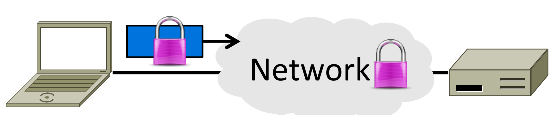
Introduction to Computer Networks

Network Security Introduction

Computer Science & Engineering
UNIVERSITY of WASHINGTON

Topic

- Network security designs to protect against a variety of threats
  - Often build on cryptography
  - Just a brief overview. Take a course!



Security Threats

- "Security" is like "performance"
  - Means many things to many people
  - Must define the properties we want
- Key part of network security is clearly stating the threat model
  - The dangers and attacker's abilities
  - Can't assess risk otherwise

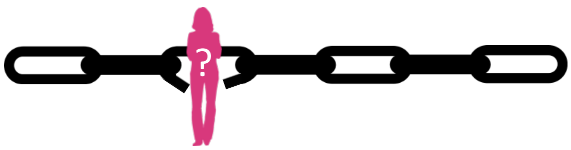
Security Threats (2)

- Some example threats
  - It's not all about encrypting messages

Attacker	Ability	Threat
Eavesdropper	Intercept messages	Read contents of message
Intruder	Compromised host	Tamper with contents of message
Impersonator	Social engineering	Trick party into giving information
Extortionist	Remote / botnet	Disrupt network services

Risk Management

- Security is hard as a negative goal
  - Try to ensure security properties and don't let anything bad happen!
- Only as secure as the weakest link
  - Could be design flaw or bug in code
  - But often the weak link is elsewhere…



Risk Management (2)

- 802.11 security … early on, WEP:
  - Cryptography was flawed; can run cracking software to read WiFi traffic
- Today, WPA2/802.11i security:
  - Computationally infeasible to break!
- So that means 802.11 is secure against eavesdropping?

Risk Management (3)

- Many possible threats
  - We just made the first one harder!
  - 802.11 is more secure against eavesdropping in that the risk of successful attack is lower. But it is not "secure".

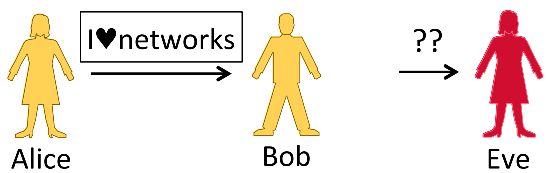
Threat Model	Old WiFi (WEP)	New WiFi (WPA2)
Break encryption from outside	Very easy	Very difficult
Guess WiFi password	Often possible	Often possible
Get password from computer	May be possible	May be possible
Physically break into home	Difficult	Difficult

Topics

- Threat models } This time
- Confidentiality
- Authentication } Crypto
- Wireless security (802.11)
- Web security (HTTPS/SSL)
- DNS security
- Virtual Private Networks (VPNs) } Applied crypto
- Firewalls
- Distributed denial-of-service } Connectivity

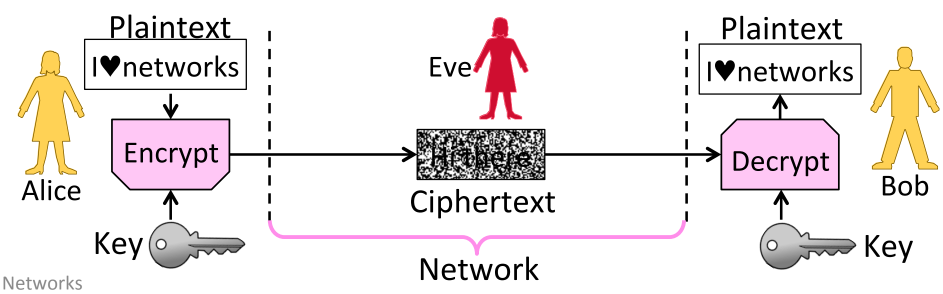
Goal and Threat Model

- Goal is to send a private message from Alice to Bob
  - This is called confidentiality
- Threat is Eve will read the message
  - Eve is a passive adversary (observes)



Encryption/Decryption Model

- Alice encrypts private message (plaintext) using key
- Eve sees ciphertext but can't relate it to private message
- Bob decrypts using key to obtain the private message



Encryption/Decryption (2)

- Encryption is a reversible mapping
  – Ciphertext is confused plaintext
- Assume attacker knows algorithm
  – Security does not rely on its secrecy
- Algorithm is parameterized by keys
  – Security does rely on key secrecy
  – Must be distributed (Achilles' heel)

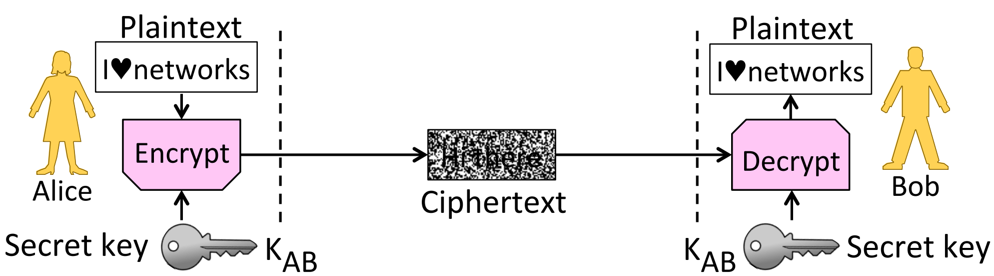
Encryption/Decryption (3)

Two main kinds of encryption:

1. Symmetric key encryption », e.g., AES
   – Alice and Bob share secret key
   – Encryption is a bit mangling box
2. Public key encryption », e.g., RSA
   – Alice and Bob each have a key in two parts: a public part (widely known), and a private part (only owner knows)
   – Encryption is based on mathematics (e.g., RSA is based on difficulty of factoring)

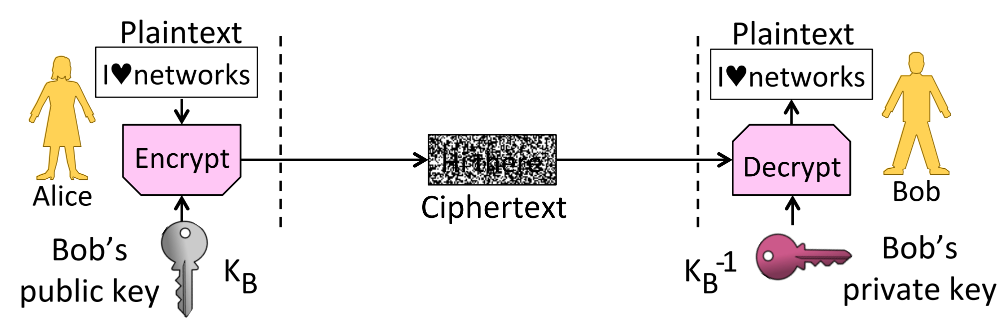
Symmetric (Secret Key) Encryption

- Alice and Bob have the same secret key, K_{AB}
  - Anyone with the secret key can encrypt/decrypt



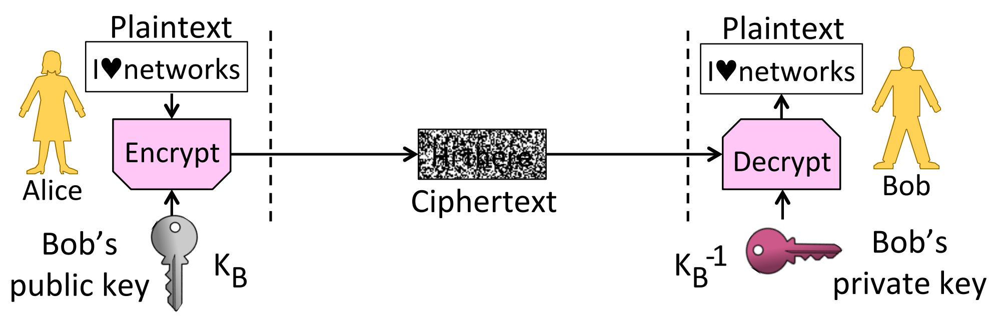
Public Key (Asymmetric) Encryption

- Alice and Bob each have public/private key pair (K_B / K_B^{-1})
  - Public keys are well-known, private keys are secret to owner

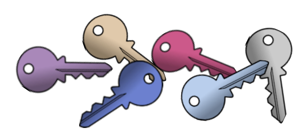


Public Key Encryption (2)

- Alice encrypts with Bob's public key K_B; anyone can send
- Bob decrypts with his private key K_B^{-1}; only he can do so



Key Distribution



- This is a big problem on a network!
  - Often want to talk to new parties
- Symmetric encryption problematic
  - Have to first set up shared secret
- Public key idea has own difficulties
  - Need trusted directory service
  - We'll look at certificates later

Symmetric vs. Public Key

- Have complementary properties
  - Want the best of both!

Property	Symmetric	Public Key
Key Distribution	Hard– share secret per pair of users	Easier– publish public key per user
Runtime Performance	Fast– good for high data rate	Slow– few, small, messages

Winning Combination

- Alice uses public key encryption to send Bob a small private message
  - It's a key! (Say 256 bits.)
- Alice and Bob send large messages with symmetric encryption
  - Using the key they now share
- The key is called a session key
  - Generated for short-term use

Introduction to Computer Networks

Message Authentication

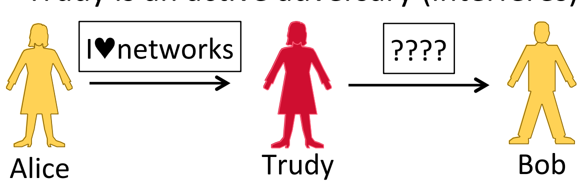
(§8.2-8.3, §8.4.2-8.4.3)

Computer Science & Engineering

UNIVERSITY *of* WASHINGTON

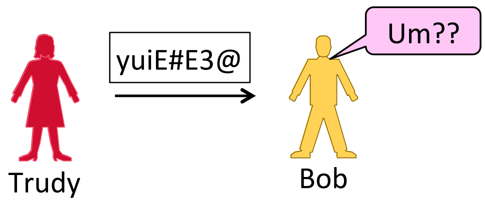
Goal and Threat Model

- Goal is to let Bob verify the message came from Alice and is unchanged
  - This is called integrity/authenticity
- Threat is Trudy will tamper with messages
  - Trudy is an active adversary (interferes)



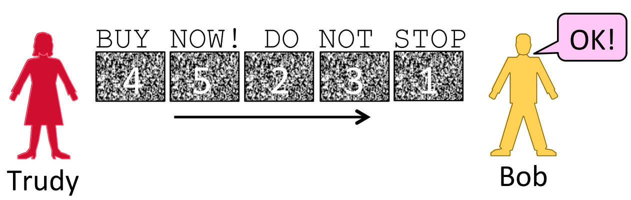
Encryption Issues

- What will happen if Trudy flips some of Alice's message bits?
  - Bob will receive an altered message



Encryption Issues

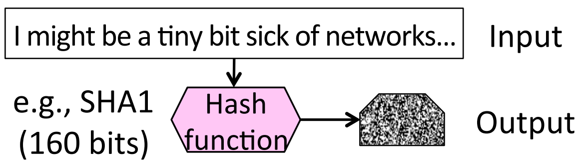
- What if Trudy reorders message?
  - Bob will receive altered message



  - Should have been (Woops)
    - "STOP DO NOT BUY NOW"

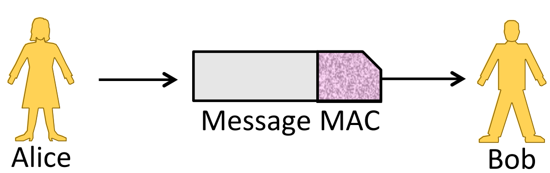
Message Digest or Cryptographic Hash

- Digest/Hash is a secure checksum
  - Deterministically mangles bits to pseudo-random output (like CRC)
  - Can't find messages with same hash
  - Acts as a fixed-length descriptor of message – very useful!



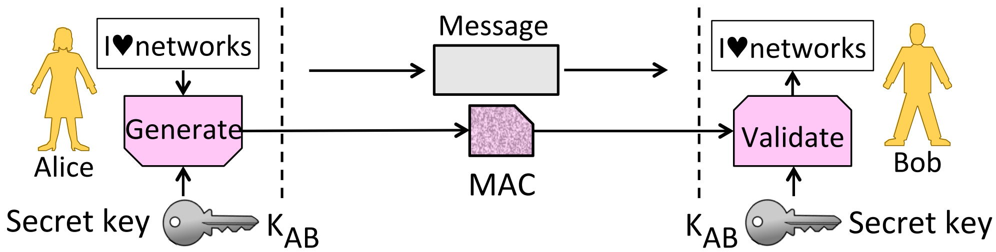
MAC (Message Authentication Code)

- MAC is a small token to validate the integrity/authenticity of a message
  - Send the MAC along with message
  - Validate MAC, process the message
  - Example: HMAC scheme



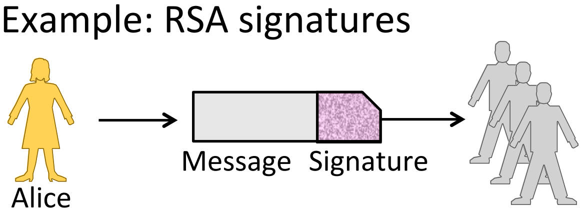
MAC (2)

- Kind of symmetric encryption operation – key is shared
  - Lets Bob validate unaltered message came from Alice
  - Doesn't let Bob convince Charlie that Alice sent the message



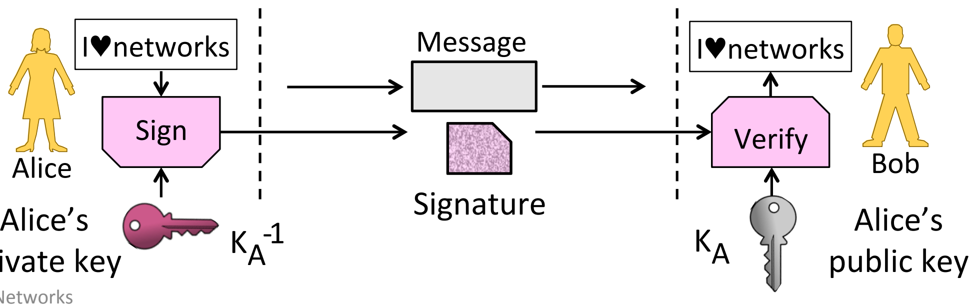
Digital Signature

- Signature validates the integrity/authenticity of a message
  - Send it along with the message
  - Lets all parties validate
  - Example: RSA signatures



Digital Signature (2)

- Kind of public key operation – public/private key parts
  - Alice signs with private key, K_A^{-1}, Bob verifies with public key, K_A
  - Does let Bob convince Charlie that Alice sent the message

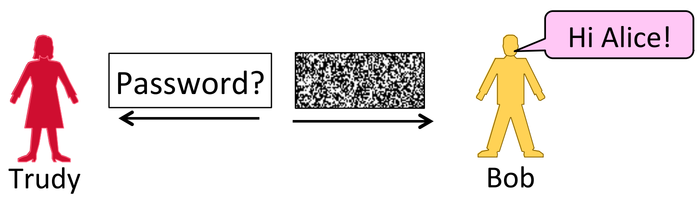


Preventing Replays

- We normally want more than confidentiality, integrity, and authenticity for secure messages!
  - Want to be sure message is fresh
- Don't want to mistake old message for a new one – a <u>replay</u>
  - Acting on it again may cause trouble

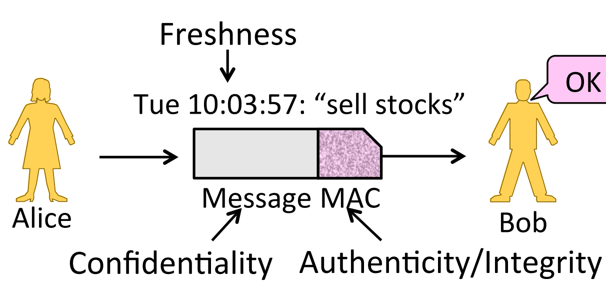
Preventing Replays (2)

- Replay attack:
  - Trudy records Alice's messages to Bob
  - Trudy later replays them (unread) to Bob; she pretends to be Alice



Preventing Replays (3)

- To prevent replays, include proof of freshness in messages
  - Use a timestamp, or <u>nonce</u>



Takeaway

- Cryptographic designs can give us integrity, authenticity and freshness as well as confidentiality.
- Real protocol designs combine the properties in different ways
  - We'll see some examples
  - Note many pitfalls in how to combine, as well as in the primitives themselves

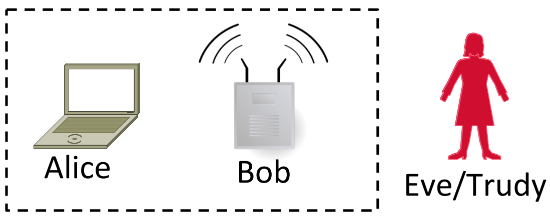
Introduction to Computer Networks

Wireless Security (§8.6.4)

Computer Science & Engineering
UNIVERSITY of WASHINGTON

Goal and Threat Model

- Unlike wired, wireless messages are broadcast to all nearby receivers
  - Don't need physical network access
  - Heightens security problems



Goal and Threat Model (2)

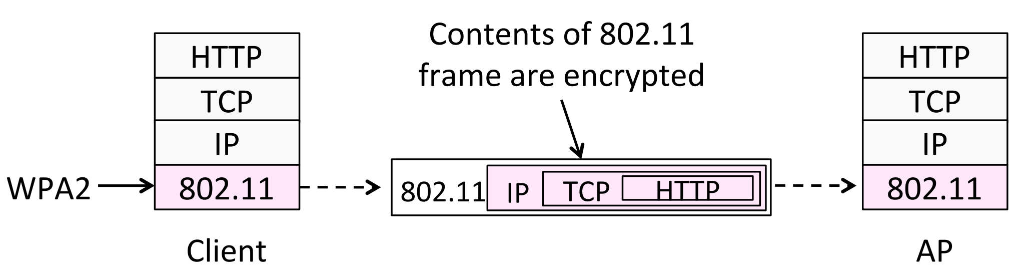
- Two main threats:
  1. Eavesdropping on conversations
  2. Unauthorized access to network
- We'll consider 802.11 setting
  - Assume external attacker can send/ receive wireless messages

802.11 Security

- Provides access control, and message confidentiality, integrity/authenticity
  - Keying based on passwords
- 802.11 standard (1999) used WEP
  - For "Wired Equivalent Privacy"
  - Badly flawed, easily broken
- 802.11i standard in 2004
  - WiFi Protected Access or WPA2
  - This is what you should use

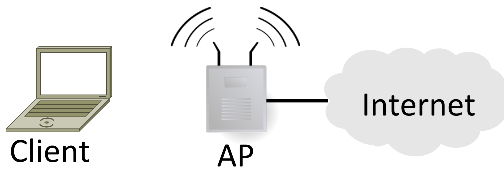
802.11 Security (2)

- Security is part of 802.11 protocol
  - Encrypted message between client and AP; removed after AP



Home Network

- AP is set up with network password
- Each client also knows password
- Client proves it knows password »
  - AP grants network access if successful

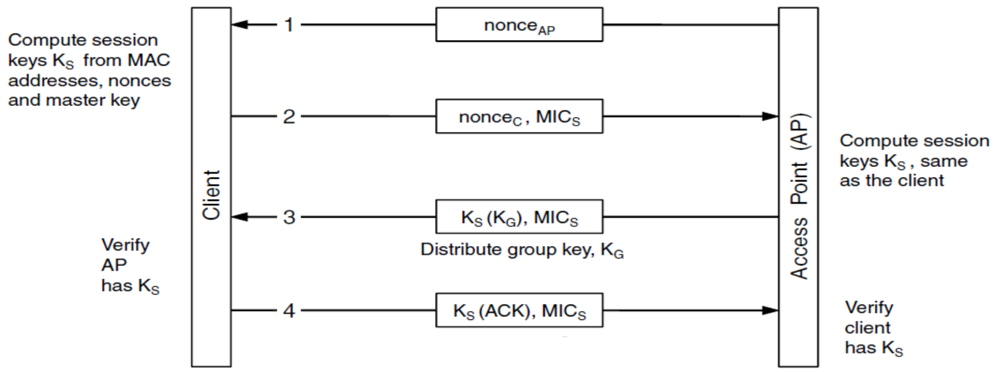


Home Network (2)

- For access, client authenticates to AP »
  - Both compute a shared session key based on the password
  - If client knows the session key it has proved that is has the password
- For usage, client/AP encrypt messages
  - For confidentiality, integrity/authenticity
  - No access without the session key
  - Also group key for AP to reach all clients

Home Network (3)

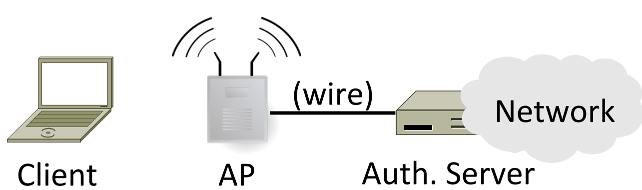
- Master key is from password; nonces for freshness
  - K_S lets client talk to AP; K_G lets AP talk to all clients



Computer Networks 39

Enterprise Network

- Network has authentication server
- Each client has own credentials
- AP lets client talk to auth. server
  - Grants network access if successful



Computer Networks 40

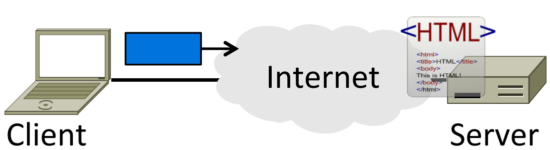
Introduction to Computer Networks

Web Security (§8.9.3, §8.5)

Computer Science & Engineering
UNIVERSITY of WASHINGTON

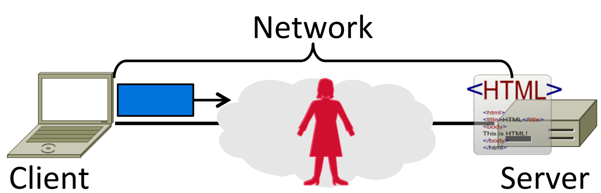
Goal and Threat Model

- Much can go wrong on the web!
  - Clients encounter malicious content
  - Web servers are target of break-ins
  - Fake content/servers trick users
  - Data sent over network is stolen …



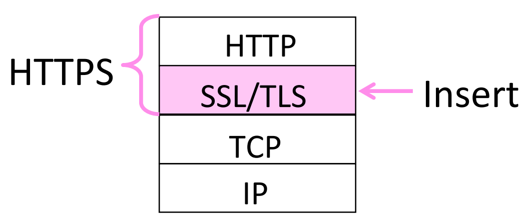
Goal and Threat Model (2)

- Goal of HTTPS is to secure HTTP
- We focus on network threats:
  1. Eavesdropping client/server traffic
  2. Tampering with client/server traffic
  3. Impersonating web servers



HTTPS Context

- HTTPS (HTTP Secure) is an add-on
  - Means HTTP over SSL/TLS
  - SSL (Secure Sockets Layer) precedes TLS (Transport Layer Security)



HTTPS Context (2)

- SSL came out of Netscape
  – SSL2 (flawed) made public in '95
  – SSL3 fixed flaws in '96
- TLS is the open standard
  – TLS 1.0 in '99, 1.1 in '06, 1.2 in '08
- Motivated by secure web commerce
  – Slow adoption, now widespread use
  – Can be used by any app, not just HTTP

SSL Operation

- Protocol provides:
  1. Verification of identity of server (and optionally client)
  2. Message exchange between the two with confidentiality, integrity, authenticity and freshness
- Consists of authentication phase (that sets up encryption) followed by data transfer phase

SSL/TLS Authentication

- Must allow clients to securely connect to servers not used before
  - Client must authenticate server
  - Server typically doesn't identify client
- Uses public key authentication
  - But how does client get server's key?
  - With certificates »

Certificates

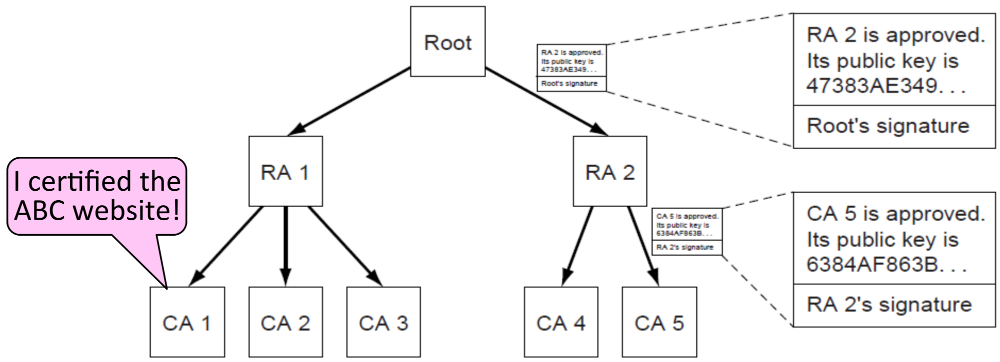
- A certificate binds public key to an identity, e.g., domain
  - Distributes public keys when signed by a party you trust
  - Commonly in a format called X.509

I hereby certify that the public key
19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A
belongs to
Robert John Smith
12345 University Avenue
Berkeley, CA 94702
Birthday: July 4, 1958
Email: bob@superdupernet.com

Signed by CA

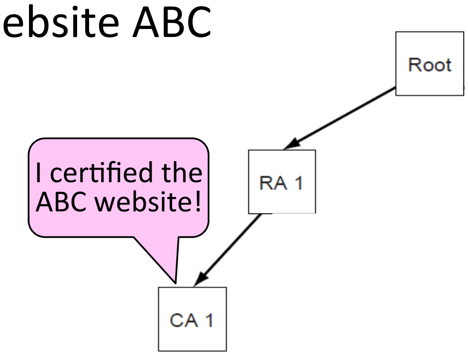
PKI (Public Key Infrastructure)

- Adds hierarchy to certificates to let many parties issue
  - Issuing parties are called CAs (Certificate Authorities)



PKI (2)

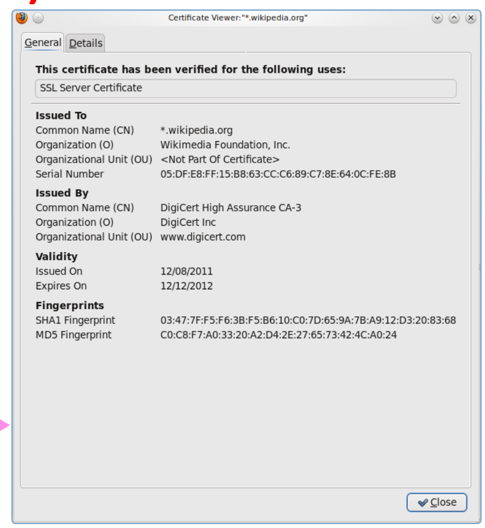
- Need public key of PKI root and trust in servers on path to verify a public key of website ABC
  - Browser has Root's public key
  - {RA1's key is X} signed Root
  - {CA1's key is Y} signed RA1
  - {ABC's key Z} signed CA1



PKI (3)

- Browser/OS has public keys of the trusted roots of PKI
  - >100 root certificates!
  - That's a problem …
  - Inspect your web browser

Certificate for wikipedia.org issued by DigiCert



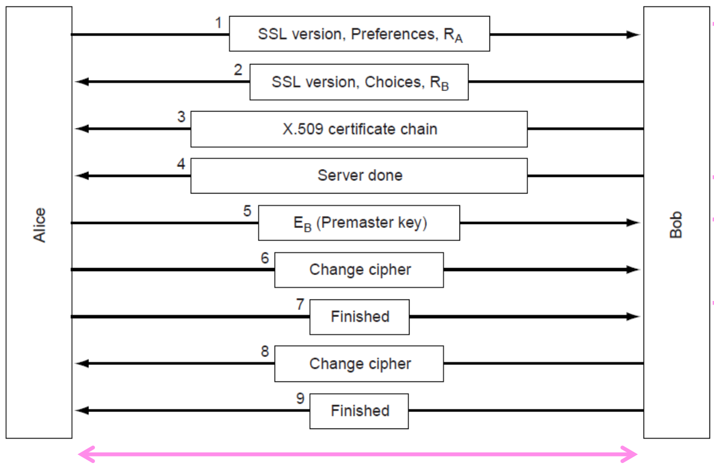
PKI (4)

- Real-world complication:
  - Public keys may be compromised
  - Certificates must then be revoked
- PKI includes a CRL (Certificate Revocation List)
  - Browsers use to weed out bad keys

SSL3 Authentication (2)

Certificate lets Alice check Bob

Switch to Alice's session key



Negotiate ciphers, send certificate, …

Real Bob can compute session key

Encrypted data

Encrypted data

Computer Networks 53

Introduction to Computer Networks

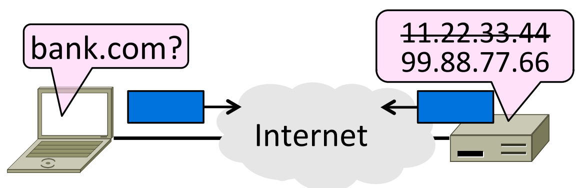
DNS Security (§8.9.2)

Computer Science & Engineering

UNIVERSITY *of* WASHINGTON

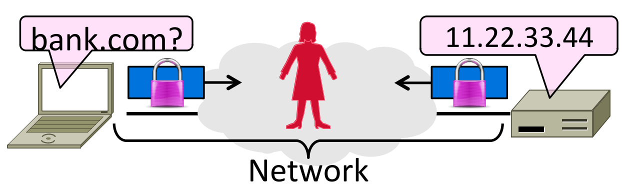
Goal and Threat Model

- Naming is a crucial Internet service
  - Binds host name to IP address
  - Wrong binding can be disastrous …



Goal and Threat Model (2)

- Goal is to secure the DNS so that the returned binding is correct
  - Integrity/authenticity vs confidentiality
- Attacker can intercept/tamper with messages on the network

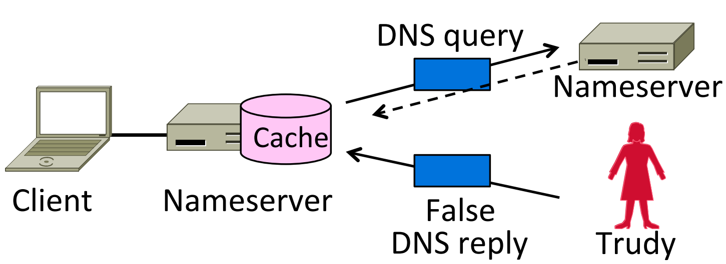


DNS Attacks

- How can a network attacker corrupt the DNS?

DNS Spoofing (2)

- To spoof, Trudy returns a fake DNS response that appears to be true
  - Fake response contains bad binding



DNS Spoofing (3)

- Lots of questions!
  1. How does Trudy know when the DNS query is sent and what it is for?
  2. How can Trudy supply a fake DNS reply that appears to be real?
  3. What happens when the real DNS reply shows up?
- There are solutions to each issue …

DNS Spoofing (4)

1. How does Trudy know when the query is sent and what it is for?

- Trudy can make the query herself!
  - Nameserver works for many clients
  - Trudy is just another client

DNS Spoofing (5)

2. How can Trudy supply a fake DNS reply that appears to be real?

- A bit more difficult. DNS checks:
  - Reply is from authoritative nameserver (e.g., .com)
  - Reply ID that matches the request
  - Reply is for outstanding query
- (Nothing about content though …)

DNS Spoofing (6)

2. How can Trudy supply a fake DNS reply that appears to be real?

- Techniques:
  - Put IP of authoritative nameserver as the source IP address
  - ID is 16 bits (64K). Send many guesses! (Or if a counter, sample to predict.)
  - Send reply right after query
- Good chance of succeeding!

DNS Spoofing (7)

3. What happens when the real DNS reply shows up?

- Likely not be a problem
  - There is no outstanding query after fake reply is accepted
  - So real reply will be discarded

DNSSEC (DNS Security Extensions)

- Extends DNS with new record types
  - RRSIG for digital signatures of records
  - DNSKEY for public keys for validation
  - DS for public keys for delegation
  - First version in ‘97, revised by ’05
- Deployment requires software upgrade at both client and server
  - Root servers upgraded in 2010
  - Followed by uptick in deployment

DNSSEC (2) – New Records

- As well as the usual A, NS records:
- RRSIG
  - Digital signatures of domain records
- DNSKEY
  - Public key used for domain RRSIGs
- DS
  - Public keys for delegated domain
- NSEC/NSEC3
  - Authenticated denial of existence

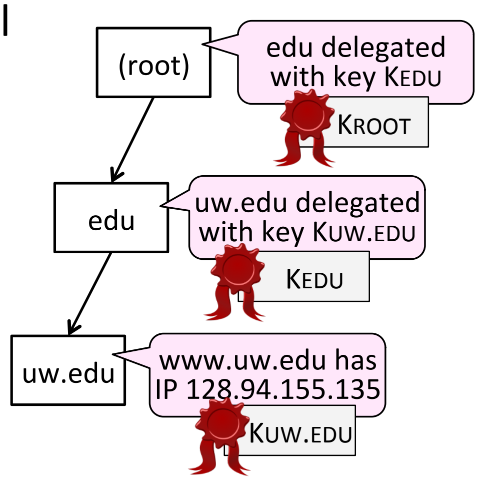
DNSSEC (3) – Validating Replies

- Clients query DNS as usual, then validate replies to check that content is authentic
- Trust anchor is root public keys
  - Part of DNS client configuration
- Trust proceeds down DNS hierarchy
  - Similar concept to SSL certificates

DNSSEC (4) – Validating Replies

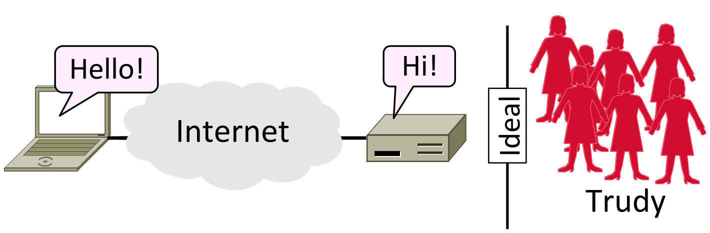
Client queries www.uw.edu as usual
- Replies include signatures/keys

Client validates answer:
1. KROOT is a trust anchor
2. Use KROOT to check KEDU
3. Use KEDU to check KUW.EDU
4. Use KUW.EDU to check IP



Goal and Threat Model

- Goal is for host to keep network connectivity for desired services
  - Threat is Trudy may overwhelm host with undesired traffic

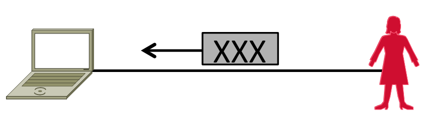


Internet Reality

- Distributed Denial-of-Service is a huge problem today!
  - Akamai Q3-12 reports DDOS against US banks peaking at 65 Gbps …
- There are no great solutions
  - CDNs, network traffic filtering, and best practices all help

Host Denial-of-Service

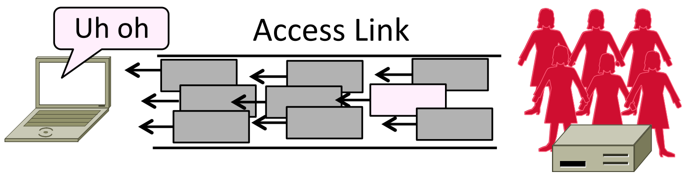
- Strange packets can sap host resources!
  - "Ping of Death" malformed packet
  - "SYN flood" sends many TCP connect requests and never follows up
  - Few bad packets can overwhelm host



- Patches exist for these vulnerabilities
  - Read about "SYN cookies" for interest

Network Denial-of-Service

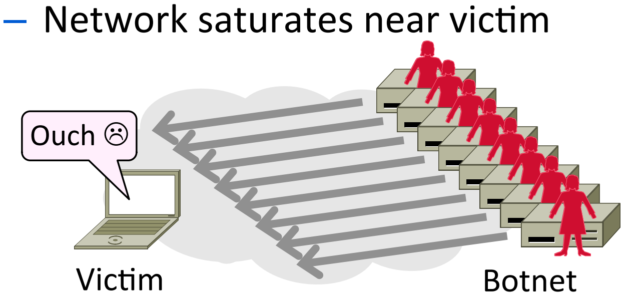
- Network DOS needs many packets
  - To saturate network links
  - Causes high congestion/loss



- Helpful to have many attackers … or Distributed Denial-of-Service

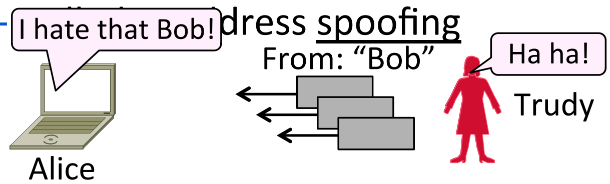
Distributed Denial-of-Service (DDOS)

- Botnet provides many attackers in the form of compromised hosts
  - Hosts send traffic flood to victim
  - Network saturates near victim



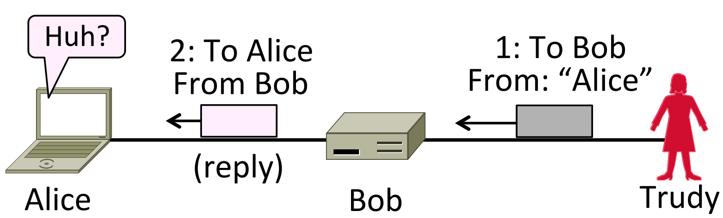
Complication: Spoofing

- Attackers can falsify their IP address
  - Put fake source address on packets
  - Historically network doesn't check
  - Hides location of the attackers
  - dress spoofing



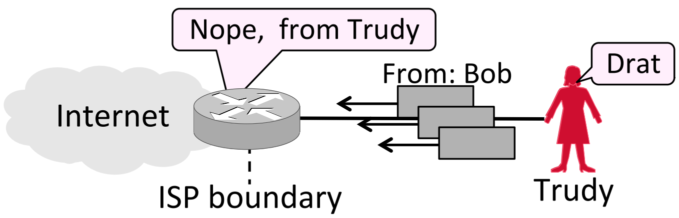
Spoofing (2)

- Actually, it's worse than that
  - Trudy can trick Bob into really sending packets to Alice
  - To do so, Trudy spoofs Alice to Bob



Best Practice: Ingress Filtering

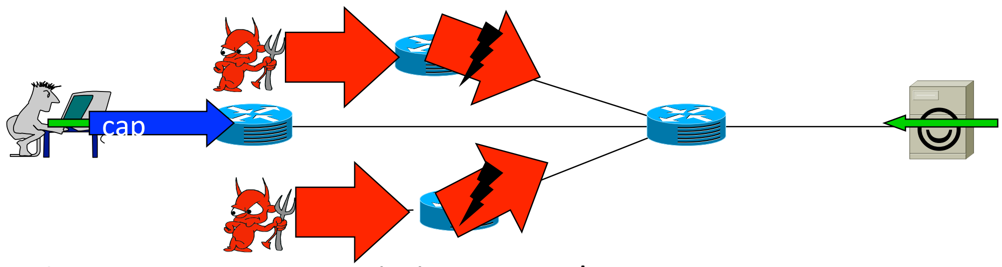
- Idea: Validate the IP source address of packets at ISP boundary (Duh!)
  - Ingress filtering is a best practice, but deployment has been slow



Flooding Defenses

1. Increase network capacity around the server; harder to cause loss
   - Use a CDN for high peak capacity
2. Filter out attack traffic within the network (at routers)
   - The earlier the filtering, the better
   - Ultimately what is needed, but ad hoc measures by ISPs today

Sketch of the capability approach



1. Source requests permission to send.
2. Destination authorizes source for limited transfer, e.g, 32KB in 10s
   - A capability is the proof of a destination's authorization.
3. Source places capabilities on packets and sends them.
4. Network filters packets based on capabilities.

77

Capabilities alone do not effectively limit DoS

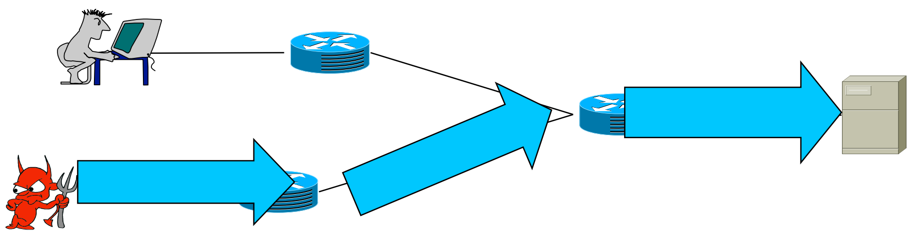
Goal: minimize the damage of the arbitrary behavior of k attacking hosts.

Problems

1. Request or authorized packet floods
2. Added functionality in a router's forwarding path
3. Authorization policies
4. Deployment

78

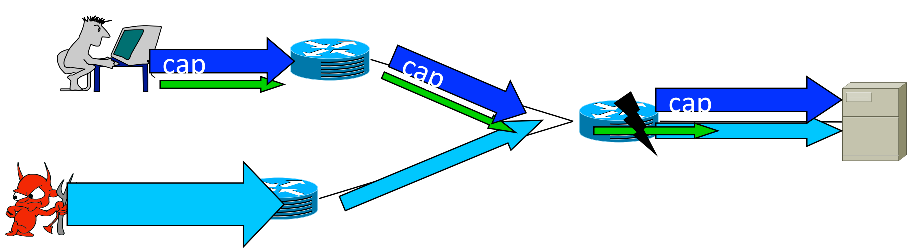
Request packet floods



- Request packets do not carry capabilities.

79

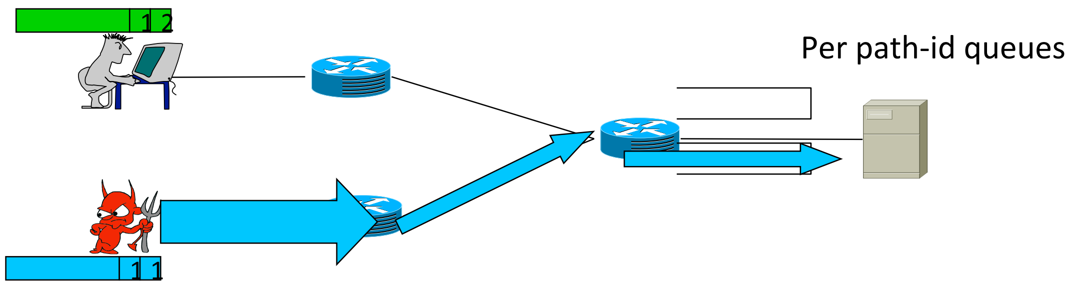
Counter request packet floods (I)



- Rate-limit request packets

80

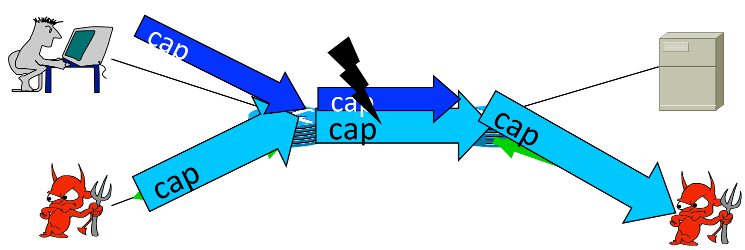
Counter request packet floods (II)



- Rate-limit request packets
- Routers insert path identifier tags
- Fair queue requests using the most recent tags

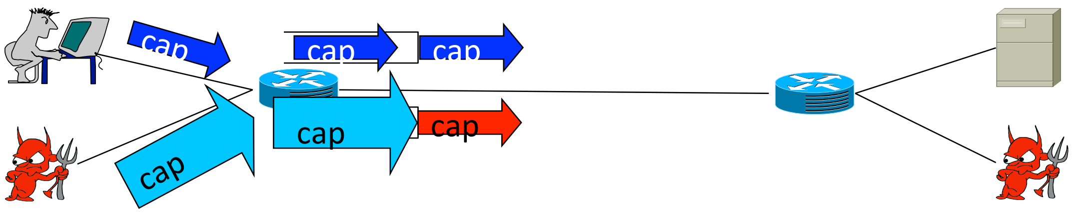
81

Authorized packet floods



82

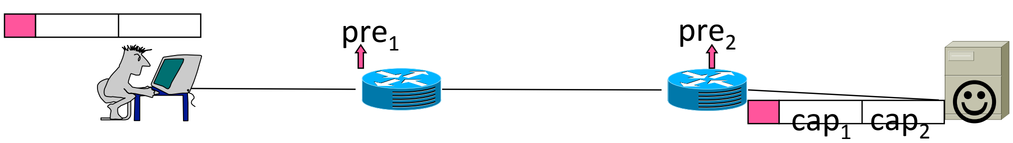
Counter authorized packet floods



- Per-destination queues
- TVA bounds the number of queues.

83

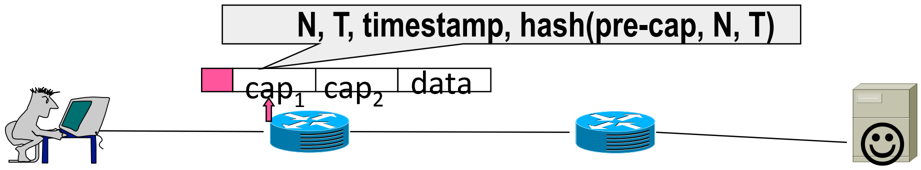
TVA's implementation of capabilities



- Routers stamp pre-capabilities on request packets
  - **(timestamp, hash(src, dst, key, timestamp)**
- Destinations return fine-grained capabilities
  - **(N, T, timestamp, hash(pre-cap, N, T))**
  - send N bytes in the next T seconds, e.g. 32KB in 10 seconds

84

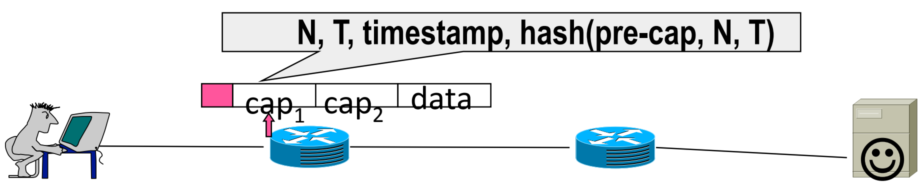
Validating fine-grained capabilities



1. A router verifies that the hash value is correct.
2. Checks for expiration: *timestamp + T · now*
3. Checks for byte bound: *sent + pkt_len · N*

85

Bounded state

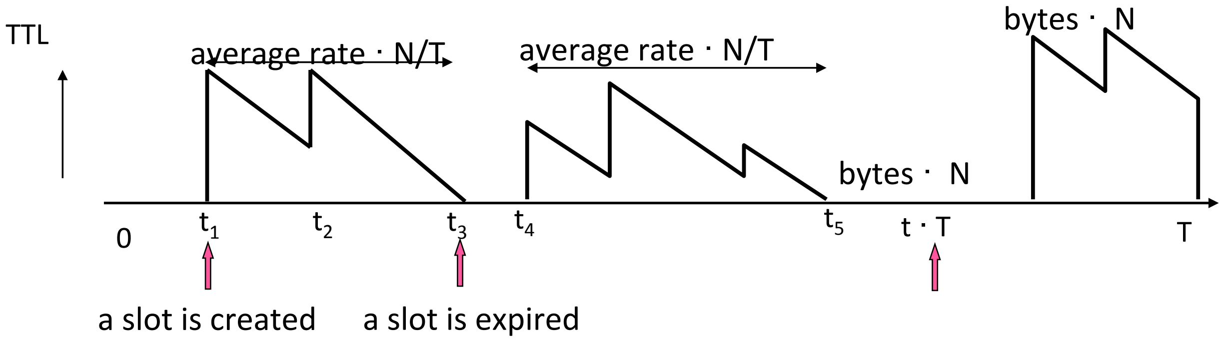


sent + pkt_len · N

- Create a slot if a capability sends faster than N/T.
- For a link with a fixed capacity C, there are at most C/(N/T) flows
- → Number of slots is bounded by C / (N/T)

86

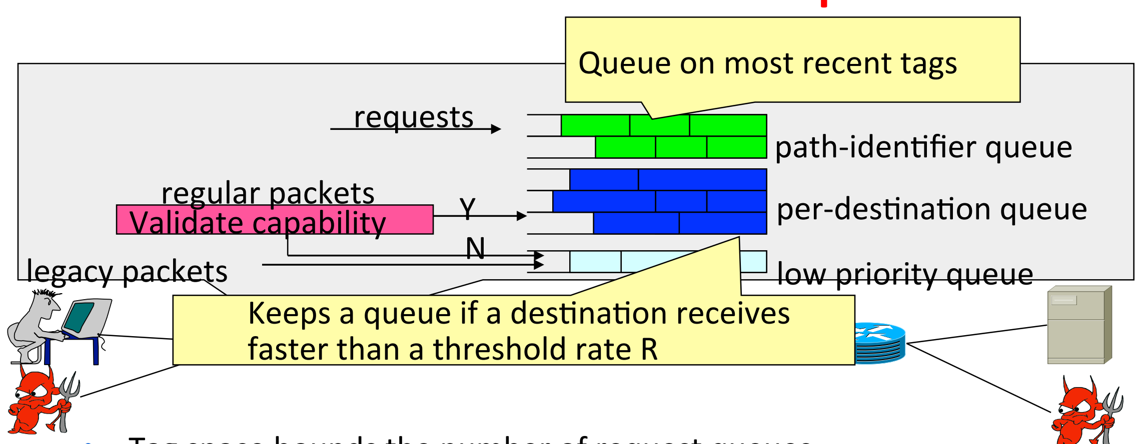
Worst case byte bound is 2N in T seconds



- If a slot expires, it indicates that a capability sends slower than N/T.

87

Bounded number of queues



- Tag space bounds the number of request queues.
- Number of destination queues is bounded by C/R

88

TVA Summary

- Key contribution
  - a comprehensive and practical capability system for the first time.
- TVA is practical in three aspects
  - Counter a broad range of attacks
  - Bounded state and computation
  - Simple and effective authorization policies
- But requires comprehensive changes to the Internet

89