CS306: Introduction to IT Security

Fall 2018

Lecture 10: Topics

Instructor: **Nikos Triandopoulos**

November 13, 2018

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

CS306: Tentative Syllabus

Week	Date	Topics	Reading	Assignment
1	Aug 28	Introduction	Ch. 1	-
2	Sep 4	Symmetric encryption	Ch. 2 & 12	Lab 1
3	Sep 11	Symmetric encryption II	Ch. 2 & 12	Lab 2, HW 1
4	Sep 18	Message authentication	Ch. 2 & 12	Lab 3, HW 1
5	Sep 25	Hash functions	Ch. 2 & 12	Lab 4
6	Oct 2	Public-key cryptography	Ch. 2 & 12	Lab 5
–	Oct 9	No class (Monday schedule)		Help session
7	Oct 16	Midterm (closed books)	All materials covered	No labs

CS306: Tentative Syllabus (continued)

Week	Date	Topics	Reading	Assignment
8	Oct 23	User authentication	Ch. 2	No labs
9	Oct 30	Access Control	Ch. 2	Lab 6
10	Nov 6	Cloud & Network security	Ch. 6	Lab 7
11	**Nov 13**	**Network security**	**Ch.6**	**Lab 8, HW2, HW3**
12	Nov 20	Web, Software & Database		
13	Nov 27	Privacy & Economics		
14	Dec 4	Legal & ethical issues		
15	Dec 11 (or later)	**Final** (closed books)	All materials covered*	

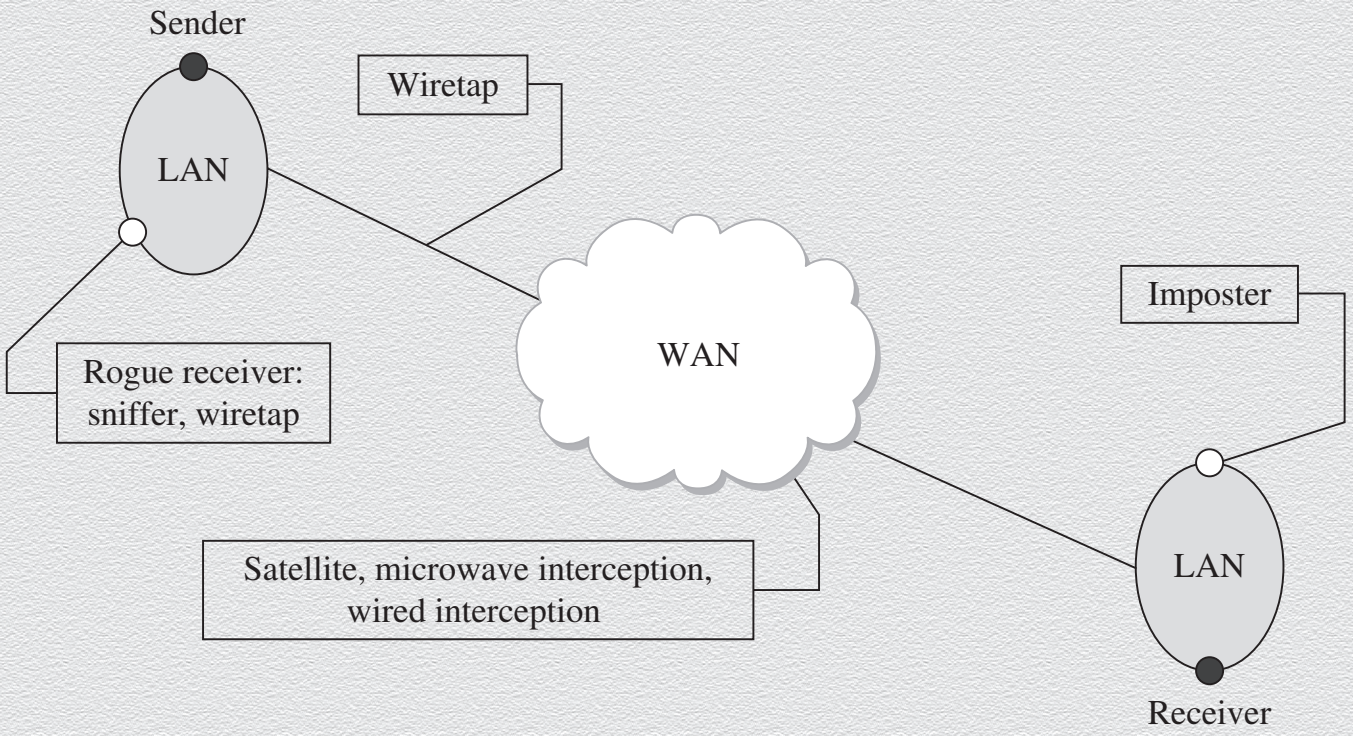
* w/ focus on what covered after midterm

Network security: Issues & countermeasures

Network transmission media

- Cable
- Optical fiber
- Microwave
- WiFi
- Satellite communication

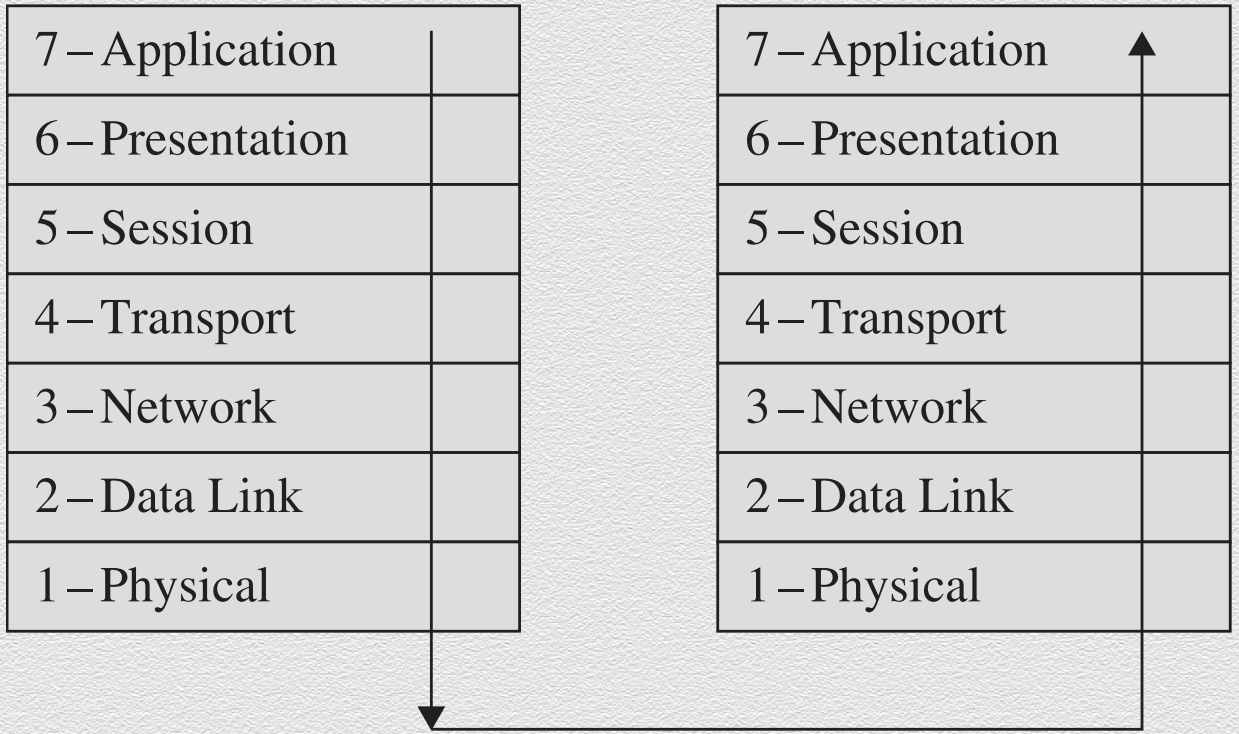
Communication media vulnerabilities



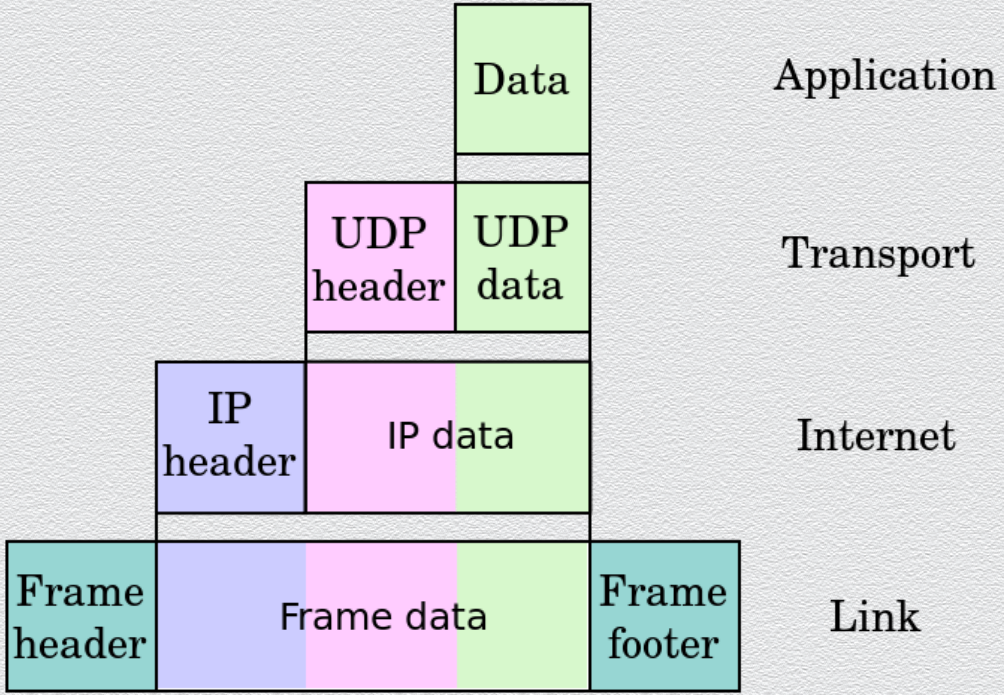
Communication media “Pros Vs. Cons”

Medium	Strengths	Weaknesses
Wire	• Widely used • Inexpensive to buy, install, maintain	• Susceptible to emanation • Susceptible to physical wiretapping
Optical fiber	• Immune to emanation • Difficult to wiretap	• Potentially exposed at connection points
Microwave	• Strong signal, not seriously affected by weather	• Exposed to interception along path of transmission • Requires line of sight location • Signal must be repeated approximately every 30 miles (50 kilometers)
Wireless (radio, WiFi)	• Widely available • Built into many computers	• Signal degrades over distance; suitable for short range • Signal interceptable in circular pattern around transmitter
Satellite	• Strong, fast signal	• Delay due to distance signal travels up and down • Signal exposed over wide area at receiving end

The OSI Model



Data encapsulation example: UDP



Threats to network communications

- Interception, or unauthorized viewing
- Modification, or unauthorized change
- Fabrication, or unauthorized creation
- Interruption, or preventing authorized access

Why networks are vulnerable to interception?

Anonymity

- An attacker can attempt many attacks, anonymously, from thousands of miles away

Many points of attack

- Large networks mean many points of potential entry

Sharing

- Networked systems open up potential access to more users than do single computers

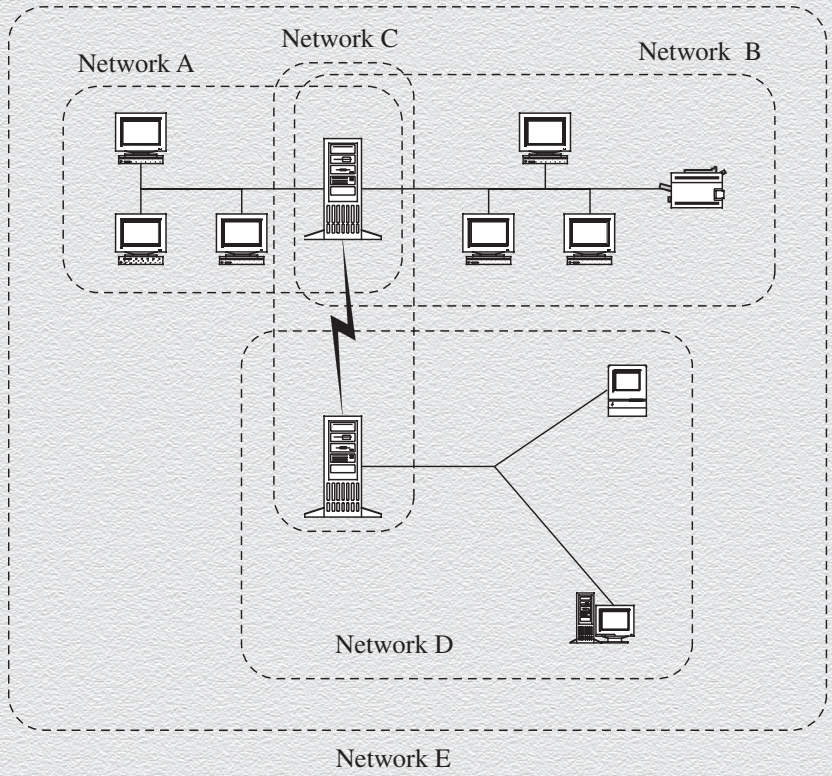
System complexity

- Very complex & hard to protect: different systems/purposes, disparate OS/vulnerabilities

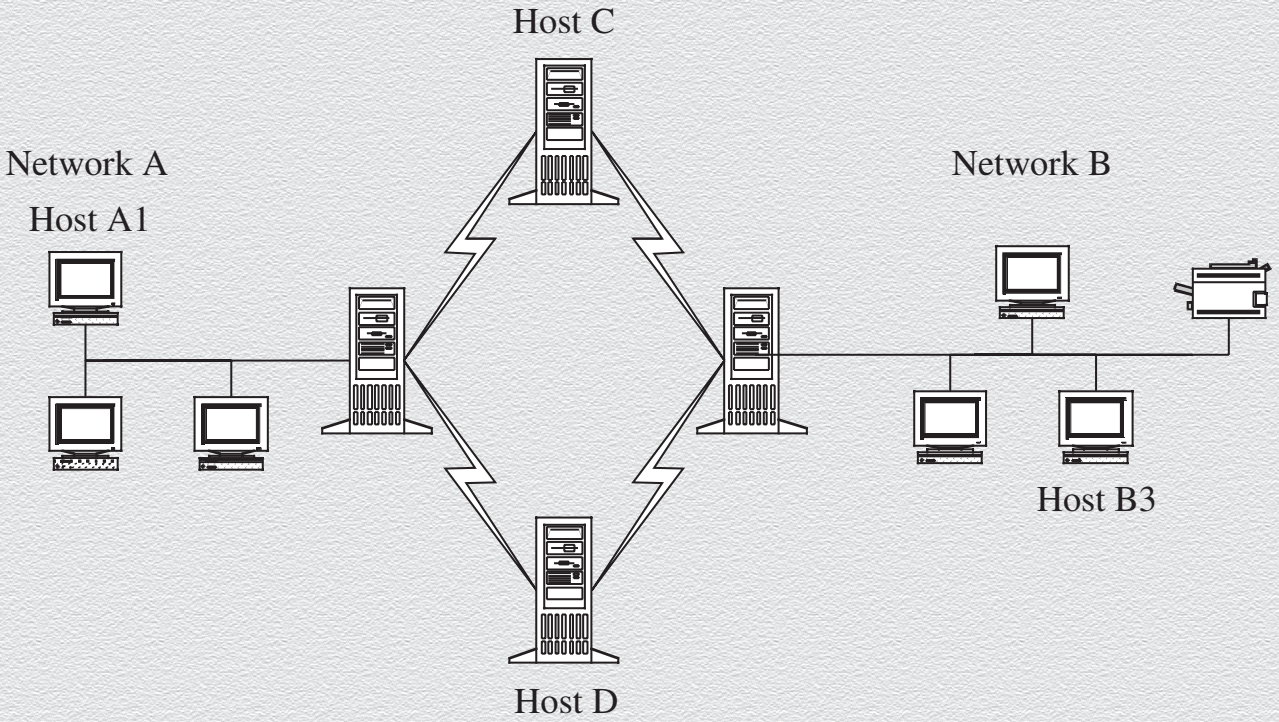
Unknown perimeter/path

- Fast changing configurations, topologies, many source-to-destination paths

Unknown perimeter



Unknown path



Modification & fabrication

Data corruption

- May be intentional or unintentional, malicious or nonmalicious, directed or random

Sequencing

- Permuting the order of data, such as packets arriving in sequence

Substitution

- Replacement of one piece of a data stream with another

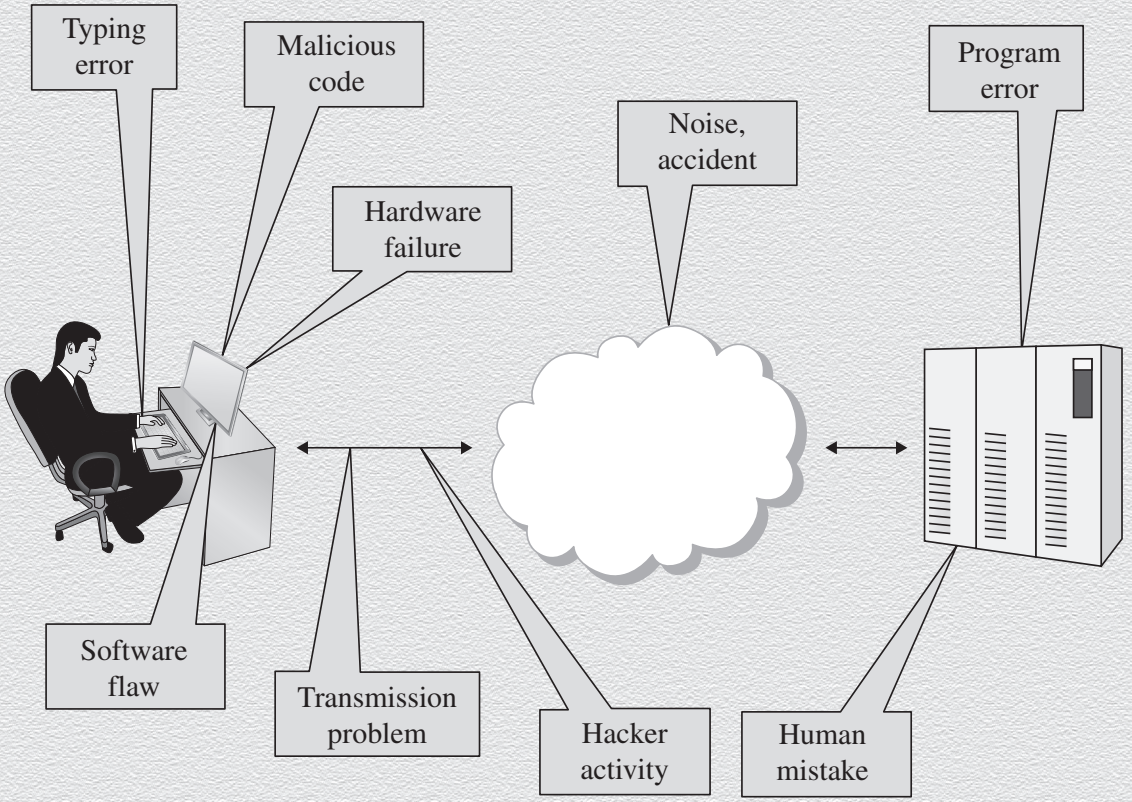
Insertion

- A form of substitution in which data values are inserted into a stream

Replay

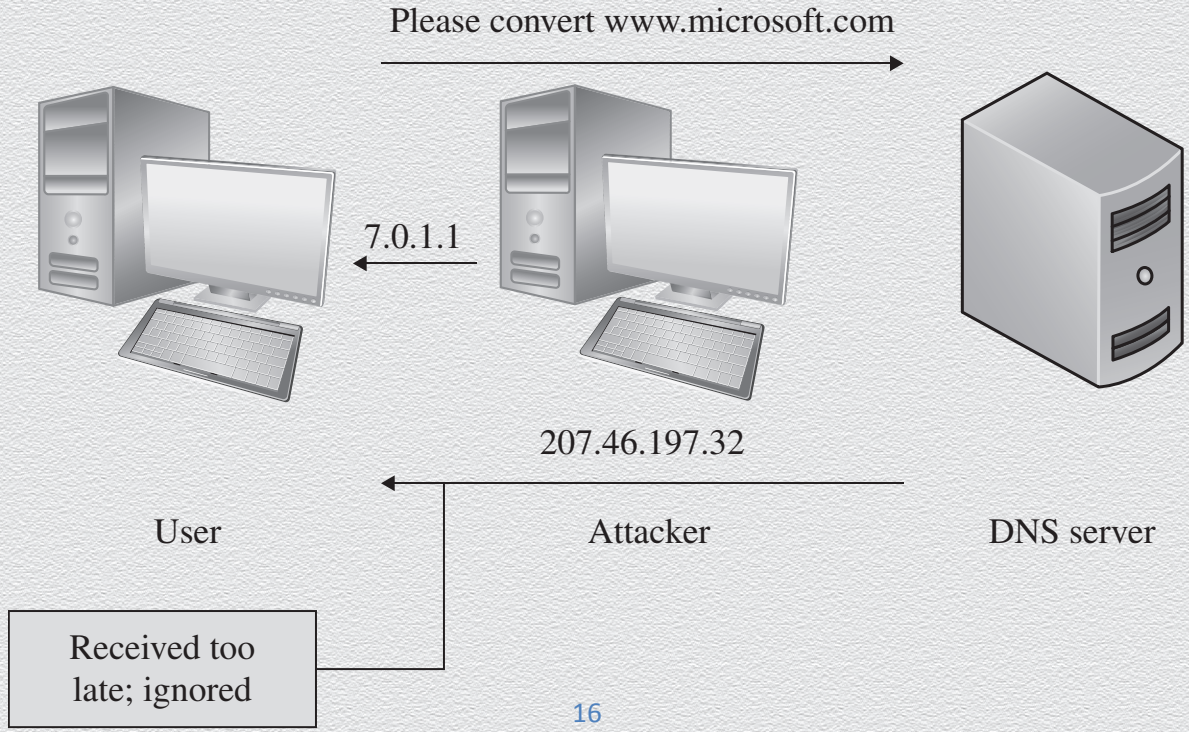
- Legitimate data are intercepted and reused

Sources of data corruption

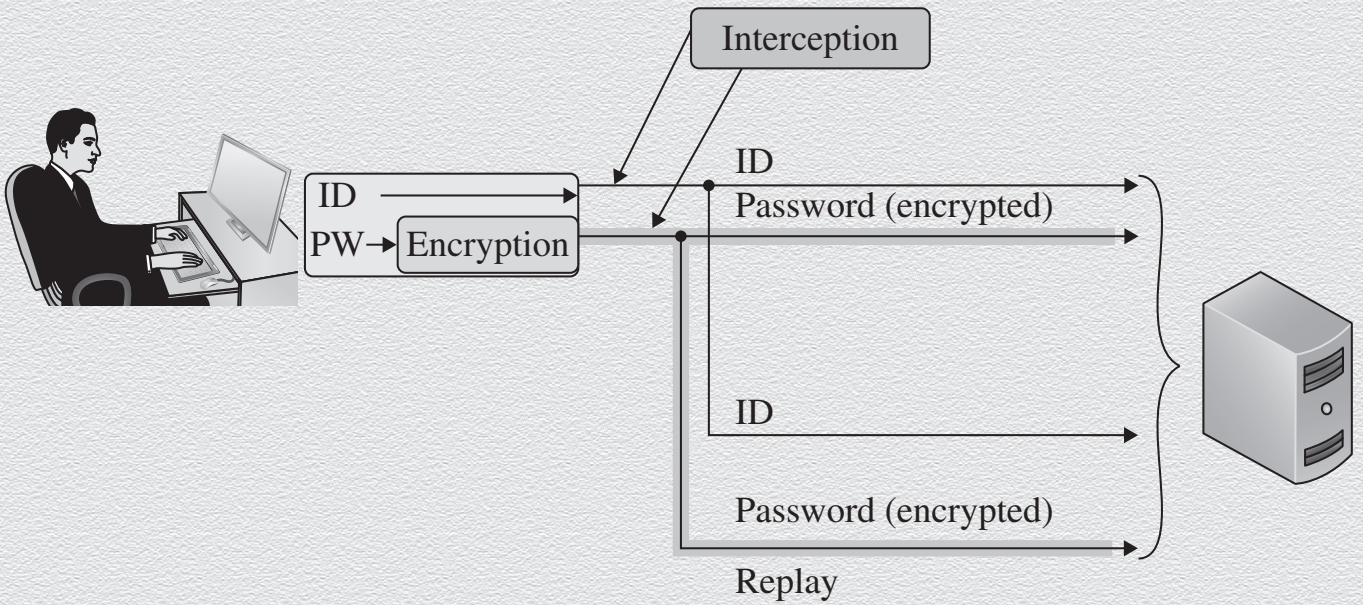


DNS spoofing (or cache poisoning)

The attacker acts as the DNS server in order to redirect the user to malicious sites



Simple replay attack



Interruption: Loss of Service

Routing

- Internet routing protocols are complicated, and one misconfiguration can poison the data of many routers

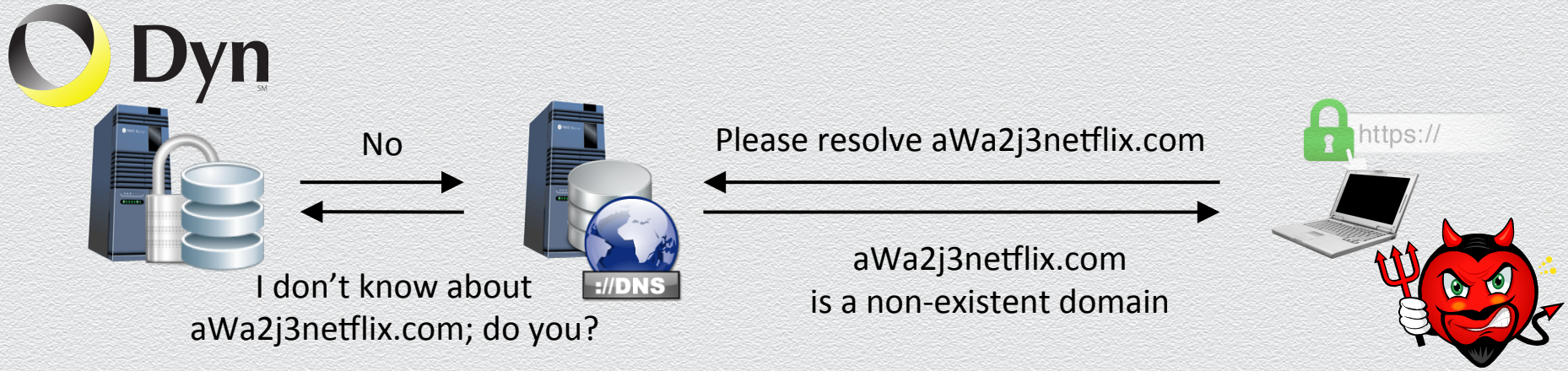
Excessive demand

- Network capacity is finite and can be exhausted; an attacker can generate enough demand to overwhelm a critical part of a network

Component failure

- Component failures tend to be sporadic and unpredictable, and will cause loss of service if not planned for

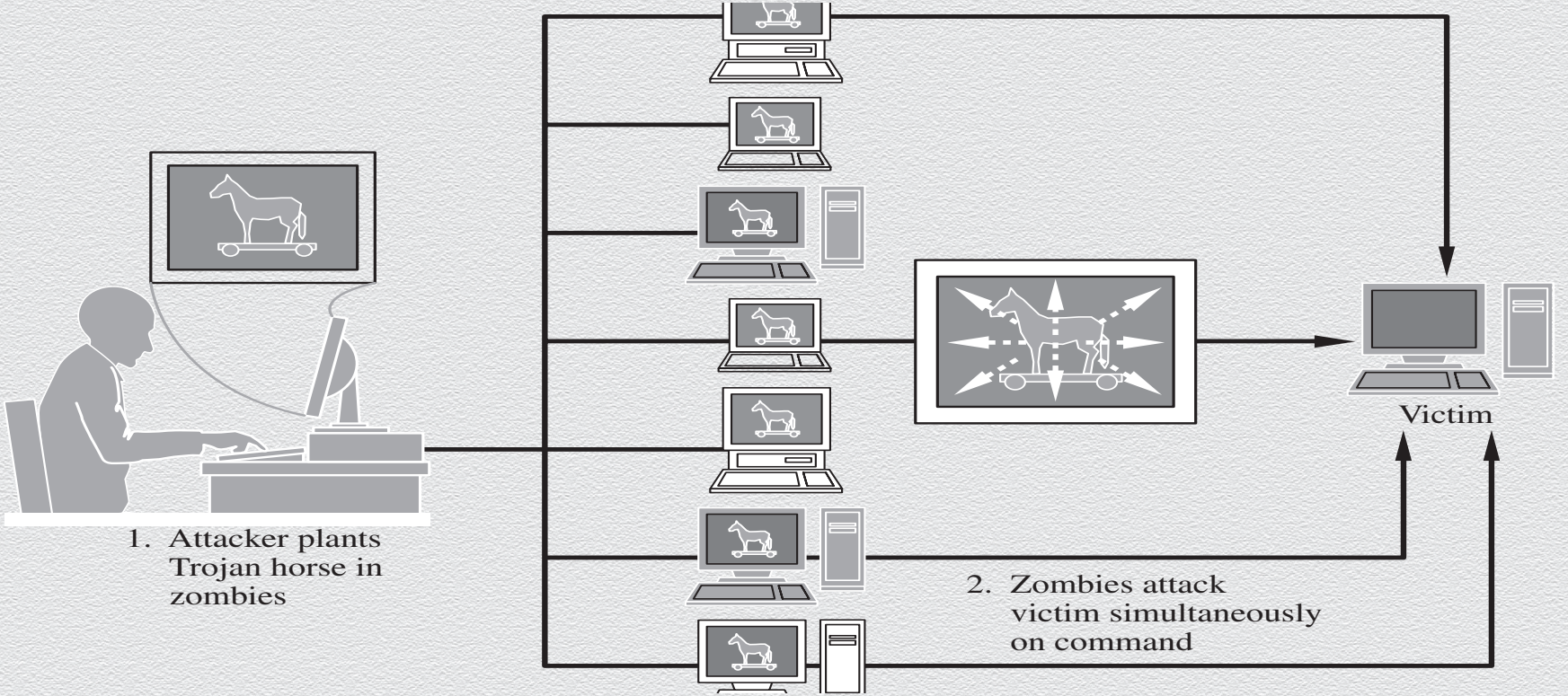
Recall: Core idea in Dyn's DDoS attack



Attack: Saturate primary (authoritative) DNS servers

- from a compromised machine ask for domain names that do not exist
- query is forwarded to fewer primary Dyn servers, i.e., defeating benefits of distribution
- ask **A LOT** of such queries to bring down the Dyn DNS service!

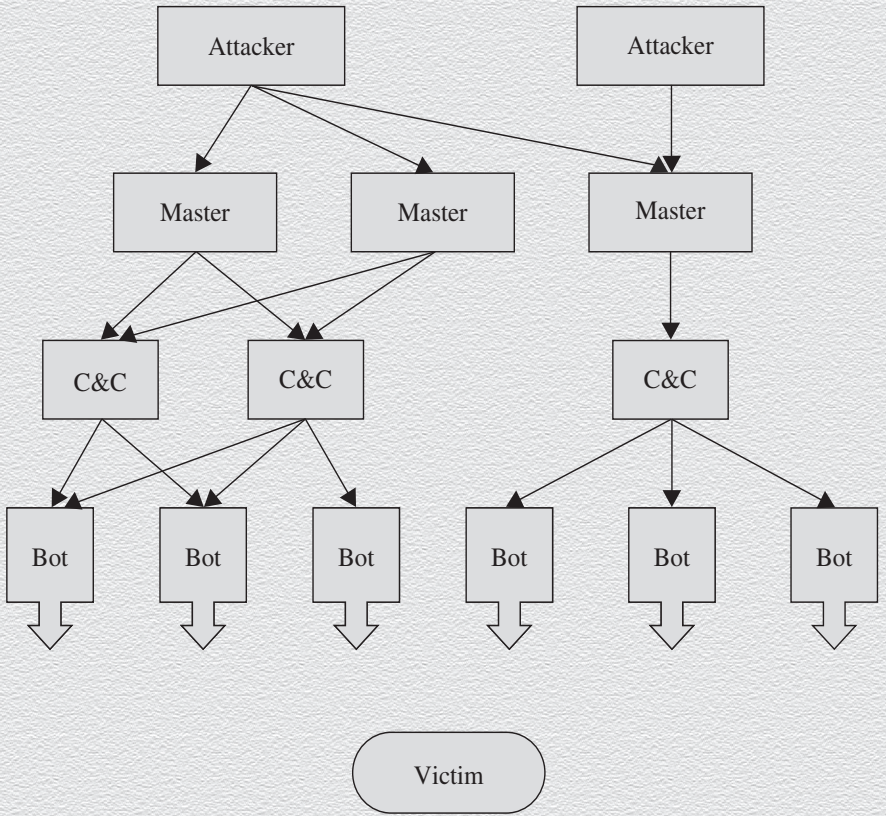
Distributed Denial of Service (DDoS)



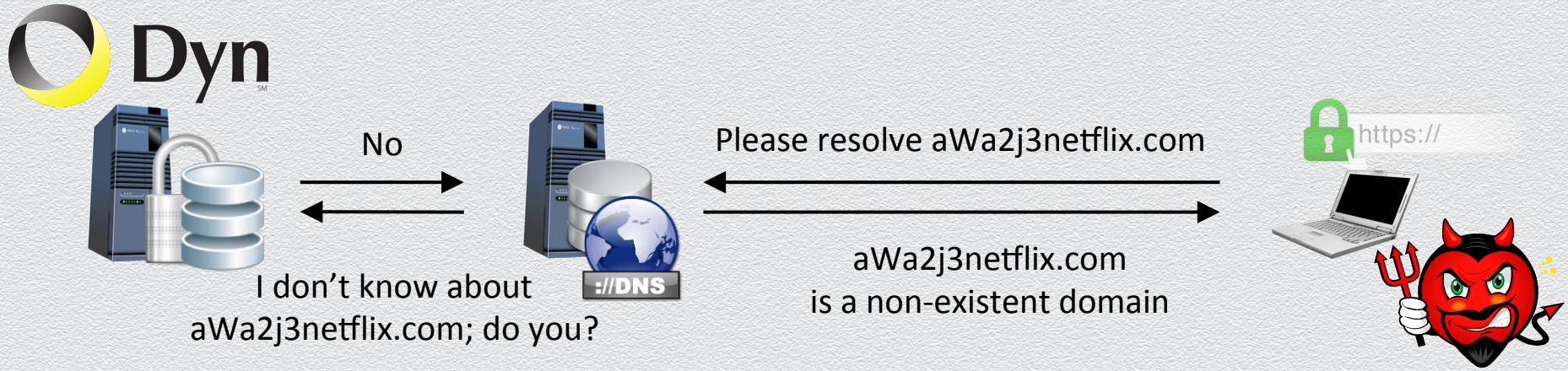
Botnets

Networks of machines running malicious code under remote control

- massive: scale to million of bots
  - comprise main tool for DDoS attacks
- stealth: remain undetected & difficult to trace
  - do little harm to the host machines
    - users won't likely remove malware
- resilient: have redundant components
  - multiple-level attacker Vs. bots separation
  - even if one master or C&C node is taken down, connectivity is maintained



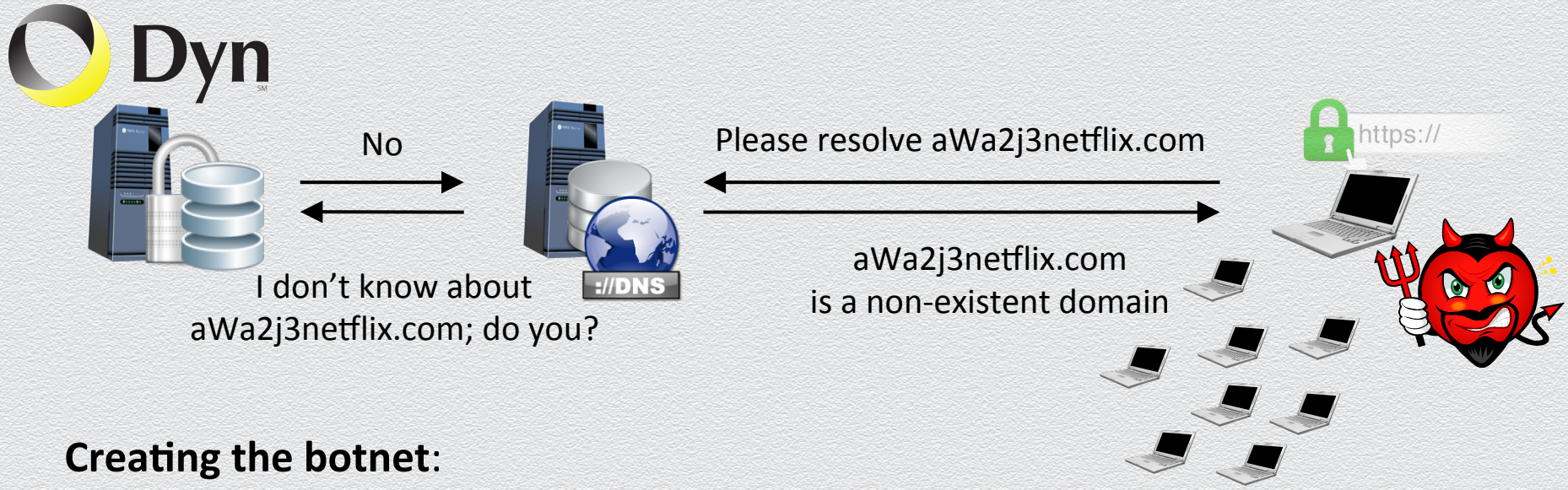
Why botnets are often behind DoS attacks?



Goal: Avoid effective countermeasures & increase "attack" traffic

- if the high-volume "attack" traffic comes from few devices, they can be filtered out by blocking their connections to the Dyn servers
- by employing a large botnet of millions of devices the attacker inflicts a larger, more devastating "attack" traffic against the victim Dyn servers

Recruiting an army: Internet of Things (IoT)



Creating the botnet:

- compromise easy targets: IoT "thin" devices, e.g., printers, cameras, home routers, …
- how? find a vulnerability on these devices...
  - all such devices used an OS with a static, hard-wired, thus known, admin password...!

The Internet of Things (IoT)

Refers to Internet-connected everyday devices

- comprise a world of so-called smart devices, such as:
  - smart appliances, such as refrigerators and dishwashers
  - smart home, such as thermostats and alarm systems
  - smart health, such as fitness monitors and insulin pumps
  - smart transportation, such as driverless cars
  - smart entertainment, such as video recorders
- potential downsides
  - loss of privacy
  - loss of control of data
  - potential for subversion
  - mistaken identification
  - uncontrolled access

Smartphones

The control hub of the IoT – important target for malware

- 2013: 143,211 distinct new forms of malware against mobile devices
- 98% targeted Android devices, far in excess of its market share
  - Android: open approach
    - unlike its competitors, does not limit the software users are allowed to install
    - thus, an easier target
  - Apple: locked-down approach
    - in contrast, only allows apps from its app store to be installed on its smartphones
    - all apps go through an approval process, which includes some security review
    - once approved, apps are signed, using a certificate approach

More generally on DoS attacks

DoS attacks are attempts to defeat a system's availability

- volumetric attacks (e.g., flooding)
  - potential weaknesses in the capacity of a computer network

insufficient capacity, overloading

- application-based attacks (e.g., routing malfunction, blocked access)
  - exhaust the resources of an application that services a particular network

blocked access

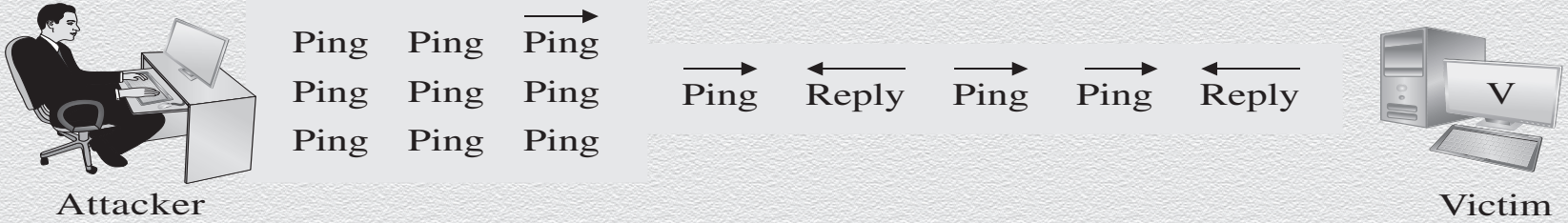
- disabled communications (e.g., access failure)
  - physically disable a communication link or disrupt a single-point of failure
- hardware or software failure (e.g., access failure)
  - failures on machines or programs (without fault-tolerance provisions)

unresponsive asset

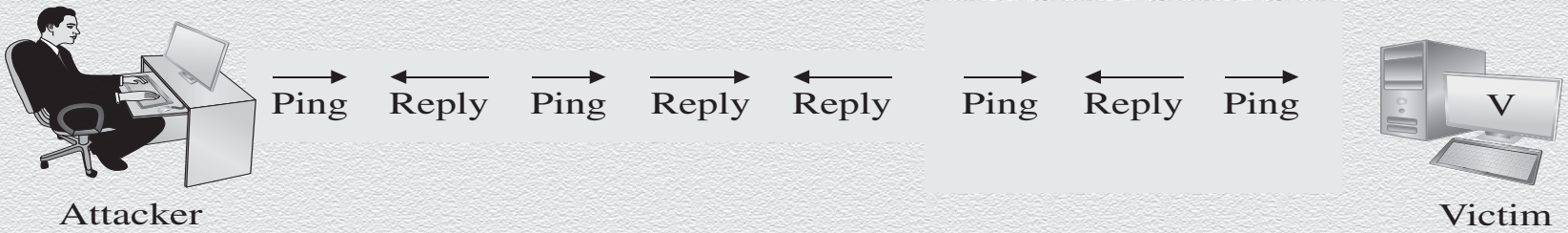
Examples

- Benign errors
  - Beth Israel Hospital system downtime, in 2002, due to mishandling of switches
- Malicious code
  - use vulnerabilities in communication protocols
  - e.g., cause uncontrolled congestion in TCP communications (Vs. UDP)
  - e.g., exploit/misuse Internet Control Message Protocols (ICMP)
    - ping (destination is reachable and functional)
    - echo (the connection between two machines is reliable)
    - destination unreachable (destination address cannot be accessed)
    - source quench (destination is saturated and source should suspend transmissions)

Ping flood (or pink of death)



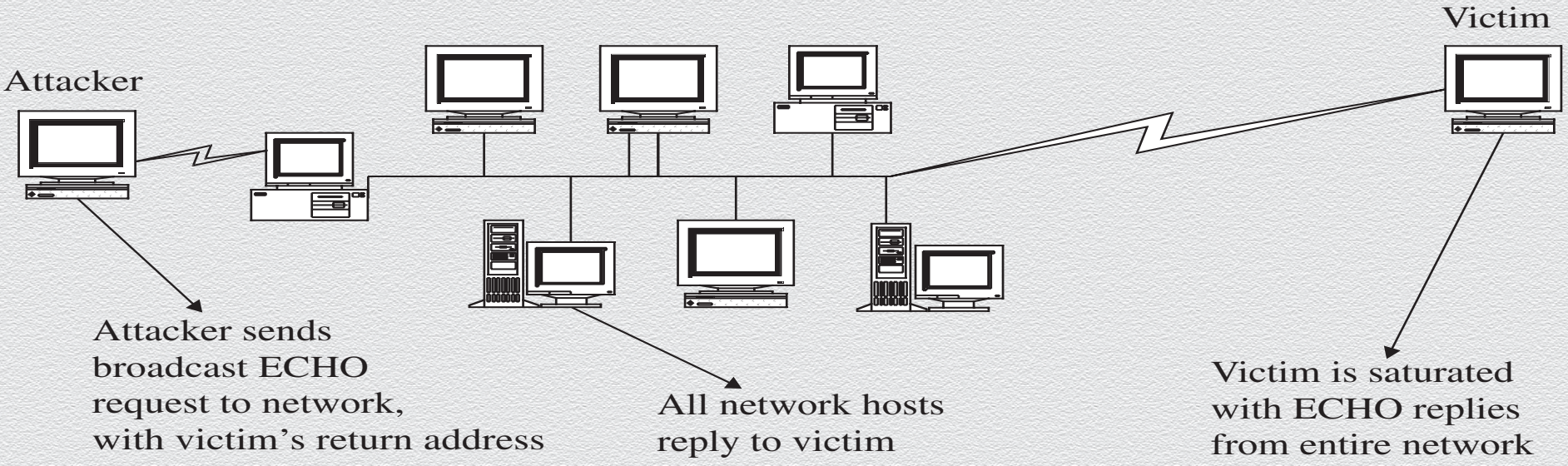
(a) Attacker has greater bandwidth



(b) Victim has greater bandwidth

Smurf Attack

Unwitting victims become accomplices



Echo-Chargen

Victim A

Victim B

Chargen packet with echo bit on →

← Echoing what you just sent me

Chargen another packet with echo bit on →

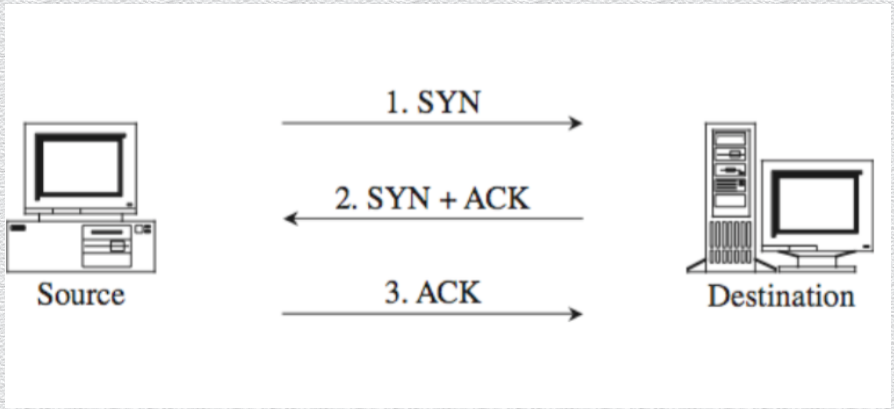
← Echoing that again

Chargen another packet with echo bit on →

SYN flood

Three-way handshake used in TCP to establish a new session

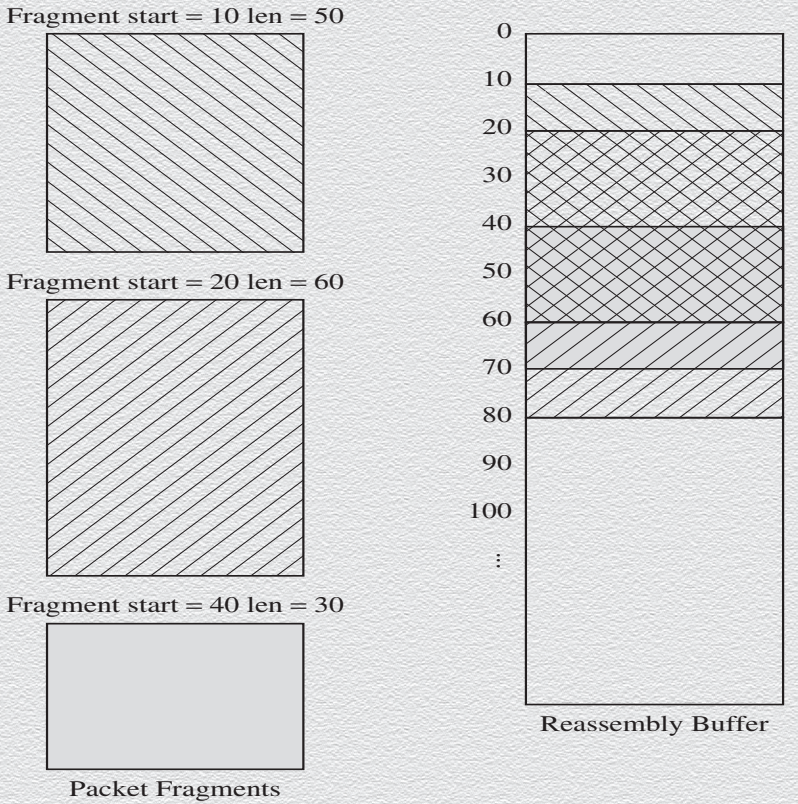
- source & destination exchange control messages to complete session creation
- destination keeps track of incomplete session-creation protocols (SYN-RECV cache)
- attacker spoofs return address of many SYN handshake messages sent to victim
- victim's cache is filled (after ~20) pending incomplete sessions and delays are created



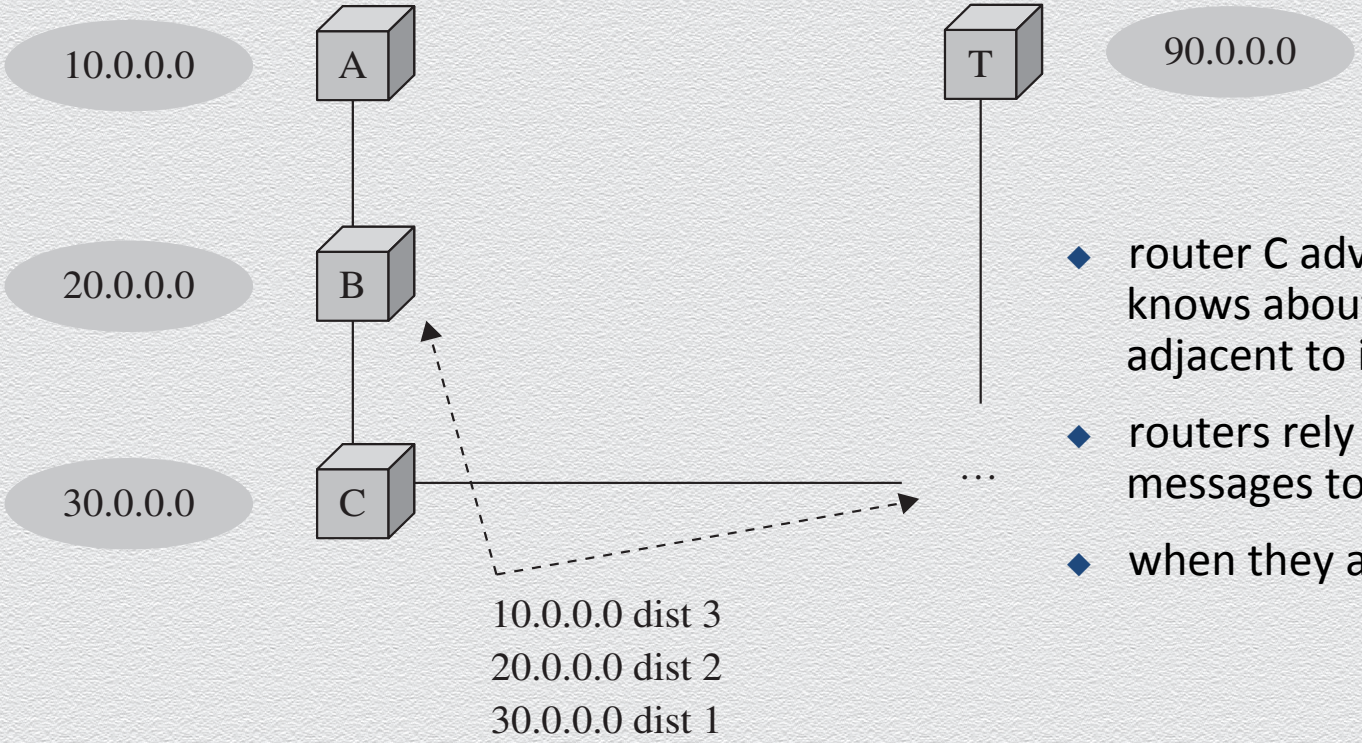
Teardrop Attack

IP datagrams allow to carry variable-length data

- each datagram specifies the length of its data payload and its offset (start-end)
- the attacker can spoof these metadata so that recipient's state remains inconsistent
- sent packets cannot possibly be reassembled, as they conflict instructions
- in extreme cases, this can cause the entire OS to lock up



Rerouting routing

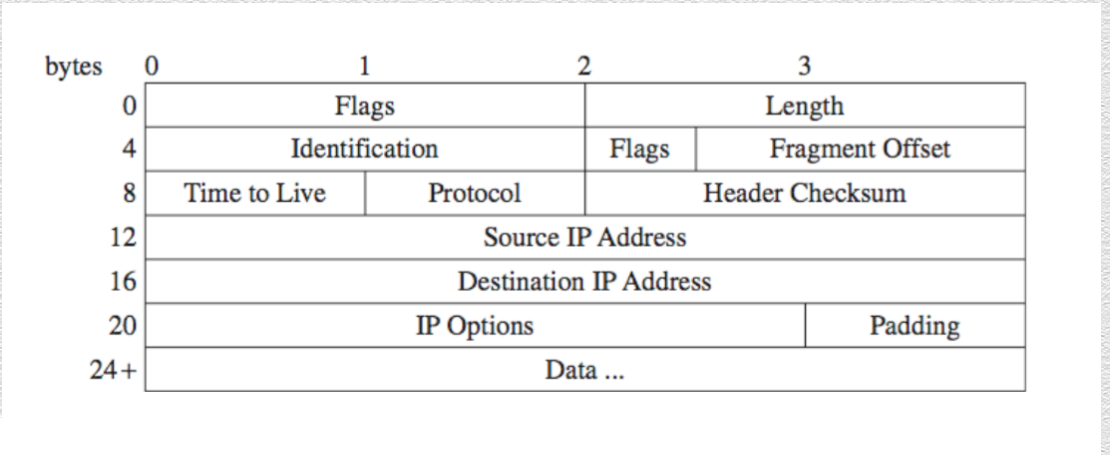


- router C advertises the routes it knows about to the routers adjacent to it
- routers rely on these advertising messages to be accurate
- when they aren't, DoS can ensue

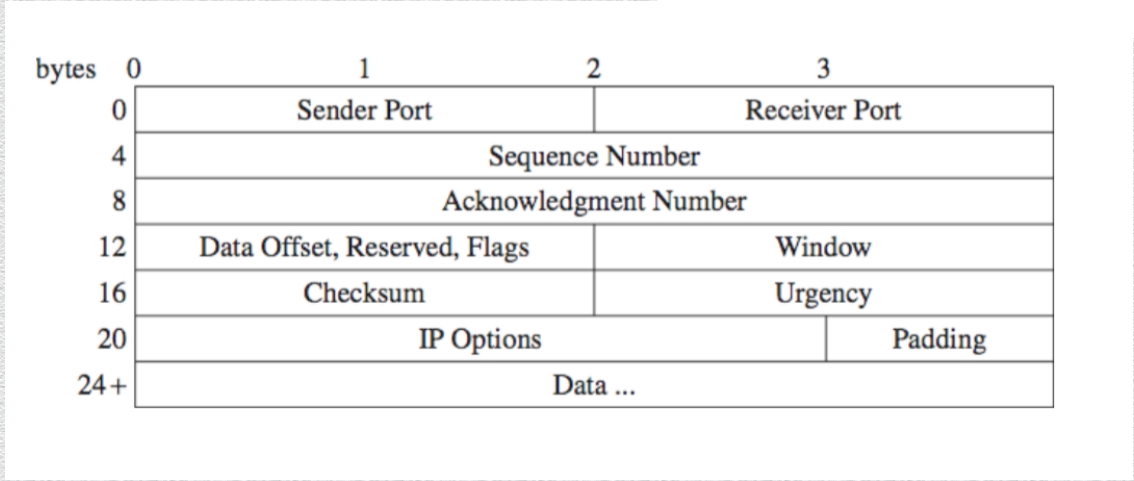
TCP/IP headers

headers of IP datagram & TCP packet

- IP: right
- TCP: below



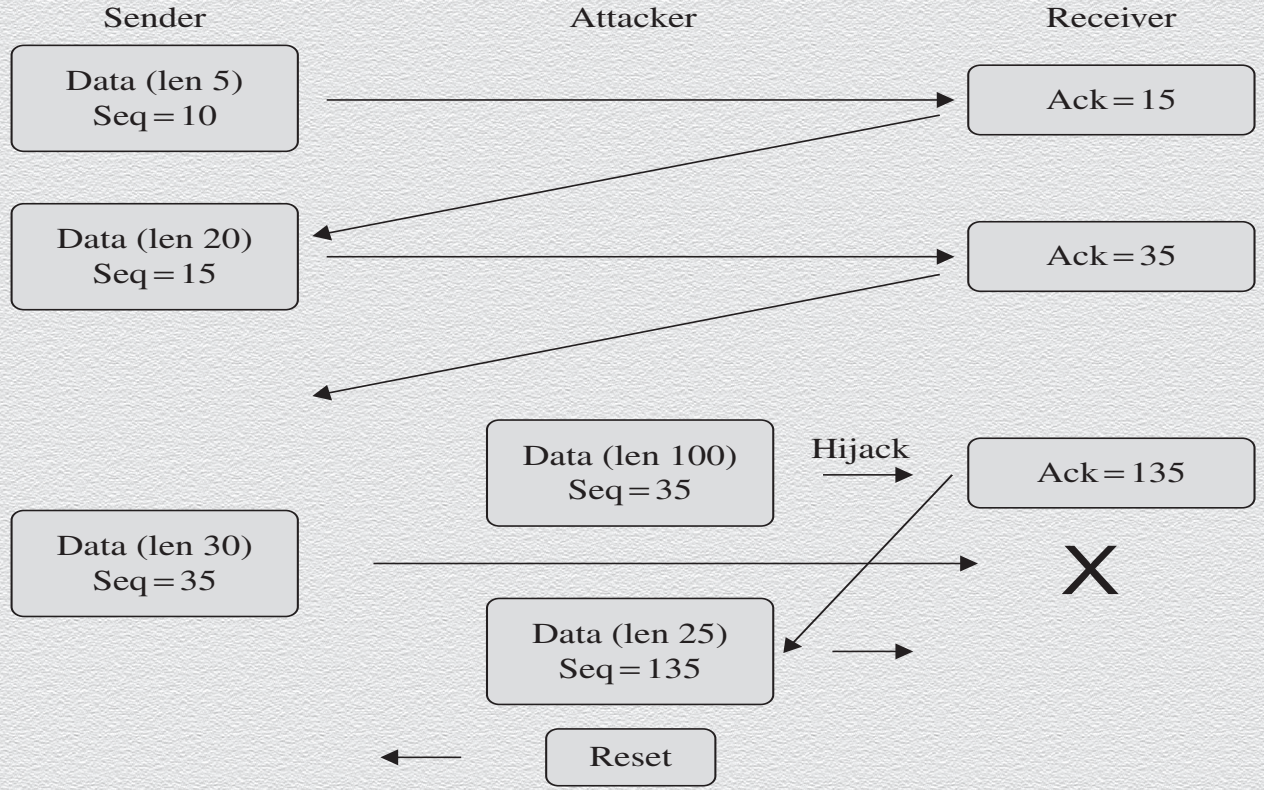
bytes	0	1	2	3
0	Flags		Length	
4	Identification		Flags	Fragment Offset
8	Time to Live	Protocol	Header Checksum	
12	Source IP Address			
16	Destination IP Address			
20	IP Options			Padding
24+	Data ...			



bytes	0	1	2	3
0	Sender Port		Receiver Port	
4	Sequence Number			
8	Acknowledgment Number			
12	Data Offset, Reserved, Flags		Window	
16	Checksum		Urgency	
20	IP Options			Padding
24+	Data ...			

Session hijacking

- an attacker is able to synchronize with a receiver while breaking synchronization with the sender and resetting the sender's connection
- the attacker continues the TCP session while the sender thinks the connection just broke off



Port scanning

```
Nmap scan report
192.168.1.1 / somehost.com (online) ping results
address: 192.168.1.1 (ipv4)
hostnames: somehost.com (user)
The 83 ports scanned but not shown below are in state: closed
Port       State      Service Reason       Product  Version  Extra info
21   tcp   open       ftp     syn-ack      ProFTPD  1.3.1
22   tcp   filtered   ssh     no-response
25   tcp   filtered   smtp    no-response
80   tcp   open       http    syn-ack      Apache   2.2.3    (CentOS)
106  tcp   open       pop3pw  syn-ack      poppassd
110  tcp   open       pop3    syn-ack      Courier pop3d
111  tcp   filtered   rpcbind no-response
113  tcp   filtered   auth    no-response
143  tcp   open        imap    syn-ack      Courier Imapd      released
2004
443  tcp   open       http    syn-ack      Apache  2.2.3      (CentOS)
465  tcp   open       unknown syn-ack
646  tcp   filtered   ldp     no-response
993  tcp   open       imap    syn-ack      Courier Imapd      released
2004
995  tcp   open               syn-ack
2049 tcp   filtered   nfs     no-response
3306 tcp   open       mysql   syn-ack      MySQL   5.0.45
8443 tcp   open       unknown syn-ack
34 sec. scanned
1 host(s) scanned
1 host(s) online
0 host(s) offline
```

Vulnerabilities in wireless networks

- Confidentiality
- Integrity
- Availability
- Unauthorized WiFi access
- WiFi protocol weaknesses
  - picking up the beacon
  - SSID in all frames
  - association issues

Failed Countermeasure: WEP

- Wired Equivalent Privacy, or WEP, was designed at the same time as the original 802.11 WiFi standards as the mechanism for securing those communications
- weaknesses in WEP were first identified in 2001, four years after release
- more weaknesses were discovered over the course of years, until any WEP-encrypted communication could be cracked in a matter of minutes

How WEP Works

- client and access point (AP) have a pre-shared key
- AP sends a random number to the client, which the client then encrypts using the key and returns to the AP
- AP decrypts the number using the key and checks that it's the same number to authenticate the client
- once the client is authenticated, the AP and client communicate using messages encrypted with the key

WEP Weaknesses

Weak encryption key

- WEP allows to be either 64- or 128-bit, but 24 of those bits are reserved for initialization vectors (IV), thus reducing effective key size to 40 or 104 bits
- Keys were either alphanumeric or hex phrases that users typed in and were therefore vulnerable to dictionary attacks

Static key

- Since the key was just a value the user typed in at the client and AP, and since users rarely changed those keys, one key would be used for many months of communications

Weak encryption process

- A 40-bit key can be brute forced easily. Flaws that were eventually discovered in the RC4 encryption algorithm WEP uses made the 104-bit keys easy to crack as well

WEP Weaknesses (cont.)

Weak encryption algorithm

- WEP used RC4 in a strange way (always a bad sign), which resulted in a flaw that allowed attackers to decrypt large portions of any WEP communication

IV collisions

- There were only 16 million possible values of IV, which, in practice, is not that many to cycle through for cracking. Also, they were not as randomly selected as they should have been, with some values being much more common than others

Faulty integrity check

- WEP messages included a checksum to identify transmission errors but did not use one that could address malicious modification

No authentication

- Any client that knows the AP's SSID and MAC address is assumed to be legitimate

WPA (WiFi Protected Access)

Designed in 2003 as a replacement for WEP, quickly followed in 2004 by WPA2

- WPA2 is the standard algorithm today

Non-static encryption key

- WPA uses a hierarchy of keys: New keys are generated for confidentiality and integrity of each session, and the encryption key is automatically changed on each packet
- This way, the keys that are most important are used in very few places and indirect ways, protecting them from disclosure

Authentication

- WPA allows authentication by password, token, or certificate

WPA (cont.)

Strong encryption

- WPA adds support for AES, a much more reliably strong encryption algorithm

Integrity protection

- WPA includes a 64-bit cryptographic integrity check

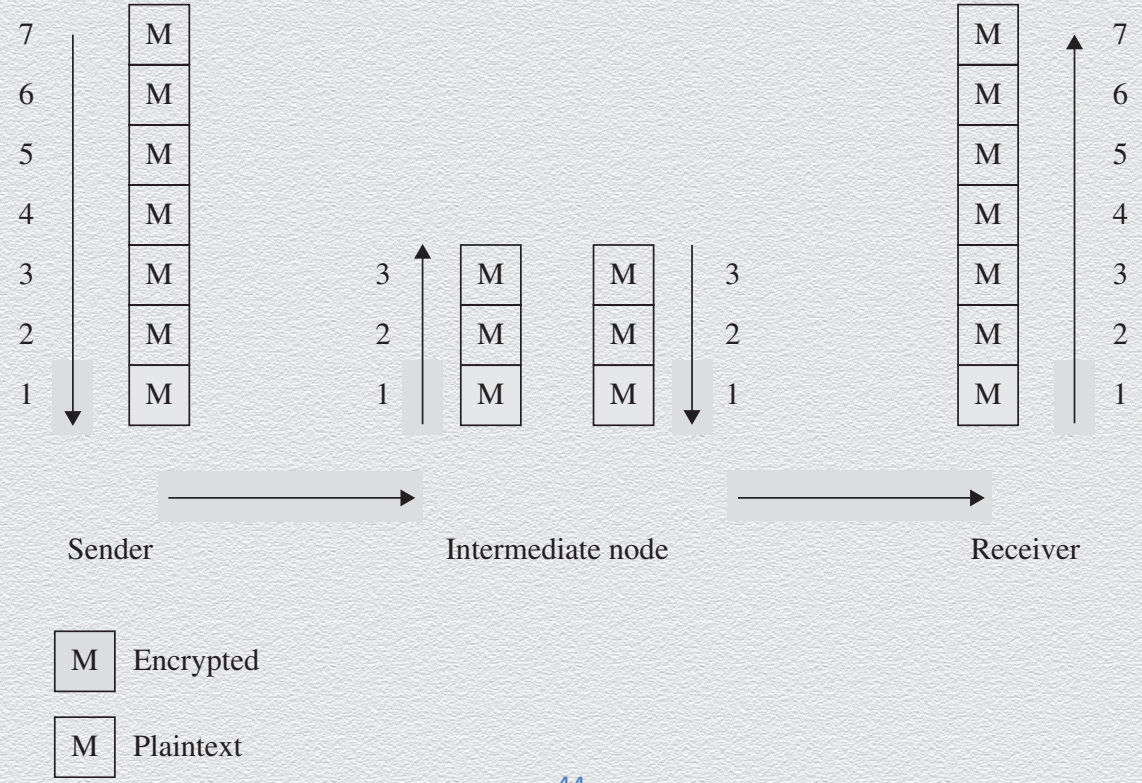
Session initiation

- WPA sessions begin with authentication and a four-way handshake that results in separate keys for encryption and integrity on both ends

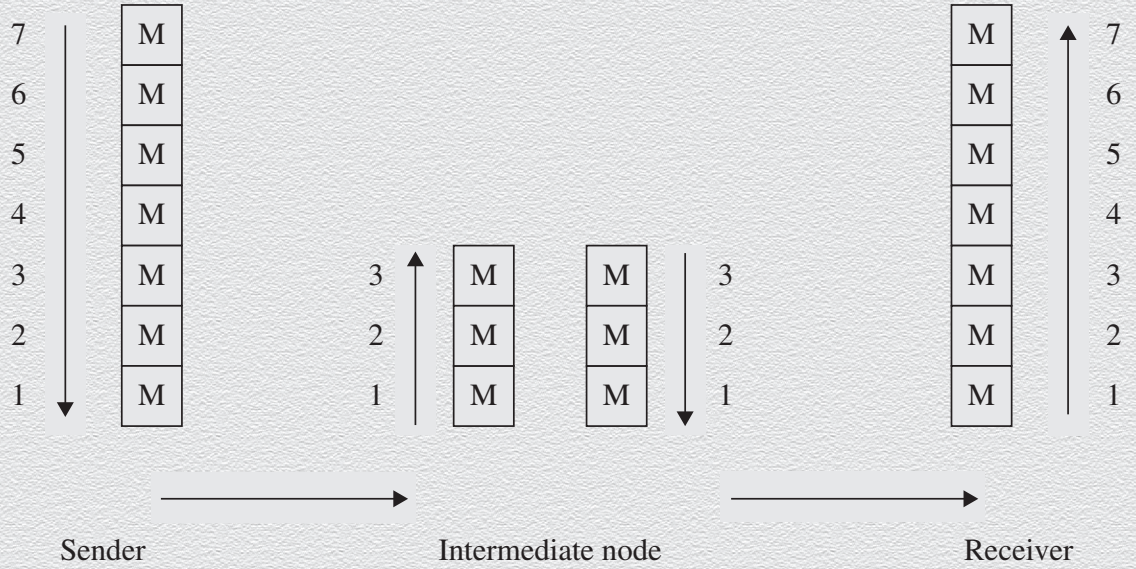
Attacks

- either of very limited effectiveness or require weak passwords

Link encryption



End-to-End encryption



Link vs. End-to-End

Link Encryption	End-to-End Encryption
Security within hosts	
Data partially exposed in sending host	Data protected in sending host
Data partially exposed in intermediate nodes	Data protected through intermediate nodes
Role of user	
Applied by sending host	Applied by user application
Invisible to user	User application encrypts
Host administrators select encryption	User selects algorithm
One facility for all users	Each user selects
Can be done in software or hardware	Usually software implementation; occasionally performed by user add-on hardware
All or no data encrypted	User can selectively encrypt individual data items
Implementation considerations	
Requires one key per pair of hosts	Requires one key per pair of users
Provides node authentication	Provides user authentication

Secure Shell (SSH)

Originally developed for UNIX but now available on most Oss

- provides an authenticated, encrypted path to the OS command line over the network
- replacement for insecure utilities such as Telnet, rlogin, and rsh
- protects against spoofing attacks and modification of data in communication

SSL & TLS

Secure Sockets Layer (SSL): designed in the 1990s

- protects communication between a web browser and server

1999 upgrade to SSL: renamed to Transport Layer Security (TLS)

- while the protocol is still commonly called SSL, TLS is the modern, and much more secure, protocol
- SSL is implemented at OSI layer 4 (transport) and provides
  - server authentication
  - client authentication (optional)
  - encrypted communication

SSL cipher suites

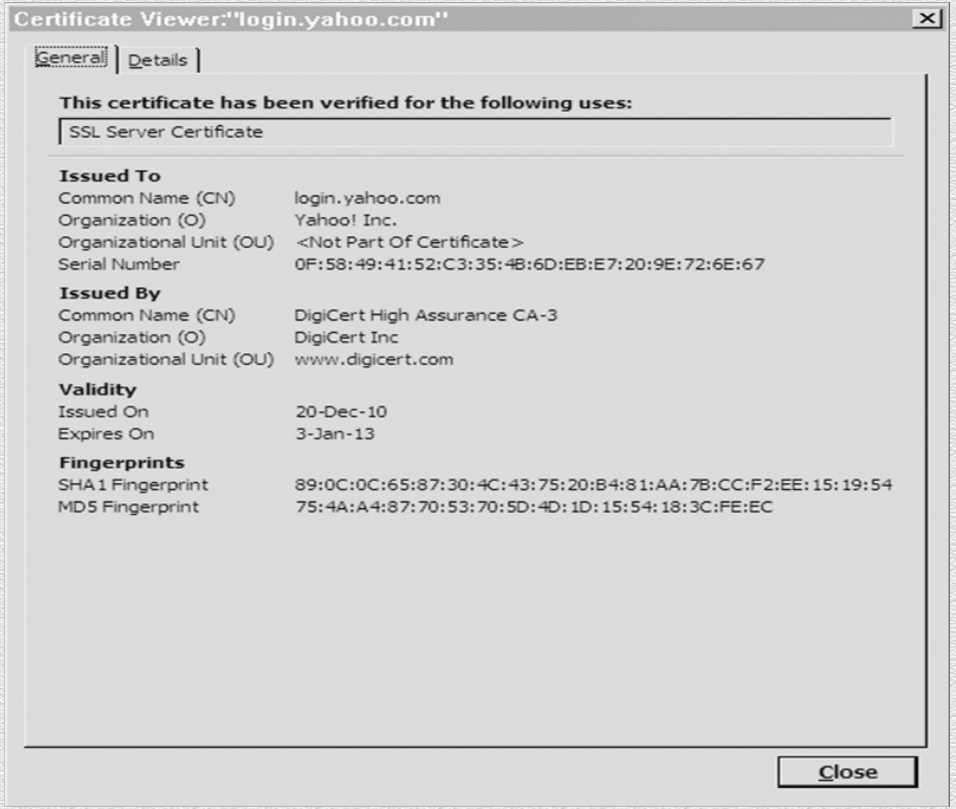
- at the start of an SSL session, the client and server negotiate encryption algorithms
  - known as the “cipher suite”
- the server sends a list of cipher suite options
- the client chooses an option from that list
- the cipher suite consists of
  - a digital signature algorithm for authentication
  - an encryption algorithm for confidentiality
  - a hash algorithm for integrity

SSL cipher suites (partial list)

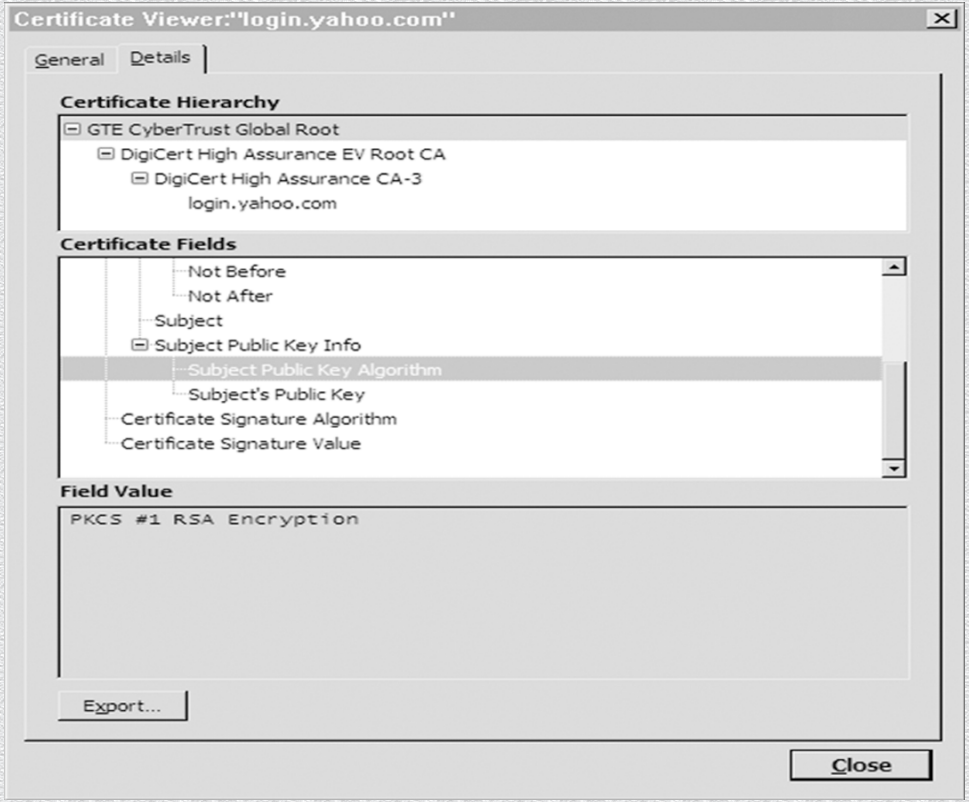
Cipher Suite Identifier	Algorithms Used
TLS_NULL_WITH_NULL_NULL	No authentication, no encryption, no hash function
TLS_RSA_WITH_NULL_MD5	RSA authentication, no encryption, MD5 hash function
TLS_RSA_EXPORT_WITH_RC4_40_MD5	RSA authentication with limited key length, RC4 encryption with a 40-bit key, MD5 hash function
TLS_RSA_WITH_3DES_EDE_CBC_SHA	RSA authentication, triple DES encryption, SHA-1 hash function
TLS_RSA_WITH_AES_128_CBC_SHA	RSA authentication, AES with a 128-bit key encryption, SHA-1 hash function
TLS_RSA_WITH_AES_256_CBC_SHA	RSA authentication, AES with a 256-bit key encryption, SHA-1 hash function
TLS_RSA_WITH_AES_128_CBC_SHA256	RSA authentication, AES with a 128-bit key encryption, SHA-256 hash function
TLS_RSA_WITH_AES_256_CBC_SHA256	RSA authentication, AES with a 256-bit key encryption, SHA-256 hash function
TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA	Diffie–Hellman digital signature standard, triple DES encryption, SHA-1 hash function
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA http://www.iana.org/go/rfc5932	RSA digital signature, Camellia encryption with a 256-bit key, SHA-1 hash function
TLS_ECDHE_ECDSA_WITH_ARIA_256_CBC_SHA384	Elliptic curve cryptosystem digital signature algorithm, Aria encryption with a 256-bit key, SHA-384 hash function

SSL session established

SSL certificate



Chain of certificates

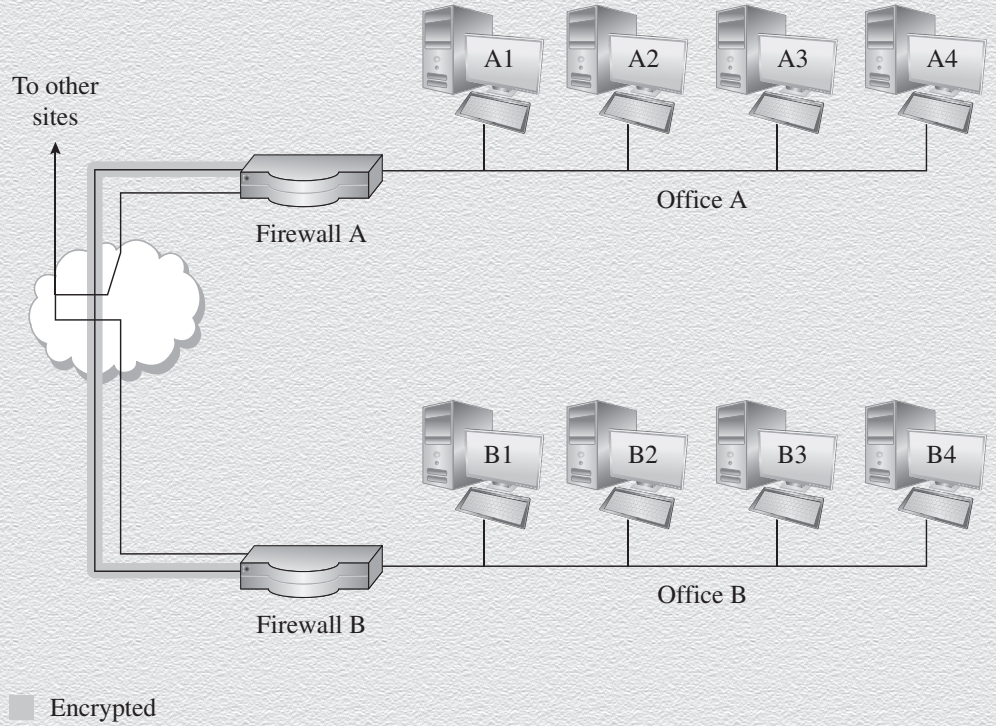


Onion routing

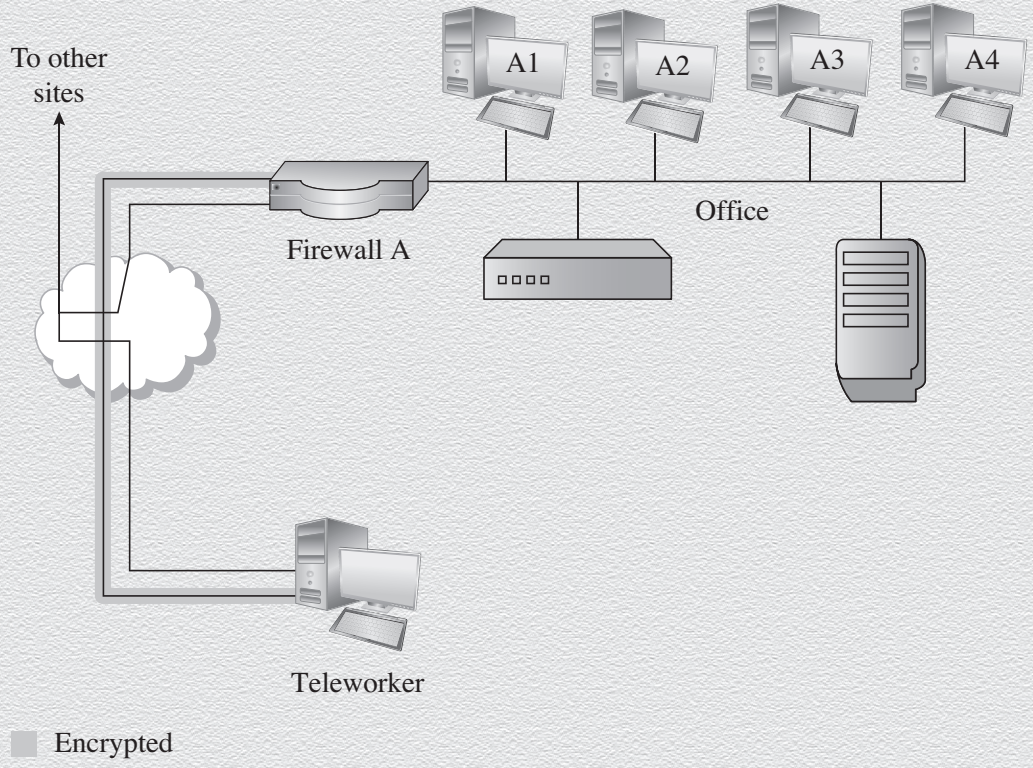
Designed for anonymity

- prevents an eavesdropper from learning source, destination, or content of data in transit in a network
- particularly helpful for evading authorities
  - e.g., users in oppressive countries want to communicate freely with the outside world
- uses asymmetric cryptography, as well as layers of intermediate hosts
  - hosts sending message to ultimate destination cannot determine original sender
  - host receiving message from original sender cannot determine ultimate destination

Virtual Private Networks (VPN)



VPN (cont.)


To other sites
A1
A2
A3
A4
Office
Firewall A
Teleworker
Encrypted

Firewalls

Devices filtering all traffic

- between a protected/“inside” and a less trustworthy/“outside” network
- most firewalls run as dedicated devices
  - easier to design correctly, inspect for bugs and optimize for performance
- implement security policies determining what traffic can or cannot pass through
- an example of a reference monitor, thus it should have three characteristics
  - always invoked (cannot be circumvented)
  - tamperproof
  - small and simple enough for rigorous analysis

Firewall security policy

Rule	Type	Source Address	Destination Address	Destination Port	Action
1	TCP	*	192.168.1.*	25	Permit
2	UDP	*	192.168.1.*	69	Permit
3	TCP	192.168.1.*	*	80	Permit
4	TCP	*	192.168.1.18	80	Permit
5	TCP	*	192.168.1.*	*	Deny
6	UDP	*	192.168.1.*	*	Deny

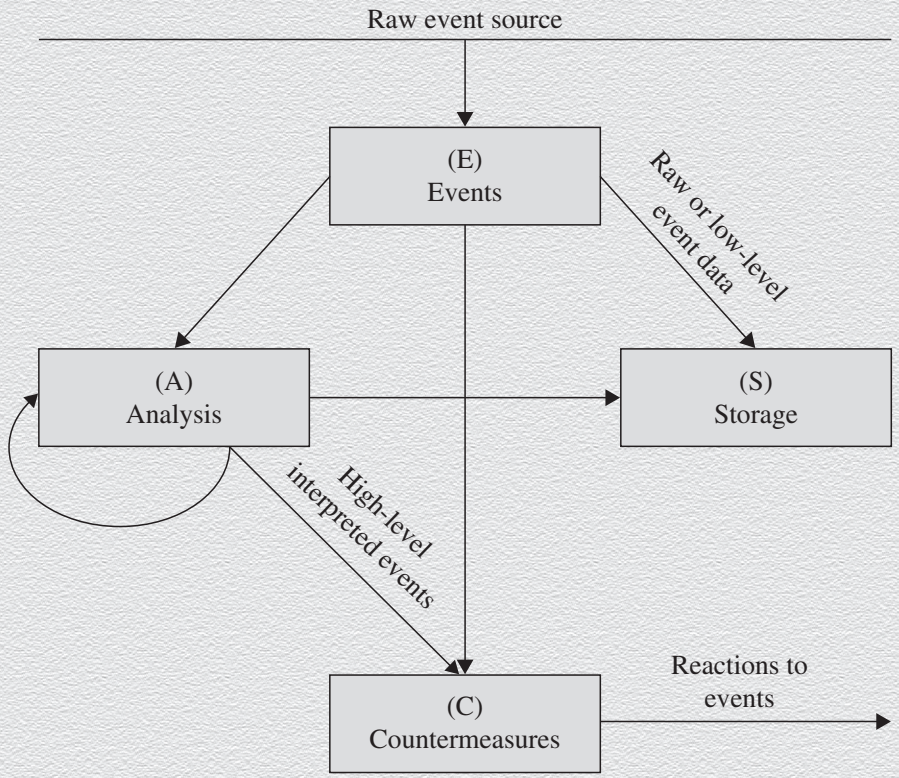
Types of firewalls

- packet filtering gateways or screening routers
- stateful inspection firewalls
- application-level gateways, also known as proxies
- circuit-level gateways
- guards
- personal or host-based firewalls

What firewalls can and cannot do

- can protect an environment only if they control the entire perimeter
- do not protect data outside the perimeter
- are the most visible part of an installation to the outside,
  so they are an attractive target for attack
- must be correctly configured
  - configuration must be updated as the environment changes; firewall activity reports must be reviewed periodically for evidence of attempted/successful intrusion
- exercise only minor control over the content admitted to the inside, meaning that inaccurate or malicious code must be controlled by means inside the perimeter

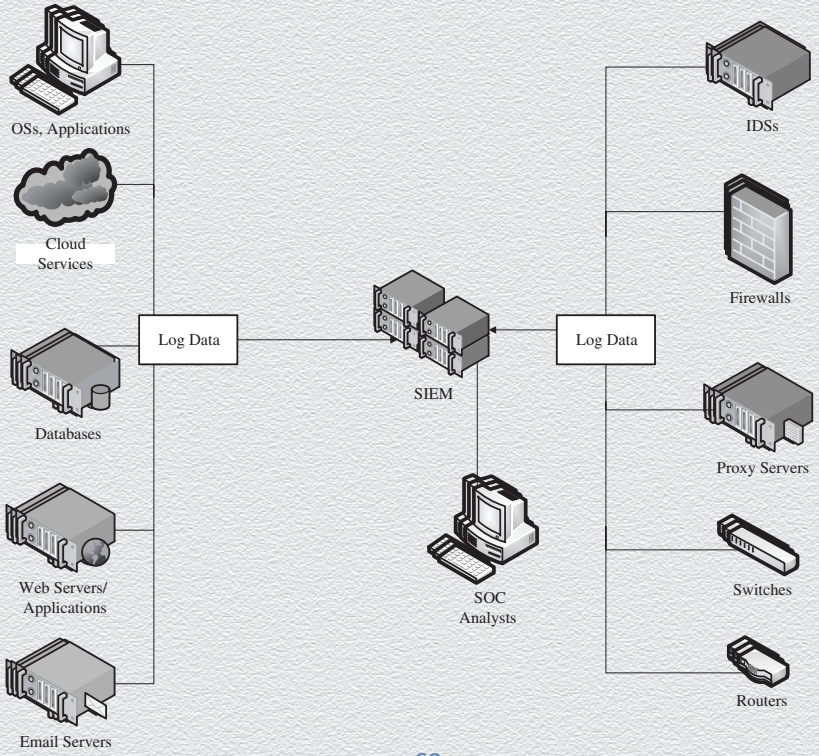
Intrusion detection systems (IDS)



Types of IDS

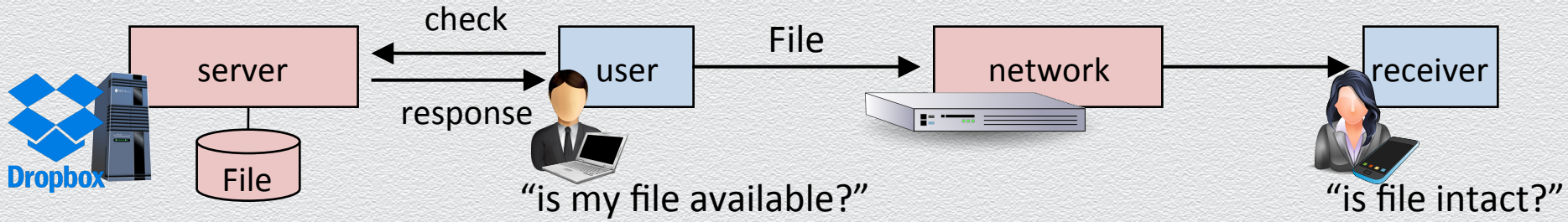
- detection method
  - signature-based, Heuristic
- location
  - front end, Internal
- scope
  - host-based IDS (HIDS), network-based IDS (NIDS)
- capability
  - passive, active (also known as intrusion prevention systems (IPS))

Security Information and Event Management (SIEM)



Authenticated error correcting codes – Falcon codes

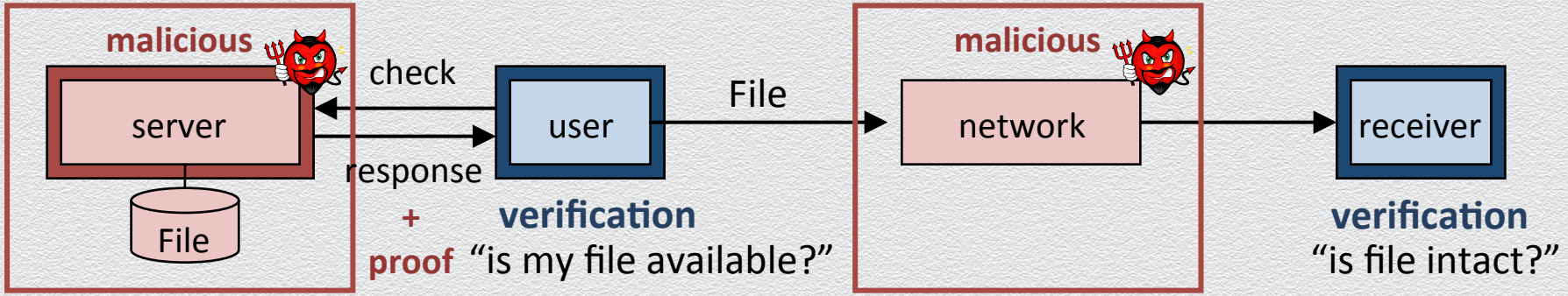
Secure data storage and transmission



Integrity checks offer reliable storage / transmission of files

- data at rest or in transit must be protected against losses or malicious corruptions
  - attacker breaks into a server or router and corrupts an ε-fraction of stream or blocks

Secure data storage and transmission



Integrity checks offer reliable storage / transmission of files

- data at rest or in transit must be protected against losses or malicious corruptions
  - attacker breaks into a server or router and corrupts an ε-fraction of stream or blocks
- error correcting codes (ECCs) provide such guarantees and find many applications
  - fault-tolerance in storage media: e.g., high-end RAM, RAID, CD/DVD, cloud, …
  - reliable transmissions: e.g., digital broadcast, cellular networks, satellite/space, …

Another two distant worlds…

Reed-Solomon (RS)

- block codes (map k symbols to n)
- **less efficient, less practical**
  - $O(n^2)$ encoding/decoding cost
- improvements
  - Tornado codes (x 10^4 faster)
  - O(n logn) cost but practically O(k n)
- **stronger resilience**
  - correction of errors within half the code's minimum distance
  - **info-theoretic guarantees**

Fountain codes

- rateless codes (k input symbols)
- **more efficient, more practical**
  - LT-codes, O(k logk) cost
- improvements
  - Raptor codes, O(k) cost
  - possibly fastest codes available
- **weaker resilience**
  - recovery from **random losses** but no or limited corruption
  - **probabilistic guarantees**

LT [Luby 02] and Raptor [Shokrollahi 06] codes

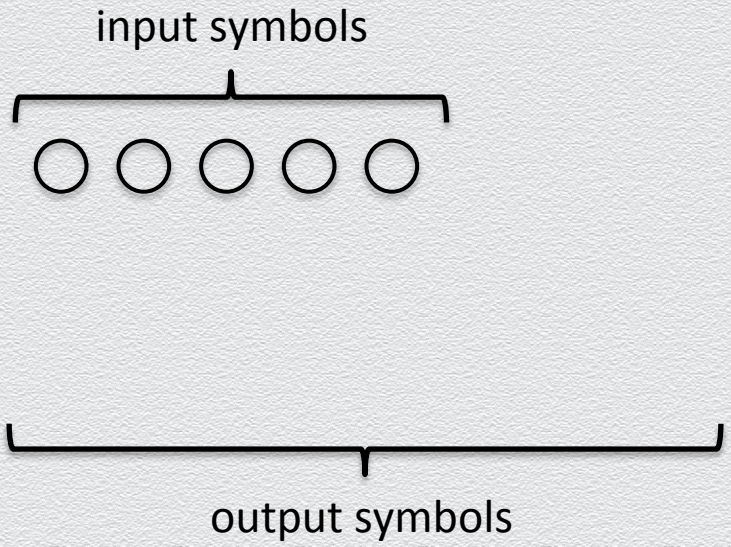
- Erasure codes, tolerate only data loss
- Rateless (can produce an unbounded number of output symbols)
- Quasi-linear & resp. linear encoding/decoding time, really fast in practice too
  - Raptor codes **apply an erasure code (pre-code)** to message symbols
- Used widely
  - RaptorQ are standardized by IETF (RFC 5053, RFC 6330)
  - cellular communications: 3G phones, 3GPP for data sharing over cellular networks
  - digital streaming media: Multimedia Broadcast Multicast Services (MBMS), IP TV/DVB
  - satellite communications: ETSI distribution systems

LT encoding & decoding

Input: k symbols, failure prob. δ, overhead ε, degree distr. D

Output: a sequence of output symbols

- encoding: for any new output symbol
  - select degree d according to D
  - output XOR of d input symbols selected uniformly at random

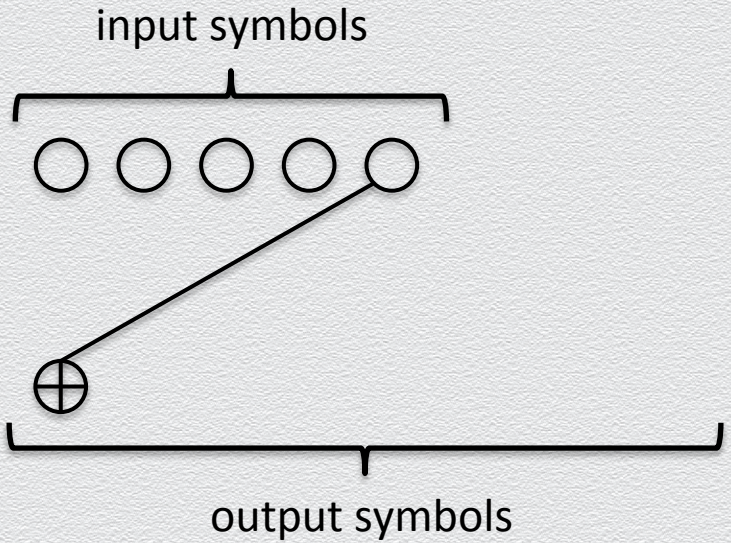


LT encoding & decoding

Input: k symbols, failure prob. δ, overhead ε, degree distr. D

Output: a sequence of output symbols

- encoding: for any new output symbol
  - select degree d according to D
  - output XOR of d input symbols selected uniformly at random

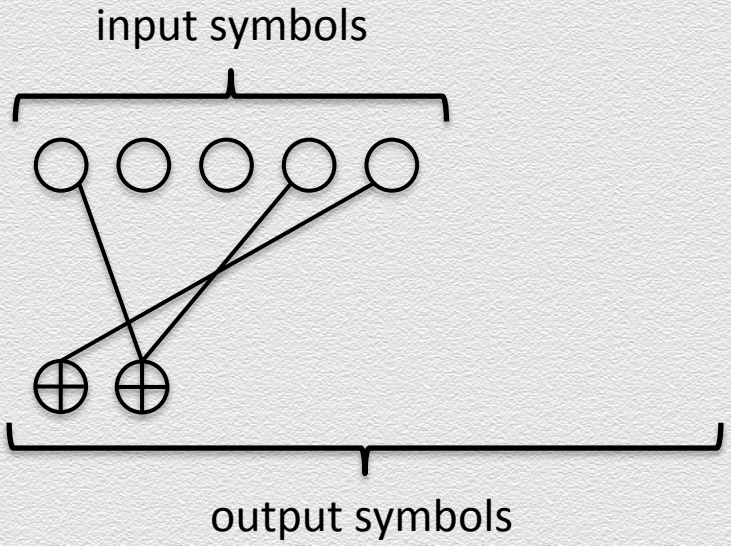


LT encoding & decoding

Input: k symbols, failure prob. δ, overhead ε, degree distr. D

Output: a sequence of output symbols

- encoding: for any new output symbol
  - select degree d according to D
  - output XOR of d input symbols selected uniformly at random

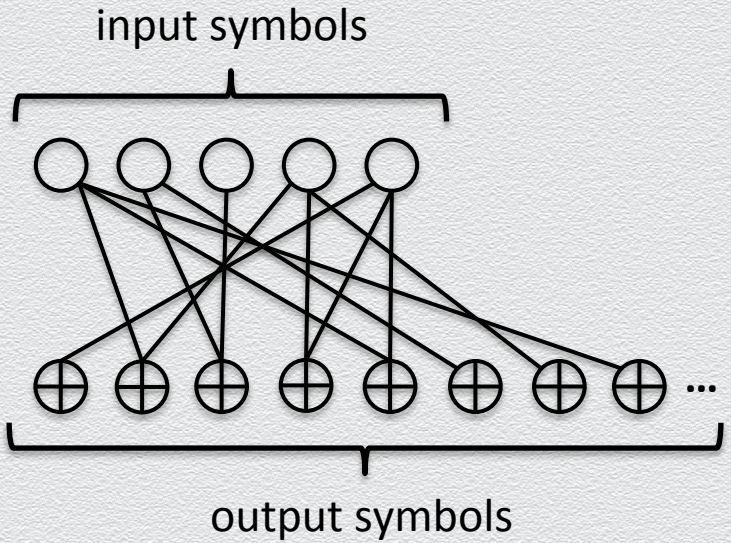


LT encoding & decoding

Input: k symbols, failure prob. δ, overhead ε, degree distr. D

Output: a sequence of output symbols

- encoding: for any new output symbol
  - select degree d according to D
  - output XOR of d input symbols selected uniformly at random

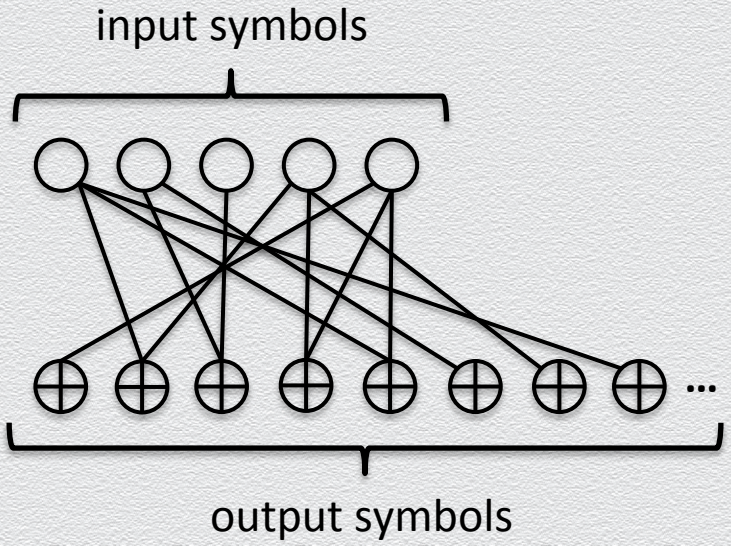


LT encoding & decoding

Input: k symbols, failure prob. δ, overhead ε, degree distr. D

Output: a sequence of output symbols

- encoding: for any new output symbol
  - select degree d according to D
  - output XOR of d input symbols selected uniformly at random
- decoding: belief-propagation algorithm

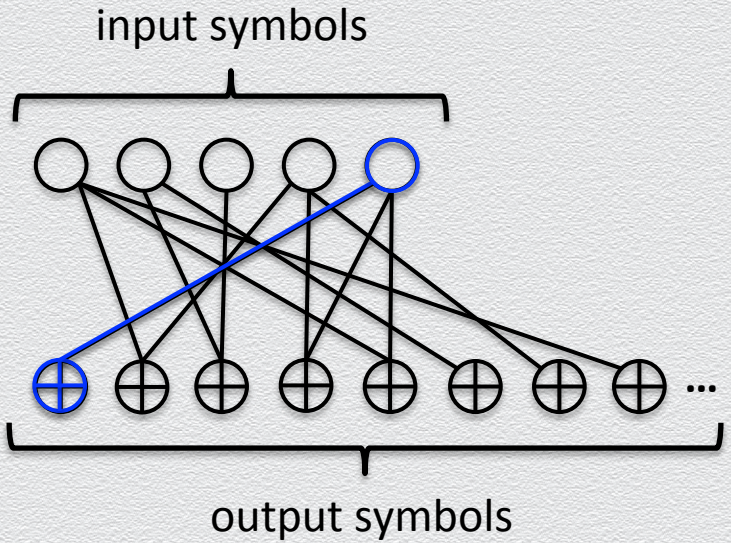


LT encoding & decoding

Input: k symbols, failure prob. δ, overhead ε, degree distr. D

Output: a sequence of output symbols

- encoding: for any new output symbol
  - select degree d according to D
  - output XOR of d input symbols selected uniformly at random
- decoding: belief-propagation algorithm

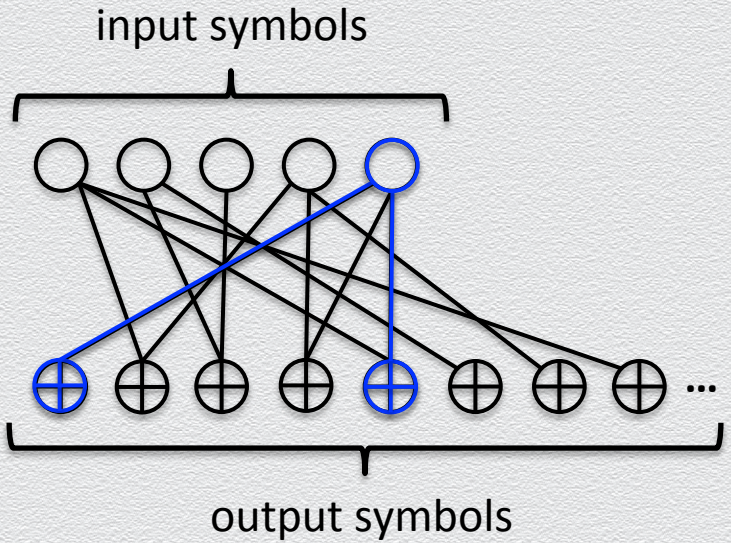


LT encoding & decoding

Input: k symbols, failure prob. δ, overhead ε, degree distr. D

Output: a sequence of output symbols

- encoding: for any new output symbol
  - select degree d according to D
  - output XOR of d input symbols selected uniformly at random
- decoding: belief-propagation algorithm

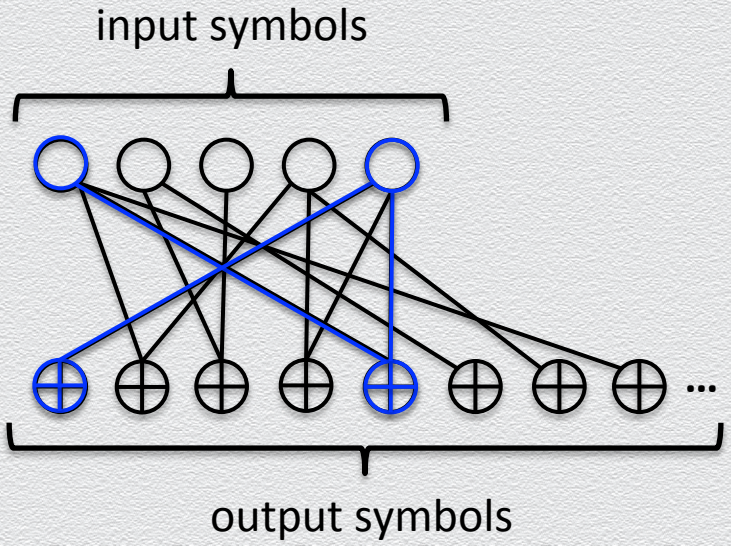


LT encoding & decoding

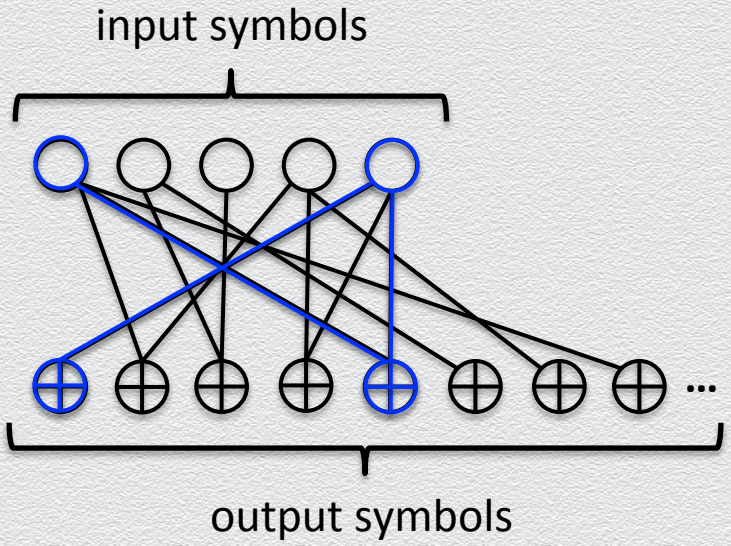
Input: k symbols, failure prob. δ, overhead ε, degree distr. D

Output: a sequence of output symbols

- encoding: for any new output symbol
  - select degree d according to D
  - output XOR of d input symbols selected uniformly at random
- decoding: belief-propagation algorithm



LT encoding & decoding



Input: k symbols, failure prob. δ, overhead ε, degree distr. D

Output: a sequence of output symbols

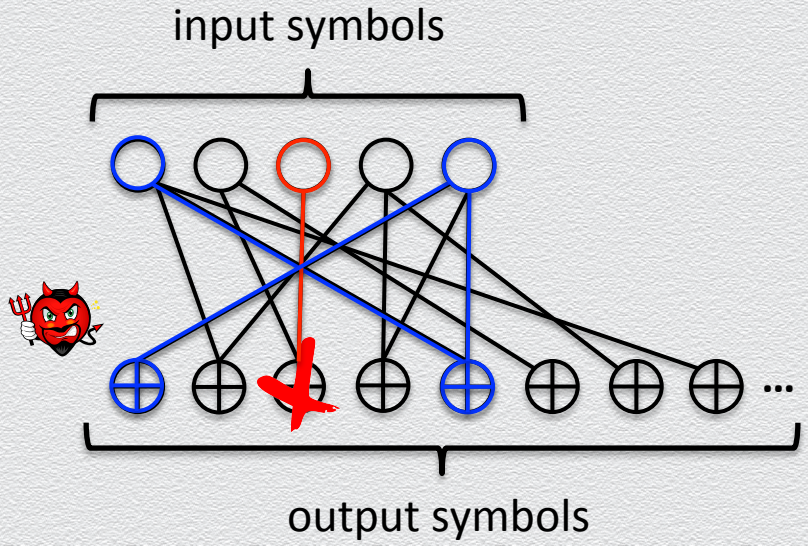
- encoding: for any new output symbol
  - select degree d according to D
  - output XOR of d input symbols selected uniformly at random
- decoding: belief-propagation algorithm
- distribution D is set so that any (1+ε)k or more output symbols, **filtered by a random erasure channel**, can be successfully decoded with prob. at least 1-δ
  - decoding guarantees abruptly vanish with adversarial errors!
    - e.g., targeted erasures

LT encoding & decoding

Input: k symbols, failure prob. δ, overhead ε, degree distr. D

Output: a sequence of output symbols

- encoding: for any new output symbol
  - select degree d according to D
  - output XOR of d input symbols selected uniformly at random
- decoding: belief-propagation algorithm
- distribution D is set so that any (1+ε)k or more output symbols, **filtered by a random erasure channel**, can be successfully decoded with prob. at least 1-δ
  - decoding guarantees abruptly vanish with adversarial errors!
    - e.g., targeted erasures



Raptor decoding is easy to disrupt!

Adversarial errors

- RFCs 5053 & 6330 suggest CRC checksums to detect corruption – easy to circumvent

Targeted erasures [Lopes, Neves 14]

- RFCs 5053 & 6330 employ weak pseudorandom generators – easy to predict
- TFTP app (UDP based) running on Raptor-encoded data, sent through a compromised router
- attack is based on brute-force pre-computation of the most harmful erasure patterns
  - almost always makes the message irrecoverable!
  - runs at line-speed and is very hard to detect
    - data independent selected erasures of a few code symbols (in most configurations)
  - disrupts file transfers even if encoding is authenticated / encrypted – TFTP over IPsec/ESP)
    - timing-based estimation of symbol ordering

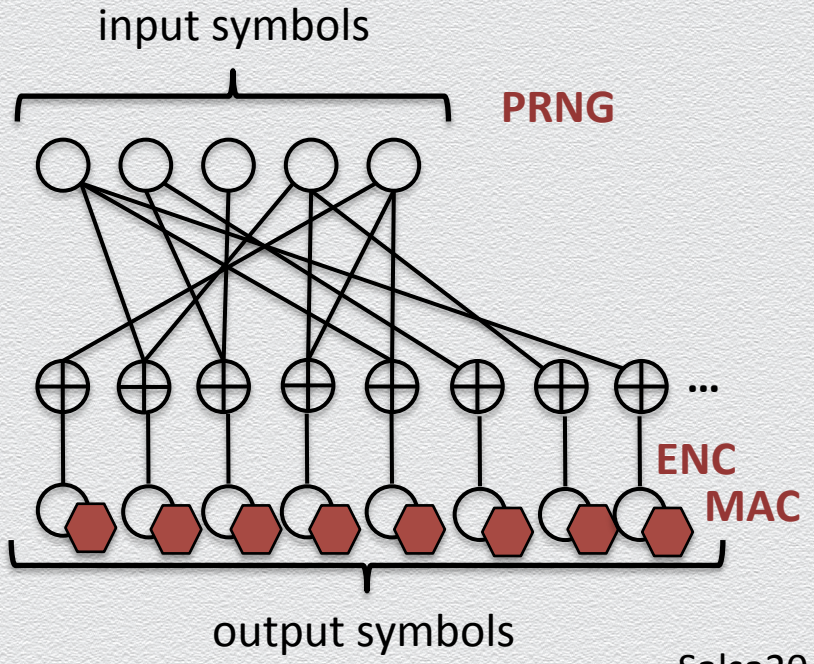
Falcon codes

[Juels et al., CCS15]

- class of authenticated error-correcting codes
  - based on LT-codes, extend to Raptor codes
  - crypto-enhanced encoding using a secret key
- **tolerate adversarial errors** & **maintain linear complexity**
  - scalable block and rateless code extensions, experimental evaluation, **practical**
  - security analysis in new adversarial model for Fountain codes, **provably secure**
- new integrity-protection secure coding primitive

brief overview of main scheme

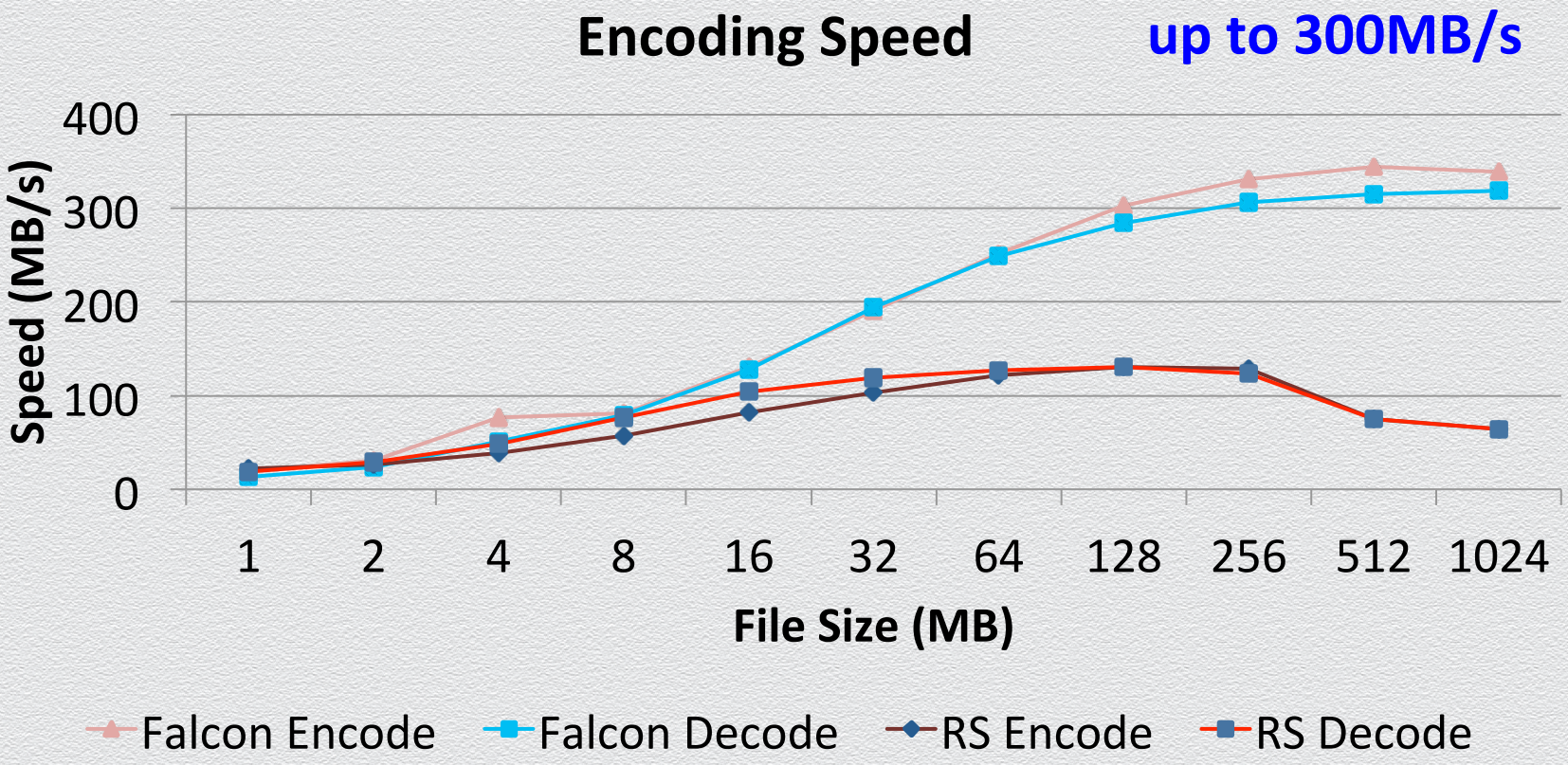
Core Falcon code



- Apply LT-encoding with strong crypto PRNG
  - seed PRNG with secret key and nonce
  - **graph structure is unpredictable**
- Encrypt & MAC each output symbol
  - **symbols don't leak graph-structure**
  - **errors are reduced to erasures**
  - one or both may be avoided if provided by data-transmission channel (e.g., ENC when tunneling over SSH, ENC and MAC in IPsec)
  - batching MACs can be useful
    - e.g., >20% speed-up: 4-byte symbols, batch size >50 symbols

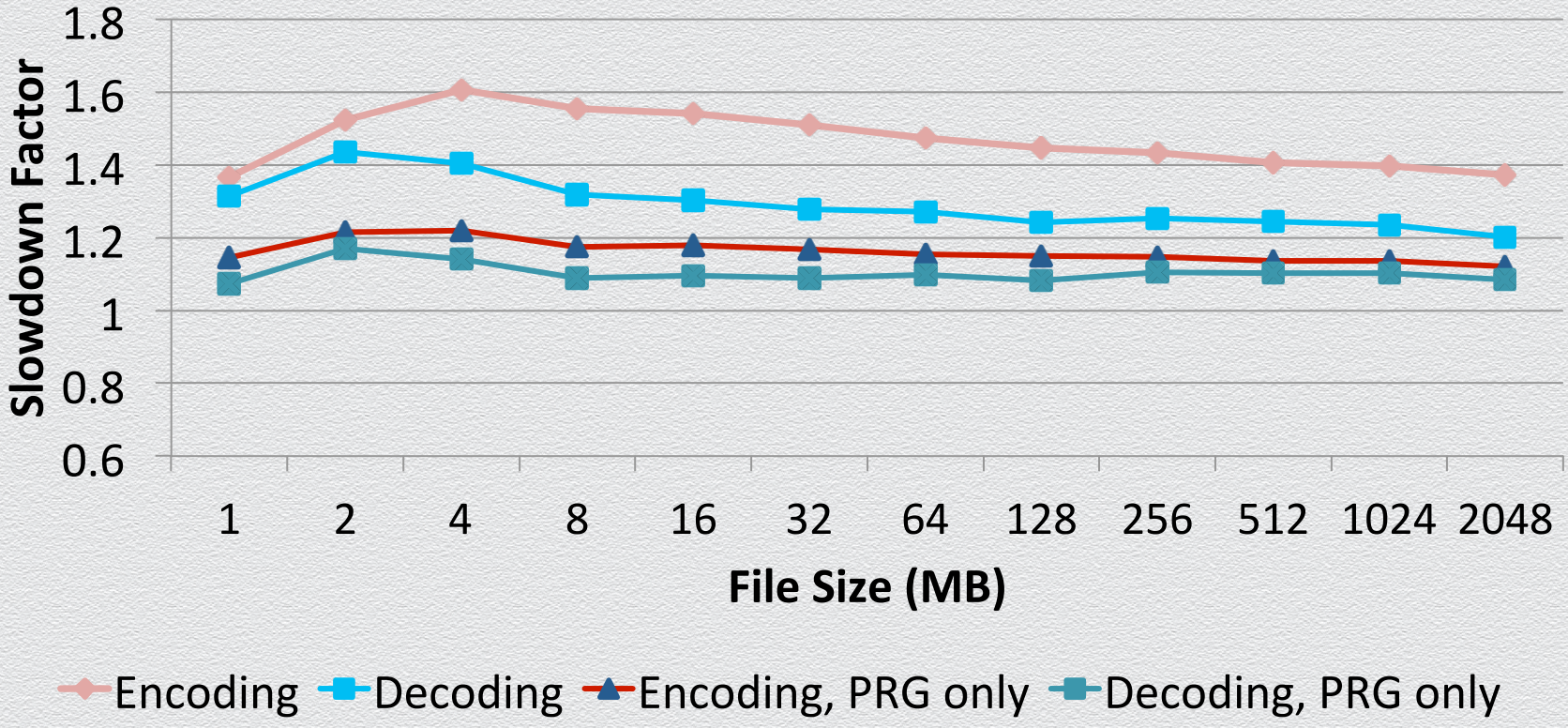
Salsa20
AES-GCM

Falcon Vs. (optimized) Reed-Solomon



Falcon Vs. (insecure) Raptor

Ratio Overhead for Falcon **close to 25%, < 60%**



Applications

Removing Raptors' vulnerabilities

- Falcon codes are **provably secure** and **efficient safeguard** against the successful attack on RFC 6330 by [Lopes, Neves, 14]

Coding throughput (in MB/s)

File size	Encode	Decode	Decode pre-comp.	FFT `Mix`
512KB	13.89	8.61	14.12	277.6
1MB	26.74	16.58	26.25	291.37
4MB	63.69	51.22	72.86	285.40
16MB	157.48	147.19	179.17	291.75
64MB	296.16	271.30	294.52	287.25
256MB	378.42	343.39	351.79	287.45
512MB	398.32	358.14	362.63	286.95

Faster secure cloud storage

- Protocols for efficiently checking that cloud files remain intact and retrievable (e.g., "proofs of retrievability" or POR protocols) require coding schemes that are secure against adversarial corruptions, which are typically also **slow**
- Falcon codes can provide significant **efficiency gainings** as drop-in replacements of Reed-Solomon codes (or other custom-made attack-tolerant codes) used today
- E.g., **39% speed-up** over the FFT-type code used in the state-of-the-art POR [Shi et al. 03]