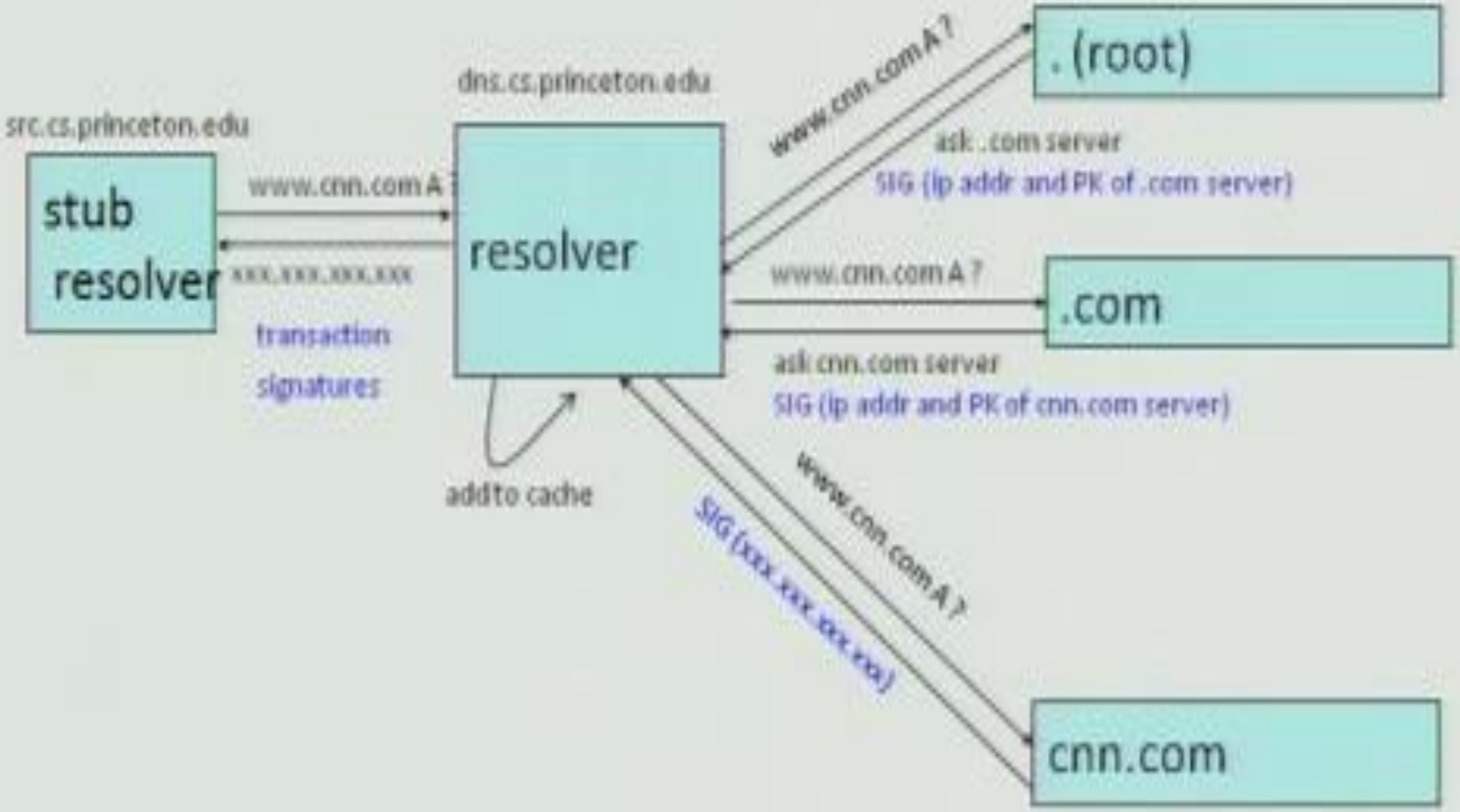
- DNSSEC protects against data spoofing and corruption
- DNSSEC also provides mechanisms to authenticate servers and requests
- DNSSEC provides mechanisms to establish authenticity and integrity

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