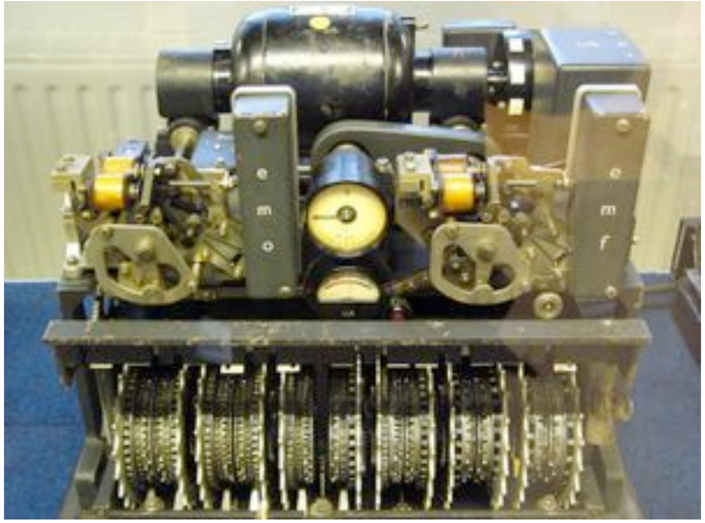


Network and Communication Security

COS 461: Computer Networks

Spring 2009 (MW 1:30-2:50 in COS 105)

Mike Freedman

Teaching Assistants: Wyatt Lloyd and Jeff Terrace

http://www.cs.princeton.edu/courses/archive/spring09/cos461/

Overview

- Network security and definitions
- Brief introduction to cryptography
  - Cryptographyic hash hunctions
  - Symmetric-key crypto
  - Public-key crypto
- IP-Sec
- DNS-Sec

(Slides partially from Nick Feamster's GATech network security course)

Internet’s Design: Insecure

- Designed for simplicity
- “On by default” design
- Readily available zombie machines
- Attacks look like normal traffic
- Internet’s federated operation obstructs cooperation for diagnosis/mitigation

Security: Definition

- *Security* is a state of well-being of information and infrastructures in which the possibility of successful yet undetected theft, tampering, and disruption of information and services is kept low or tolerable

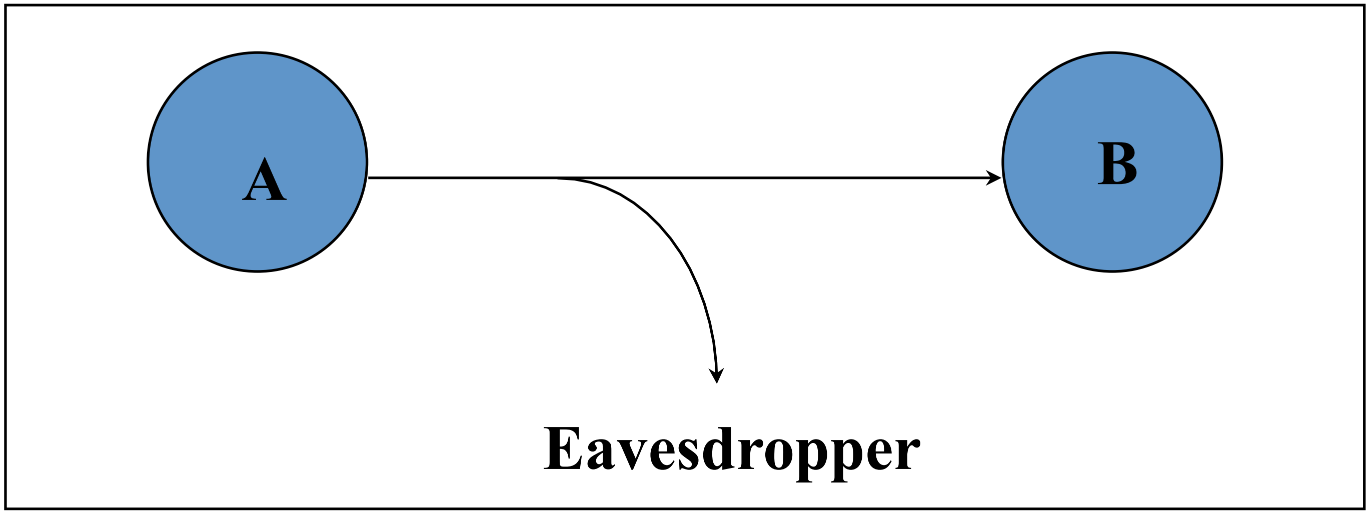
- Security rests on confidentiality, authenticity, integrity, and availability

Basic Components

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

Eavesdropping - Message Interception (Attack on Confidentiality)

- Unauthorized access to information
- Packet sniffers and wiretappers
- Illicit copying of files and programs



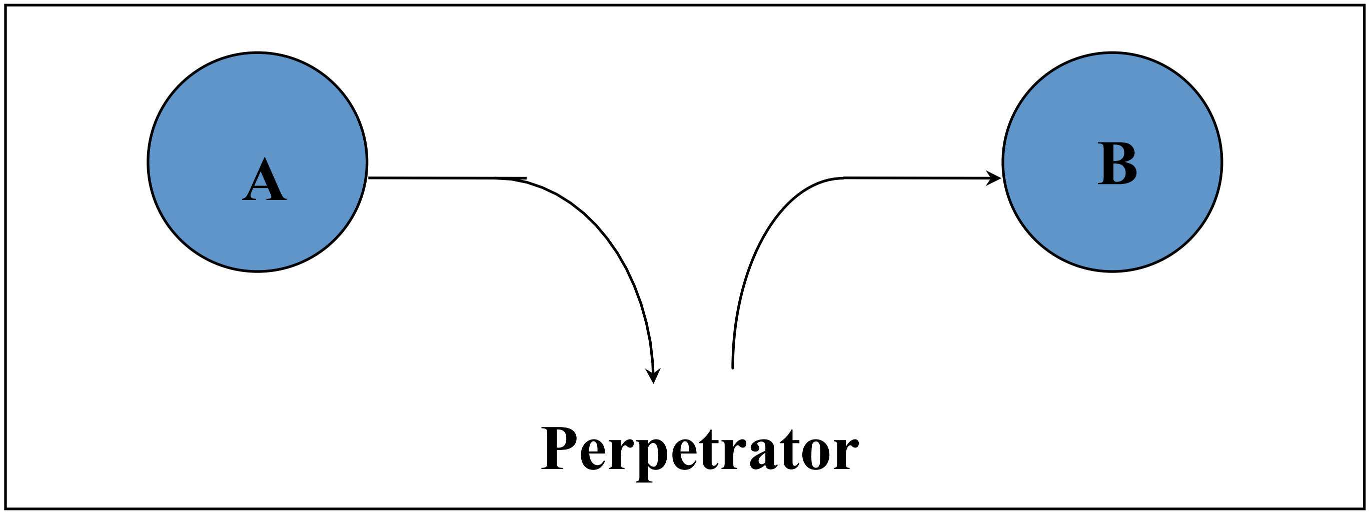
Eavesdropping Attack: Example

- tcpdump with promiscuous network interface
  - On a switched network, what can you see?

- What might the following traffic types reveal about communications?
  - DNS lookups (and replies)
  - IP packets without payloads (headers only)
  - Payloads

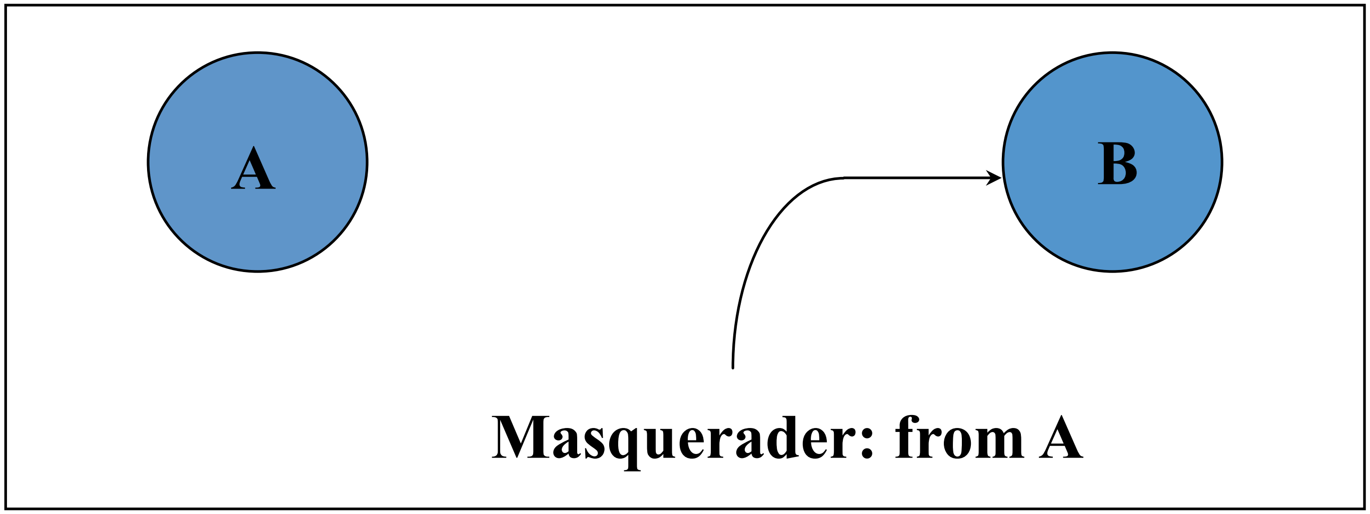
Integrity Attack - Tampering

- Stop the flow of the message
- Delay and optionally modify the message
- Release the message again



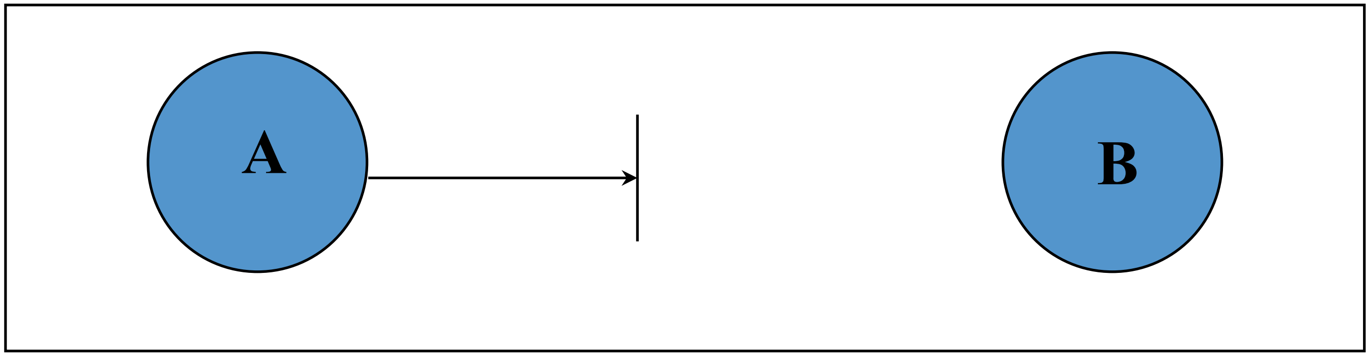
Authenticity Attack - Fabrication

- Unauthorized assumption of other's identity
- Generate and distribute objects under this identity



Attack on Availability

- Destroy hardware (cutting fiber) or software
- Modify software in a subtle way
- Corrupt packets in transit



- Blatant *denial of service* (DoS):
  - – Crashing the server
  - – Overwhelm the server (use up its resource)

Impact of Attacks

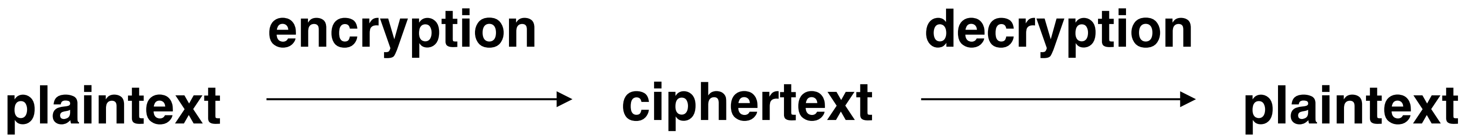
- Theft of confidential information
- Unauthorized use of
  - Network bandwidth
  - Computing resource
- Spread of false information
- Disruption of legitimate services

All attacks can be related and are dangerous!

Introduction to Cryptography

What is Cryptography?

- Comes from Greek word meaning “secret”
  - Primitives also can provide integrity, authentication
- Cryptographers invent secret codes to attempt to hide messages from unauthorized observers



plaintext

encryption

ciphertext

decryption

plaintext

- Modern encryption:
  - *Algorithm* is public, *key* is secret and provides security
  - May be symmetric (secret) or asymmetric (public)

Cryptographic Algorithms: Goal

- Given key, relatively easy to compute

- Without key, hard to compute (invert)

- “Level” of security often based on “length” of key

Three Types of Functions

- Cryptographic hash Functions
  – Zero keys

- Secret-key functions
  – One key

- Public-key functions
  – Two keys

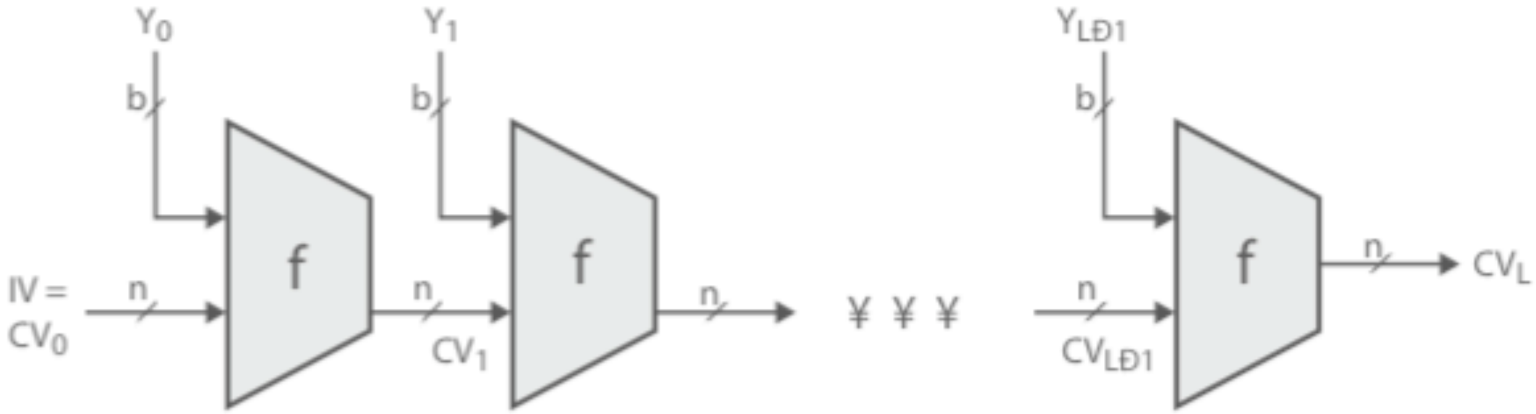
Cryptographic hash functions

Cryptography Hash Functions

- Take message, *m*, of arbitrary length and produces a smaller (short) number, *h(m)*

- Properties
  - Easy to compute *h(m)*
  - Pre-image resistance: Hard to find an *m*, given *h(m)*
    - *"One-way function"*
  - Second pre-image resistance: Hard to find two values that hash to the same *h(m)*
    - *E.g.* discover collision: *h(m) == h(m') for m != m'*
  - Often assumed: output of hash fn's "looks" random

Hash Algorithm Structure



IV = Initial value

CV_i = chaining variable

Y_i = ith input block

f = compression algorithm

L = number of input blocks

n = length of hash code

b = length of input block

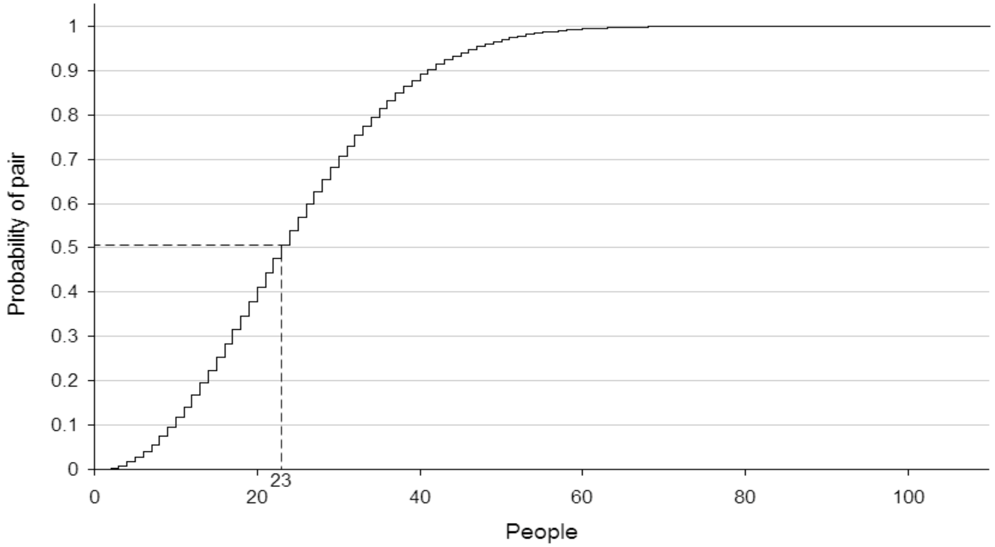
Hash and MAC Algorithms

- Hash Functions
  - Condense arbitrary size message to fixed size
  - By processing message in blocks
  - Through some compression function
  - Either custom or block cipher based

- Message Authentication Code (MAC)
  - Fixed sized authenticator for some message
  - To provide authentication for message
  - By using block cipher mode or hash function

How hard to find collisions? Birthday Paradox

- Compute probability of *different* birthdays
- Random sample of *n* people taken from *k*=365 days
- Probability of no repetition:
  - *P = 1 - (1)(1 - 1/365)(1 – 2/365)(1 – 3/365) … (1 – (n-1)/365)*
  - *P ~ 1 – e^-(n(n-1)/2k*
  - *Let k=n, P ~ 2^N/2*



How Many Bits for Hash?

- If *m* bits, takes $2^{m/2}$ to find weak collision
  - – Still takes 2^m to find strong (pre-image) collision
- 64 bits, takes 2^{32} messages to search (easy!)
- Now, MD5 (128 bits) considered too little
- SHA-1 (160 bits) getting old

Example use

- Password hashing
  - Can't store passwords in a file that could be read
    - Concerned with insider attacks!
  - Must compare typed passwords to stored passwords
    - Does hash (typed) == hash (password) ?
  - Actually, a "salt" is often used: hash (input || salt)

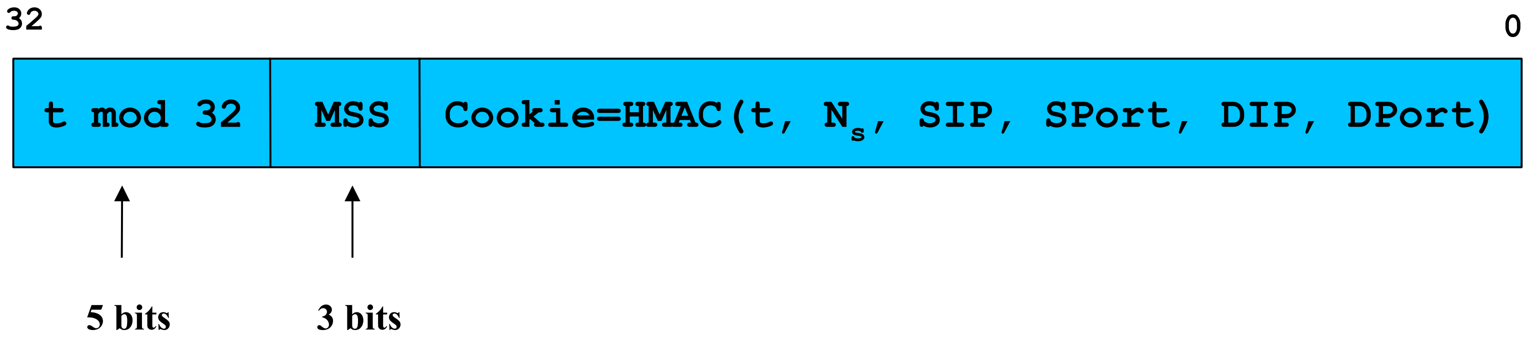
- File-sharing software (Freenet, BitTorrent)
  - File named by F_{name} = hash (data)
  - Participants verify that hash (downloaded) == F_{name}

Example use #2: TCP SYN cookies

- General idea
  - Client sends SYN w/ ACK number
  - Server responds to Client with SYN-ACK cookie
    - sqn = f (time, rand nonce, src ip, src port, dest ip, dest port)
    - Server does not save state
  - Honest client responds with ACK (sqn)
  - Server checks response
  - If matches SYN-ACK, establishes connection
- Prevents resource-exhausting attack by clients

Example use #2: TCP SYN cookies

- TCP SYN/ACK seqno encodes a cookie
  - 32-bit sequence number
    - **t mod 32:** counter to ensure sequence numbers increase every 64 seconds
    - **MSS:** encoding of server MSS (can only have 8 settings)
    - **Cookie:** easy to create and validate, hard to forge
      - Includes timestamp, nonce, 4-tuple



32

0

t mod 32 | MSS | Cookie=HMAC(t, N_s, SIP, SPort, DIP, DPort)

5 bits

3 bits

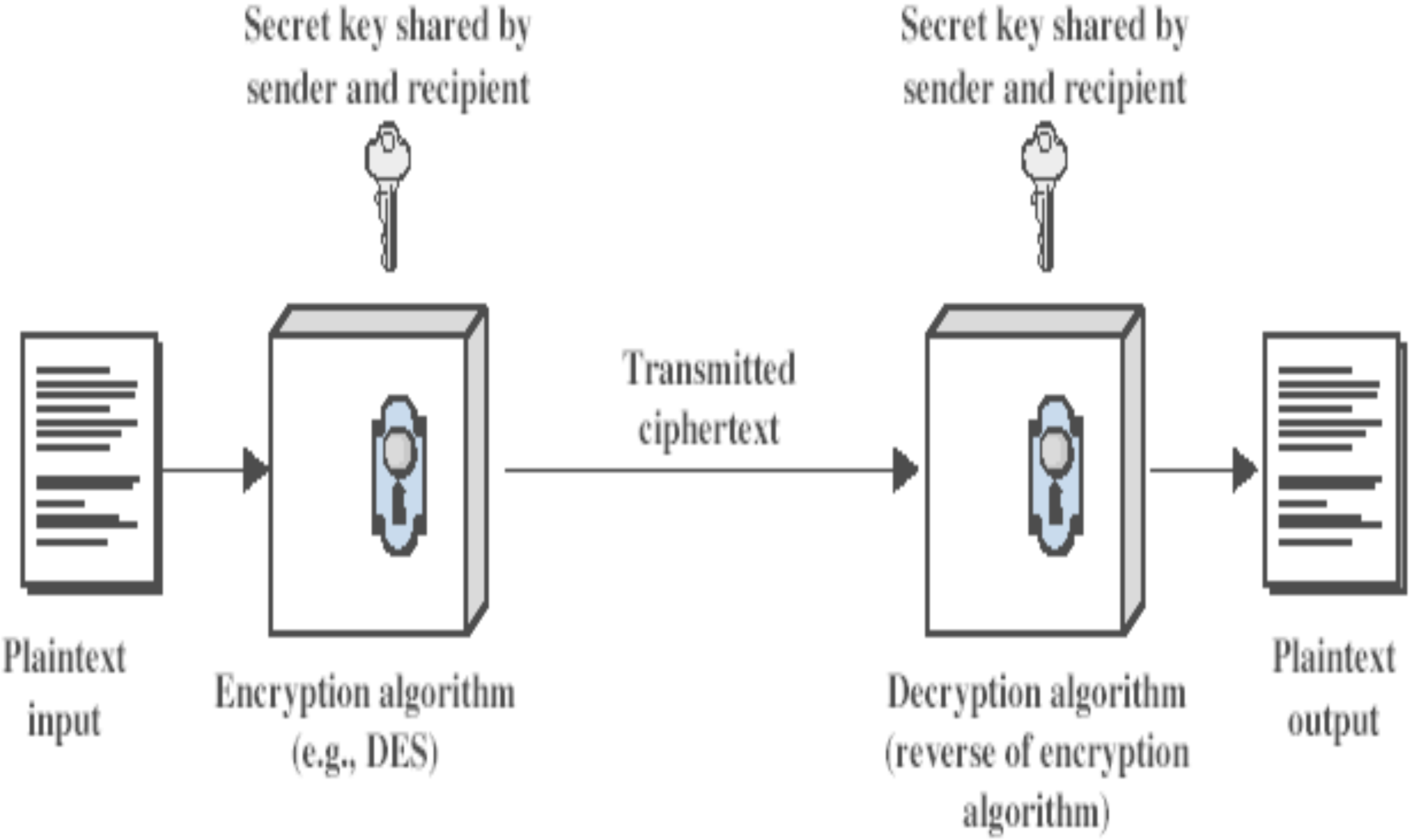
Symmetric (Secret) Key Cryptography

Symmetric Encryption

- Also: “conventional / private-key / single-key”
  - Sender and recipient share a common key
  - All classical encryption algorithms are private-key

- Was only type of encryption prior to invention of public-key in 1970’s
  - And by far most widely used
  - Typically more computationally efficient

Symmetric Cipher Model


Secret key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Requirements

- Two requirements
  - a strong encryption algorithm
  - a secret key known only to sender / receiver
- Mathematically:

  $Y = \mathrm{E}_K(X)$; $X = \mathrm{D}_K(Y)$

- **Goal:** Given key, generate a 1-to-1 mapping to ciphertext that looks random if key unknown

- Assume encryption algorithm is known
- Implies a secure channel to distribute key

Block vs. Stream Ciphers

- Block ciphers process messages in blocks, each of which is then en/decrypted
  - Each block 64-bits or more
  - DES, AES, etc…
- Stream ciphers process messages a bit or byte at a time when en/decrypting (e.g., MD4)

Public-Key Cryptography

Why Public-Key Cryptography?

- Developed to address two key issues:
  - **Key distribution** – how to secure communication without having to trust a KDC with your key
  - **Digital signatures** – how to verify msg comes intact from claimed sender (w/o prior establishment)

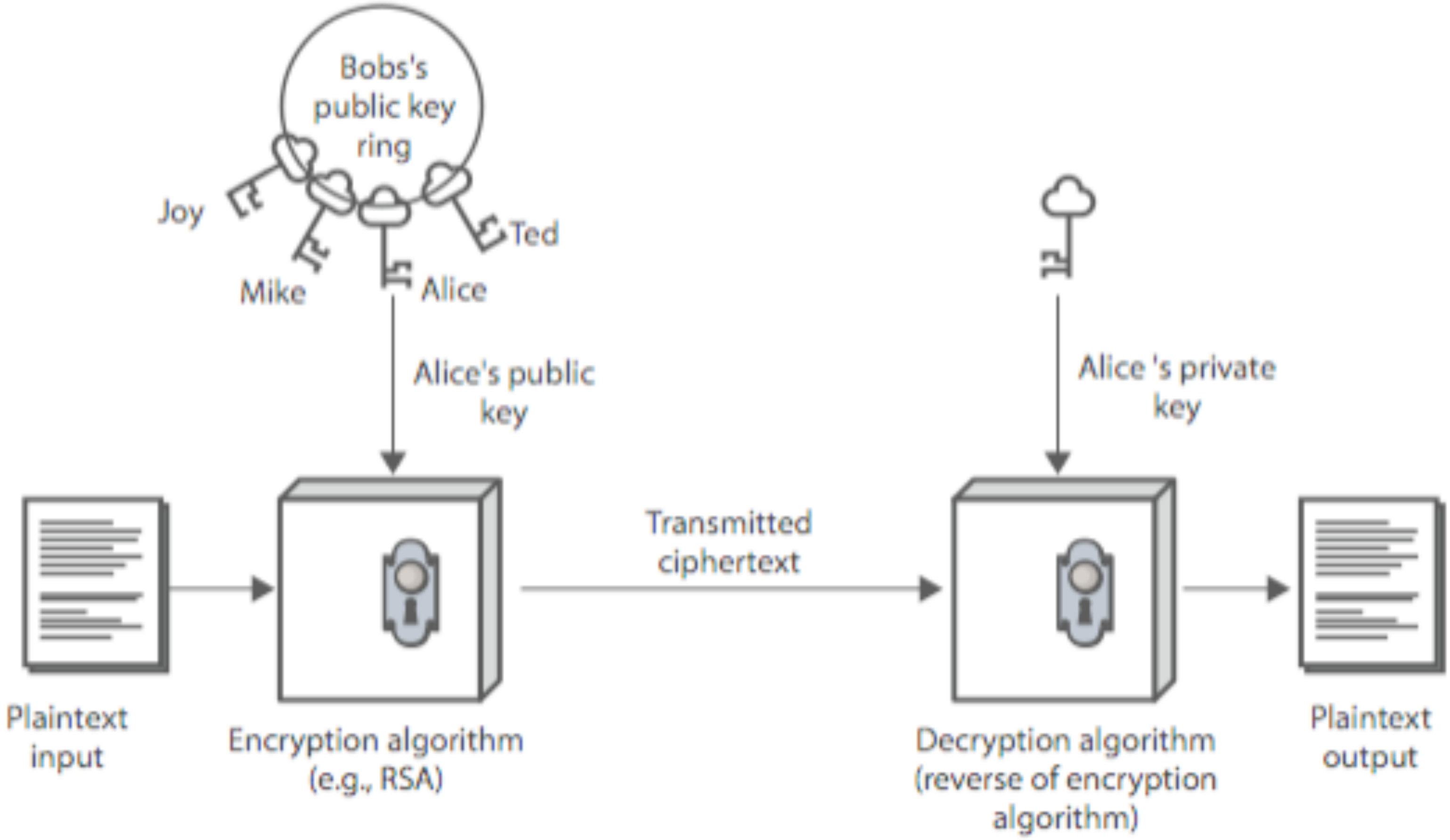
- Public invention due to Whitfield Diffie & Martin Hellman in 1976
  - known earlier in classified community

Public-Key Cryptography

- **Public-key/two-key/asymmetric** cryptography involves the use of **two** keys:
  - A **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
  - A **private-key**, known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures**

- Is **asymmetric** because
  - Those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures
  - If “one-way function” goes c ← F(m), then public-key encryption is a “trap-door” function:
    - Easy to compute c ← F(m)
    - Hard to compute m ← F^{-1}(m) without knowing k
    - Easy to compute m ← F^{-1}(m,k) by knowing k

Public-Key Cryptography



(a) Encryption

Security of Public Key Schemes

- Like private key schemes brute force **exhaustive search** attack is always theoretically possible
  - But keys used are too large (> 1024bits)
- Security relies on a **large enough** difference in difficulty between **easy** (compute) and **hard** (invert without trapdoor) problems
  - More generally the **hard** problem is known, but is made hard enough to be impractical to break
- Requires the use of **very large numbers**
  - Hence is **slow** compared to private key schemes

RSA

- Rivest, Shamir, & Adleman in 1977
  - best known & widely used public-key scheme
- Based on exponentiation in a finite field over integers modulo a prime
  - Exponentiation takes $O((\log n)^3)$ operations (easy)
  - Uses large integers (e.g., 1024 bits)
- **Security** due to cost of factoring large numbers
  - factorization takes $O(e^{\log n \log \log n})$ operations (hard)

RSA Algorithm

- Key generation
  - Generate two large primes `p` and `q`; compute `n=p*q`
  - Find `e,d` such that `e*d = 1 mod (p-1)(q-1)`
- To encrypt a message M the sender:
  - Obtain **public key** of recipient `PU={e,n}`
  - Compute $C = M^e \bmod n$, where $0 \leq M < n$
- To decrypt the ciphertext C the owner:
  - Uses private key `PR={d,n}`
  - Computes $M = C^d \bmod n$
- Note that msg M must be smaller than the modulus n
  - Otherwise, hybrid encryption:
    - Generate random symmetric key *r*
    - Use public key encryption to encrypt *r*
    - Use symmetric key encryption under *r* to encrypt *M*

IP Security

IP Security

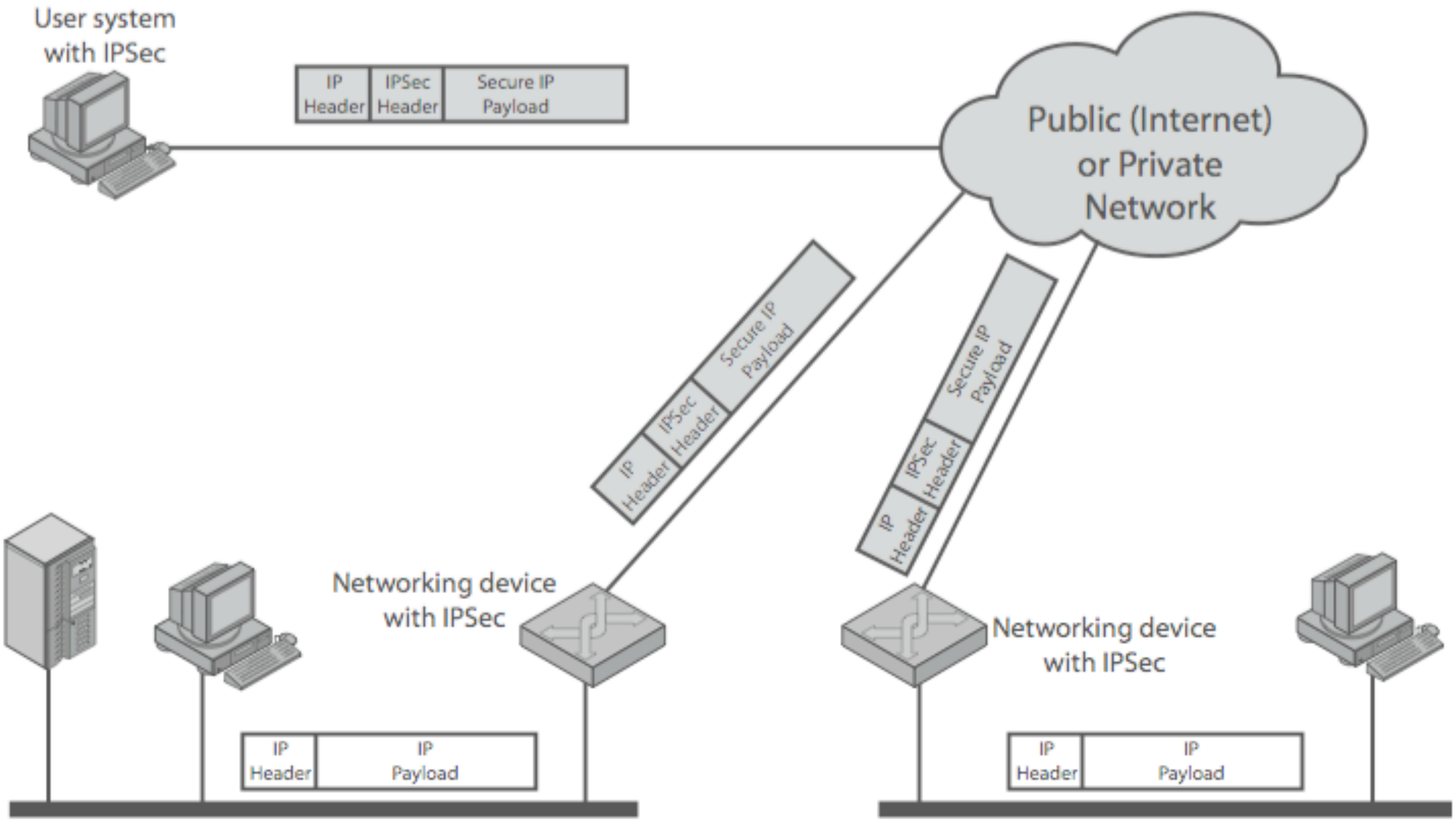
- There is range of app-specific security mechanisms
  - eg. S/MIME, PGP, Kerberos, SSL/HTTPS
- However there are security concerns that cut across protocol layers
- Implement by the network for all applications?

Enter IPSec!

IPSec

- General IP Security mechanisms
- Provides
  - authentication
  - confidentiality
  - key management
- Ppplicable to use over LANs, across public & private WANs, and for the Internet

IPSec Uses


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

Benefits of IPSec

- If in a firewall/router:
  - Provides strong security to all traffic crossing the perimeter
  - Resistant to bypass
- Is below transport layer, hence transparent to applications
- Can be transparent to end users
- Can provide security for individual users
- Secures routing architecture

IP Security Architecture

- Specification is quite complex
- Defined in numerous RFC's
  - Incl. RFC 2401 / 2402 / 2406 / 2408
- Mandatory in IPv6, optional in IPv4
- Have two security header extensions:
  - Authentication Header (AH)
  - Encapsulating Security Payload (ESP)

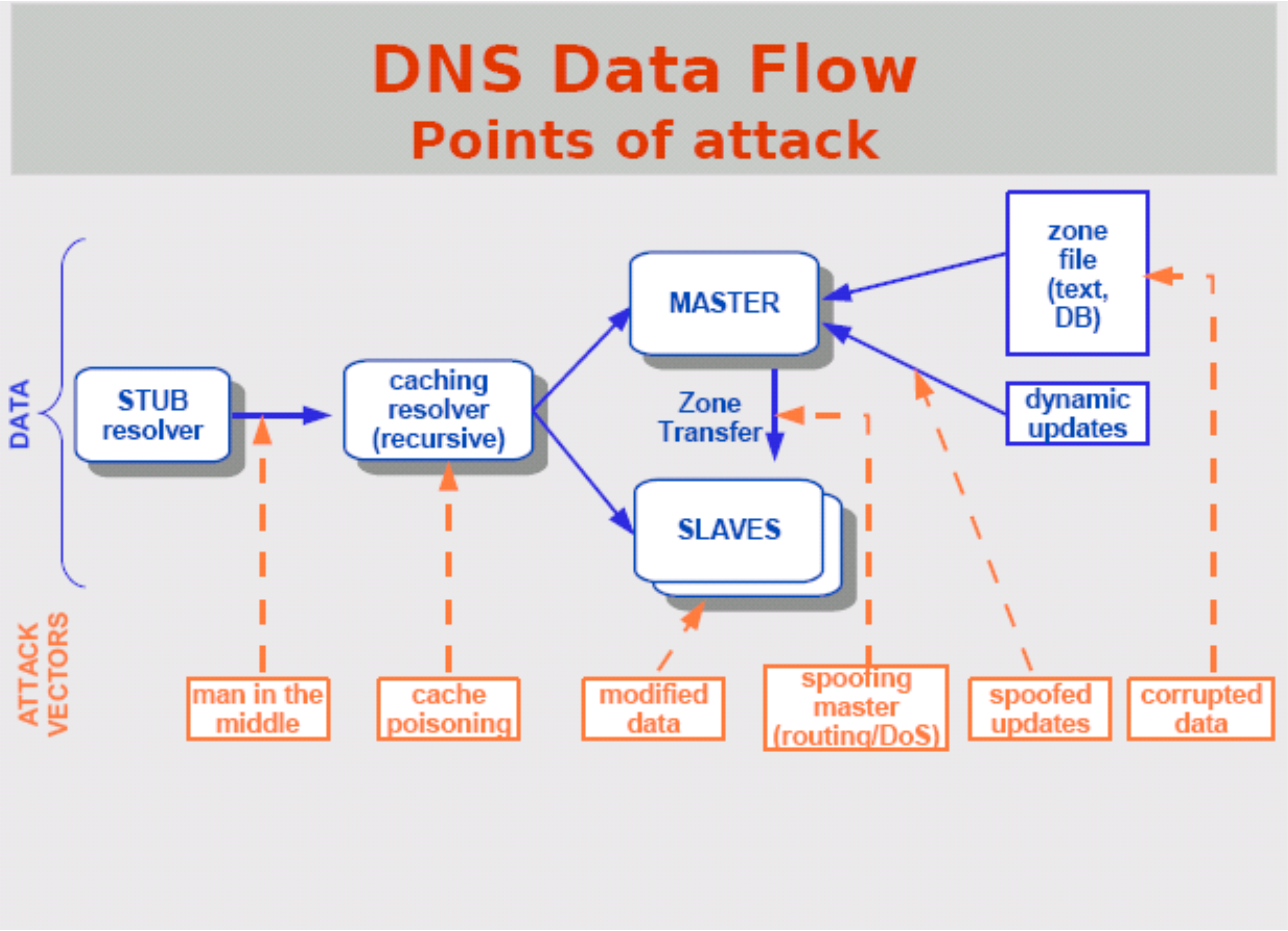
IPSec Services

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets
  - A form of partial sequence integrity via seq #'s
  - But not as robust as if on top of TCP (why not?)
- Confidentiality (encryption)
- Limited traffic flow confidentiality

Transport vs. Tunnel Mode ESP

- Transport mode is used to encrypt & optionally authenticate IP data
  - Data protected but header left in clear
  - Can 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

DNS Security



DNS Data Flow

Points of attack

Source: http://nsrc.org/tutorials/2009/apricot/dnssec/dnssec-tutorial.pdf

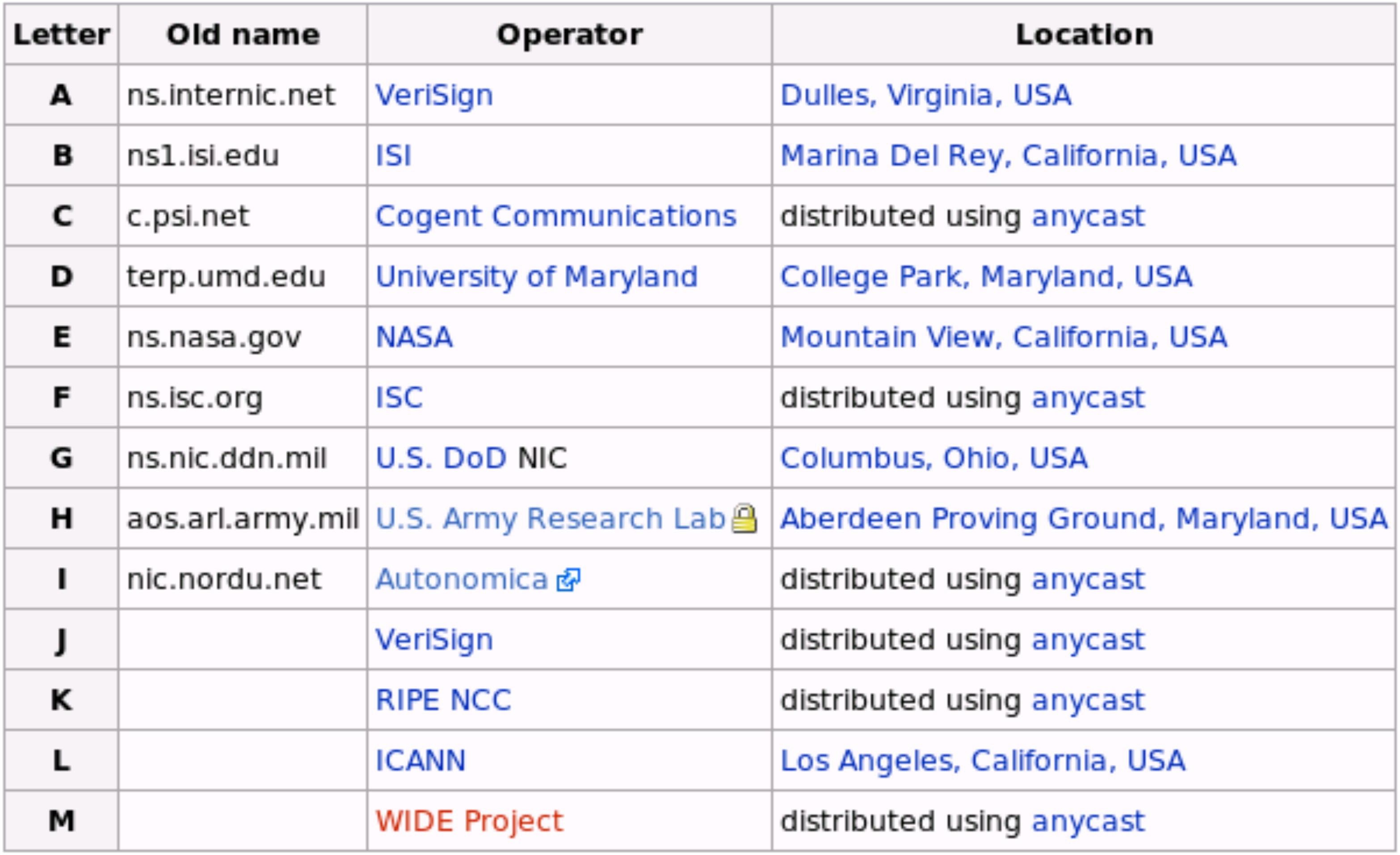
Root level DNS attacks

- 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

Do you trust the TLD operators?

- Wildcard DNS record for all .com and .net domain names not yet registered by others
  - September 15 – October 4, 2003
  - February 2004: Verisign sues ICANN
- Redirection for these domain names to Verisign web portal: "to help you search"
  - and serve you ads…and get "sponsored" search

Defense: Replication and Caching

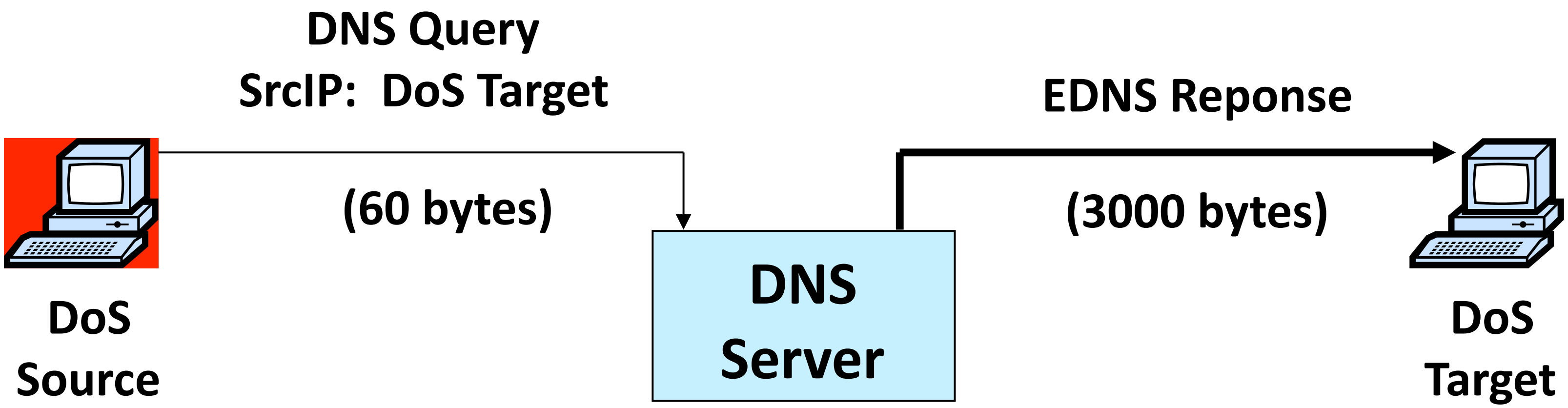


Letter	Old name	Operator	Location
A	ns.internic.net	VeriSign	Dulles, Virginia, USA
B	ns1.isi.edu	ISI	Marina Del Rey, California, USA
C	c.psi.net	Cogent Communications	distributed using anycast
D	terp.umd.edu	University of Maryland	College Park, Maryland, USA
E	ns.nasa.gov	NASA	Mountain View, California, 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

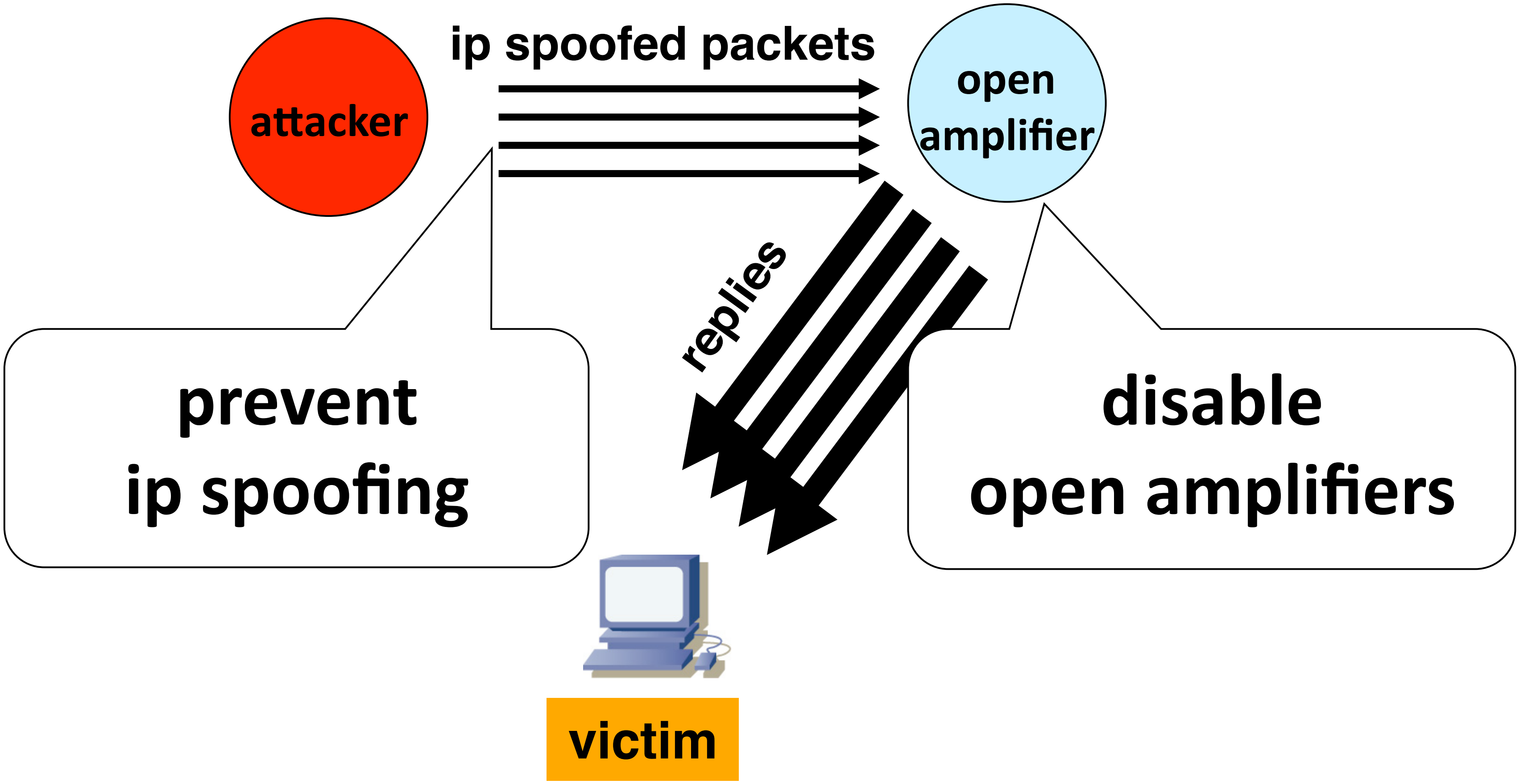
DNS Amplification Attack

DNS Amplification attack: (×40 amplification)



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

Solutions



attacker

ip spoofed packets

open amplifier

replies

victim

prevent ip spoofing

disable open amplifiers

But should we believe it?
Enter DNSSEC

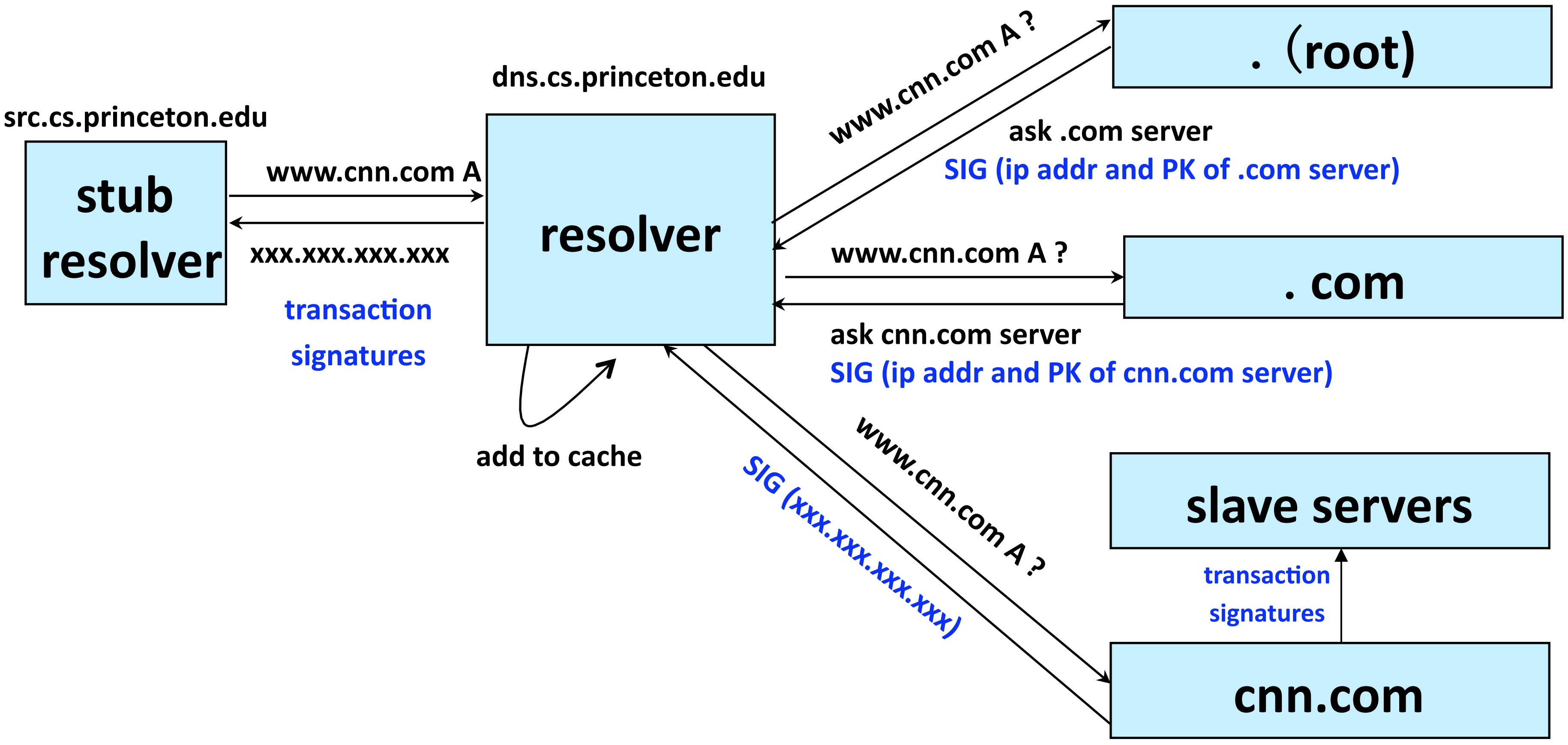
- DNSSEC protects against data spoofing and corruption
- DNSSEC also provides mechanisms to authenticate servers and requests
- DNSSEC provides mechanisms to establish authenticity and integrity

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 nameserver'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 ?



Summary

- Network security and definitions
- Introduction to cryptography
  - Cryptographic hash functions:
    - Zero keys, hard to invert, hard to find collisions
  - Symmetric-key crypto
    - One key, hard to invert, requires key distribution
  - Public-key crypto
    - Two keys, hard to invert, more expensive
- Application to crypto to help secure IP communication and DNS lookup