IPv6 Security
APNIC 26 – Christchurch, New Zealand
August 2008

Merike Kaeo
merike@doubleshotsecurity.com

Agenda

- Introduction To Security Issues
- Operational Best Practices
- Filtering and Firewalls
- Crypto Fundamentals
- IPsec Technology in Depth

Intro To Security Issues

Basic Terms

- Threat
  - Any circumstance or event with the potential to cause harm to a networked system
    - Denial of Service / Unauthorized Access / Impersonation / Worms / Viruses
- Vulnerability
  - A weakness in security procedures, network design, or implementation that can be exploited to violate a corporate security policy
    - software bugs / configuration mistakes / network design flaw
- Risk
  - The possibility that a particular vulnerability will be exploited
    - *Risk analysis:* The process of identifying security risks, determining their impact, and identifying areas requiring protection

What Can Intruders Do?

- Eavesdrop - compromise routers, links, or DNS
- Send arbitrary messages (spoof IP headers and options)
- Replay recorded messages
- Modify messages in transit
- Write malicious code and trick people into running it
- Exploit bugs in software to ‘take over’ machines and use them as a base for future attacks

What Are Security Goals?

- Controlling Data Access
- Controlling Network Access
- Protecting Information in Transit
- Ensuring Network Availability
- Preventing Intrusions
- Responding To Incidences

Security Properties

- Confidentiality
  - Access to information is restricted to those who are privileged to see it
- Integrity
  - Having trust that information has not been altered during its transit from sender to intended recipient
- Accountability
  - Non-repudiation: property of a cryptographic system that prevents a sender from denying later that he or she sent a message or performed a certain action
- Availability
  - Information or resources are accessible when required

Security Services

- Authentication
  - Process of verifying the claimed identity of a device, user and/or application
- Authorization
  - Rights and permissions granted to a user, device or application that enables access to resources
- Access Control
  - Means by which authorized user has access to resources
- Encryption
  - Mechanism by which information is kept confidential
- Auditing
  - Process that keeps track of networked activity

Fundamental Issues

- What is meant by *Securing The Network* ?
- Design security into IPv6 networks that do not blindly mimic the current IPv4 architectures
  - Don't break working v4 infrastructure
  - Don't re-architect current problems and place limitations on IPv6 capabilities
- Requires some thought to policy
  - Where are you vulnerable today ?
  - What new application capabilities are possible with IPv6?
  - New risk assessment will help (re)define appropriate security policy
- Security policy will dictate which security measures to implement

Why Worry About Security?

- How much you worry depends on risk assessment analysis
  - *Risk analysis*: the process of identifying security risks, determining their impact, and identifying areas requiring protection
- Must compare need to protect asset with implementation costs
- Define an effective security policy with incident handling procedures

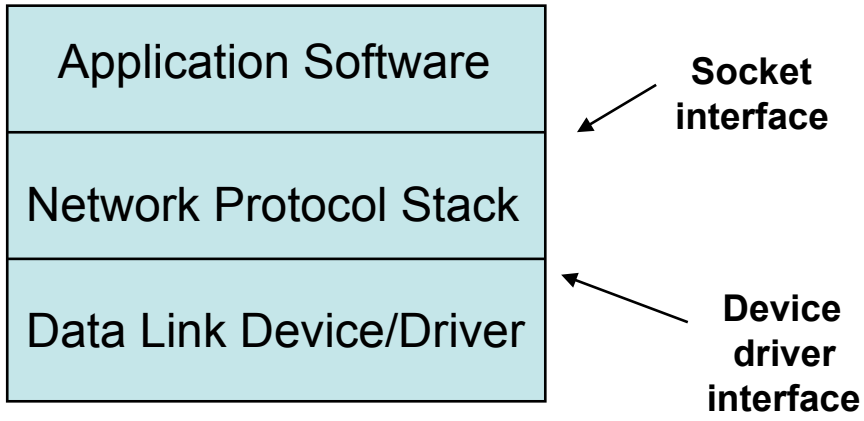
First Step…..Security Policy

- Design Policy
  - Study and analyze your network environment
  - Develop a threat model
  - Perform a security vulnerability assessment
- Implement Policy
  - Use appropriate technology
  - Train all employees
- Enforce Policy
  - Automate and audit

Characteristics of a Good Policy

- Can it be implemented technically?
- Are you able to implement it organizationally?
- Can you enforce it with security tools and/or sanctions?
- Does it clearly define areas of responsibility for the users, administrators, and management?
- Is it flexible and adaptable to changing environments?

Security
host / network / application



Need to implement security solutions at all layers in a reasonable fashion.

So….Big Question: What Is Reasonable?

Degrees of Security

Will I Go Bankrupt ?

- Spend More Money
- Spend More Time

Is It An Embarrassment ?

NEED TO DO A RISK ANALYSIS !

DOUBLE SHOT SECURITY

Risk Analysis & Assessment

- Identify Critical Assets
  - Hardware, software, data, people, documentation
- Place a Value on the Asset
  - Intangible asset – importance or criticality
  - Tangible asset – replacement value and/or training costs
- Determine Likelihood of Security Breaches
  - What are threats and vulnerabilities ?

Risk Mitigation vs Cost

Risk mitigation: the process of selecting appropriate controls to reduce risk to an acceptable level.

The ***level of acceptable risk*** is determined by comparing the risk of security hole exposure to the cost of implementing and enforcing the security policy.

Assess the cost of certain losses and do not spend more to protect something than it is actually worth.

Traditional IT Security Policies

- Physical security controls
  - Media
  - Equipment location
  - Environmental safeguards
- Logical security controls
  - Subnet boundaries
  - Routing boundaries
  - Logical access control
- System and data integrity
  - Firewalls
  - Network services
- Data confidentiality
- Verify / Monitor / Audit
  - Accounting
  - Management
  - Intrusion detection

Added Policy Considerations

- Policies and procedures for staff
  - Secure backups
  - Equipment certification
  - Use of Portable Tools
  - Audit Trails
  - Incident Handling
- Security awareness training for users of the network
  - Critical for airline personnel
  - Added challenge of non-network savvy maintenance personnel

Incident Handling

- You will have to deal with a security incident
- DON'T PANIC!! :)
- Systematically assess vulnerabilities and where to possibly place more effort on auditing / monitoring
- Detect / Assess / Respond
  - Automate as much as possible
  - Requires detailed operational guidance

Sample Policy Modules

- Acceptable Use Policy
- Application Service Provider Policy
- Audit Vulnerability Scanning Policy
- Dial-In Access Policy
- Password Protection Policy
- Remote Access Policy
- Router Security Policy
- Server Security Policy

Security Policy Summary

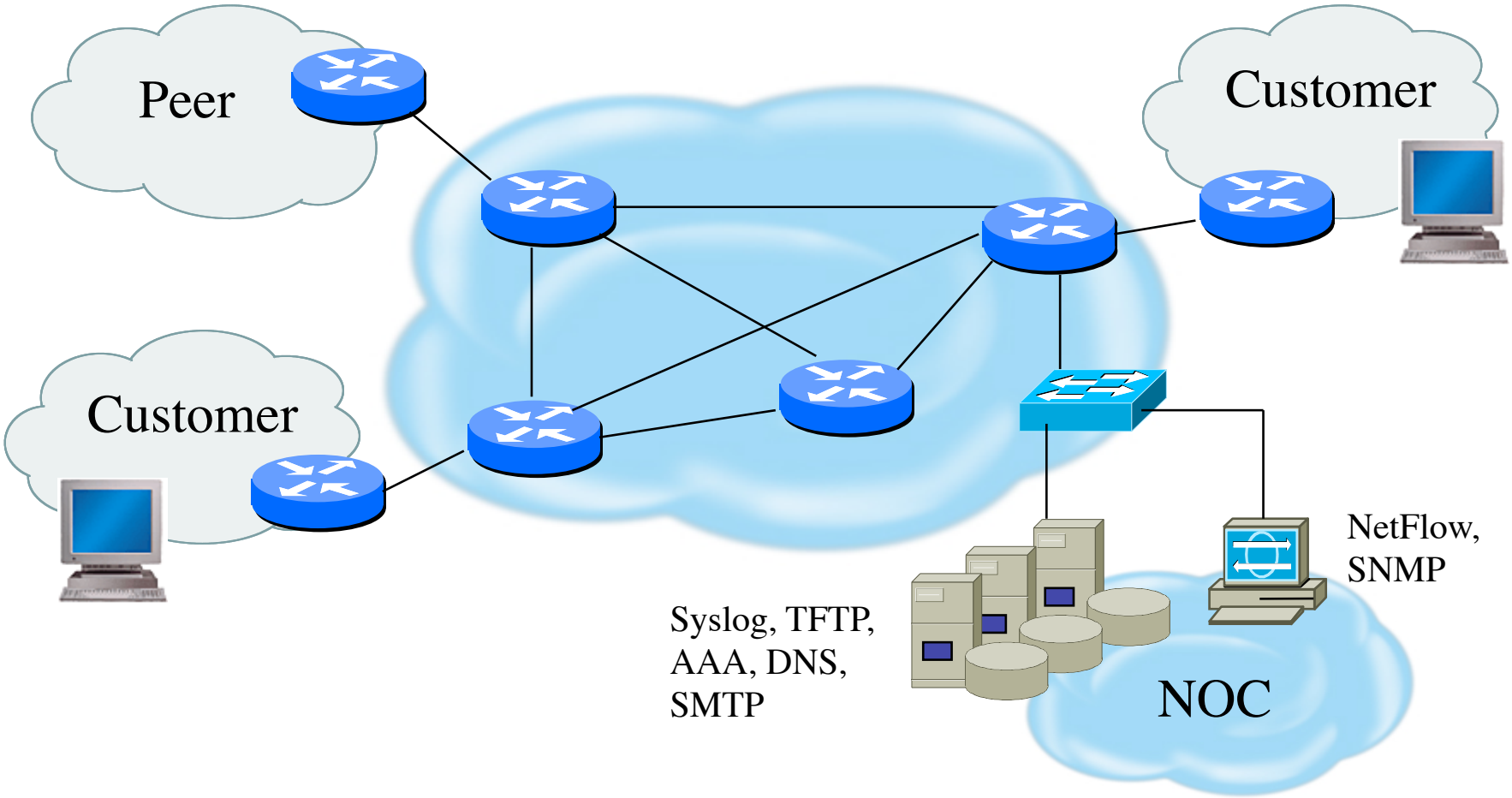
- Understand your environment
  - Know your assets
  - Know points of vulnerability
- Compare costs vs risks
- Limit scope of access
- Be reasonable

Operational Best Practices

Basic Security Problem

- Unwanted Traffic
  - SPAM
  - Phishing
  - DDoS (BOTnets, any large traffic flow)
- Does Bandwidth Matter?
- Does Location Matter?
- What Can We Do?

Infrastructure Security



How Are We Protected?

- Understand the Problem
- Establish an Effective Security Policy
  - physical security
  - logical security
  - control/management plane
  - routing plane
  - data plane
- Have Procedures In Place For Incident Response
  - assessing software vulnerability risk
  - auditing configuration modifications

Security Services

- User Authentication
- User Authorization
- Data Origin Authentication
- Access Control
- Data Integrity
- Data Confidentiality
- Auditing / Logging
- DoS Mitigation

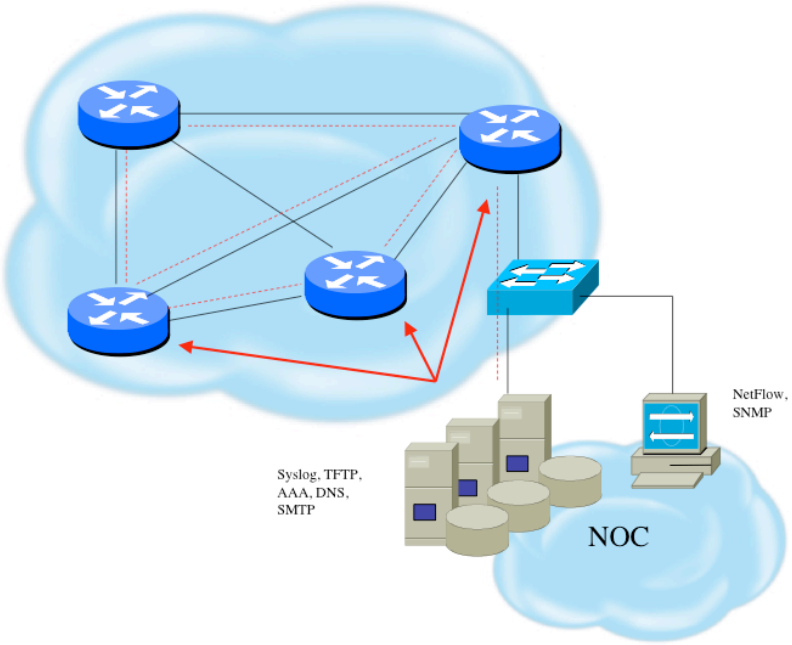
Functional Security Considerations

- Device Physical Access
- Device Management
  - In-band
  - Out-Of-Band (OOB)
- Data Path
- Routing Control Plane
- Software Upgrade / Configuration Integrity
- Logging
- Filtering
- DoS Tracking /Tracing
  - Sink Hole Routing
  - Black-Hole Triggered Routing
  - Unicast Reverse Path Forwarding (uRPF)
  - Rate Limiting

Device Physical Access

- Equipment kept in highly restrictive environments
- Console access
  - password protected
  - access via OOB management
- Individual users authenticated
- Social engineering training and awareness

Device In-Band Management



SSH primarily used; Telnet only from jumphosts

All access authenticated

- Varying password mechanisms
- AAA usually used
- Single local database entry for backup

Each individual has specific authorization

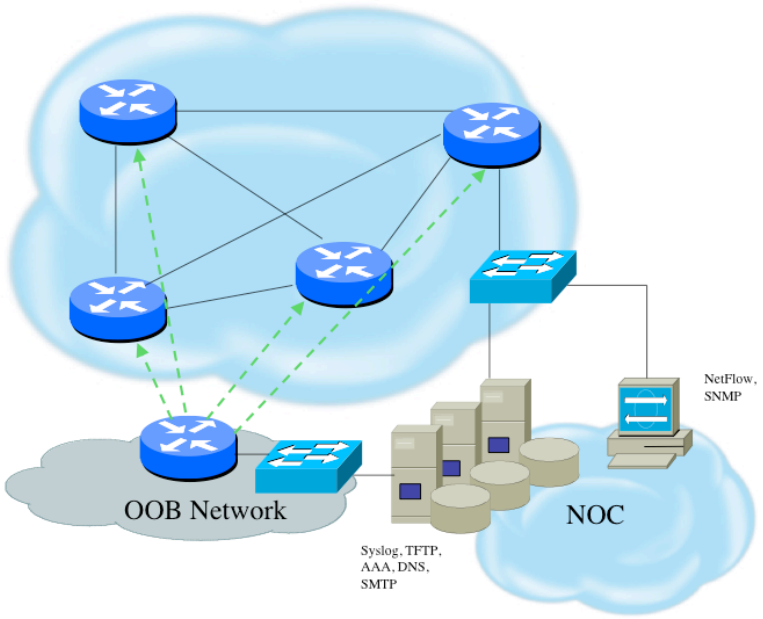
Strict access control via filtering

Access is audited with triggered pager /email notifications

SNMP is read-only

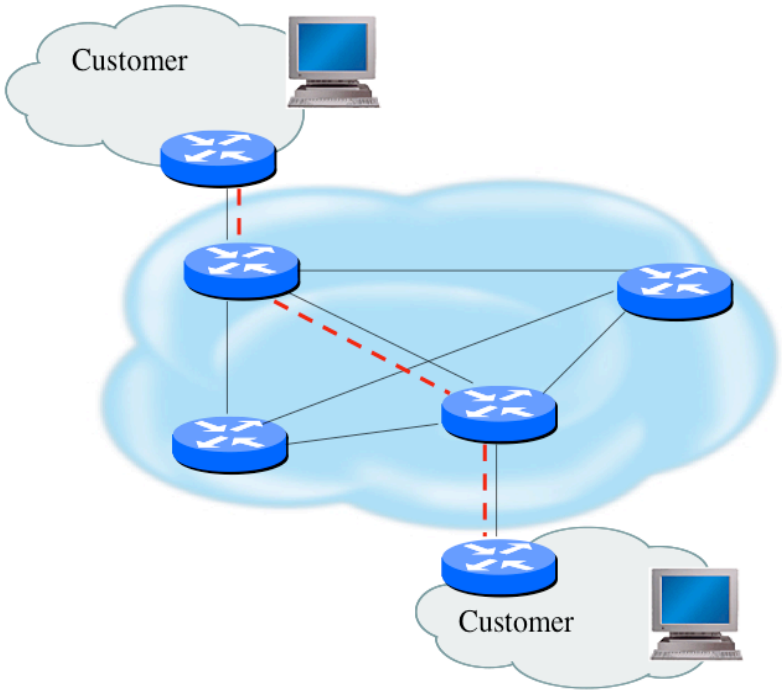
- Community strings updated every 30-90 days

Device OOB Management



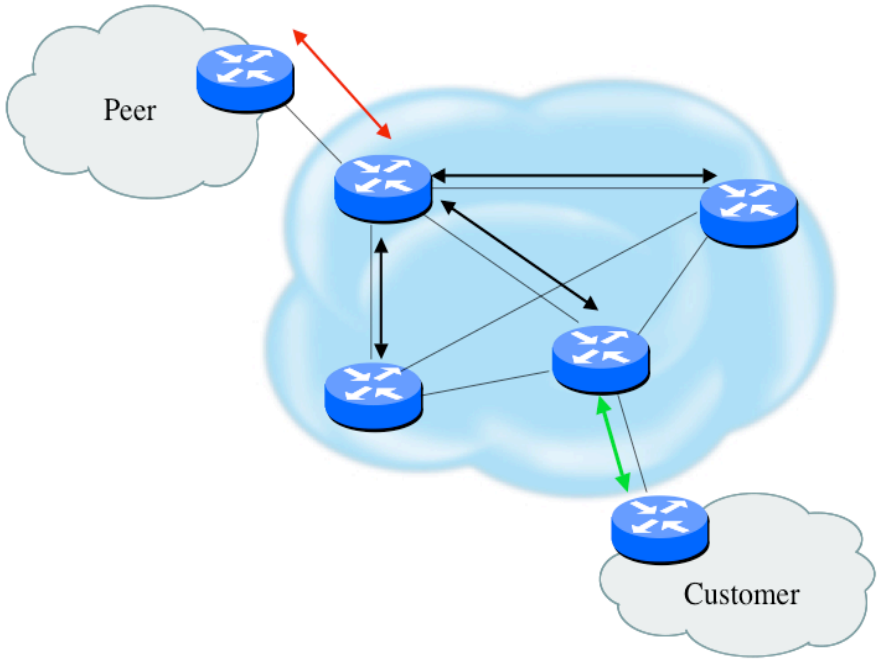
- SSH primarily used; Telnet only from jumphosts
- All access authenticated
  - Varying password mechanisms
  - AAA usually used (server typically different for in-band vs OOB)
  - Single local database entry for backup
- Each individual has specific authorization
- Strict access control via filtering
- Access is audited with triggered pager/email notifications
- SNMP is read-only
  - community strings updated every 30-90 days

Data Path



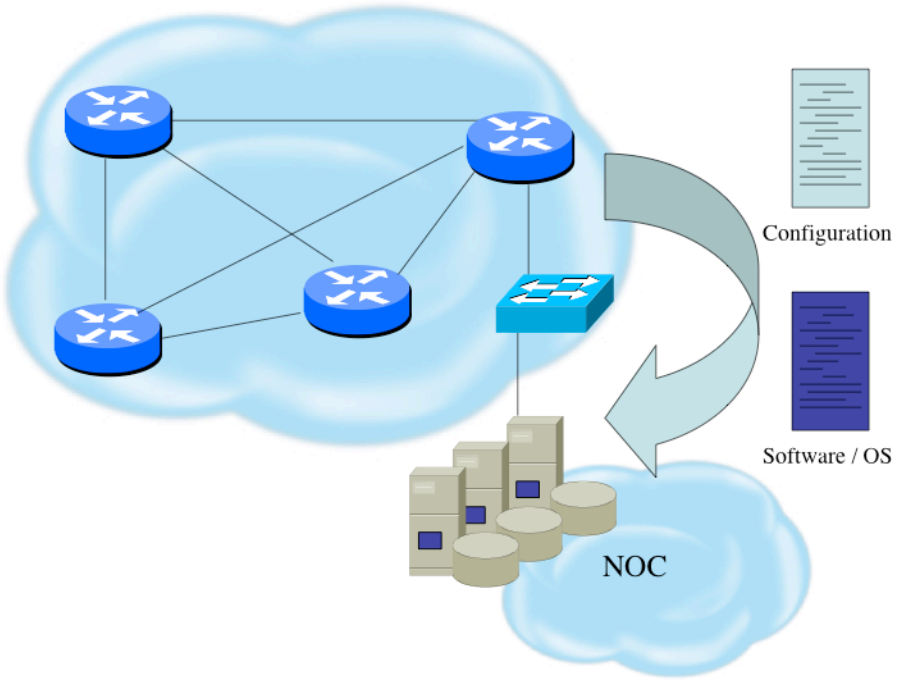
- Filtering and rate limiting are primary mitigation techniques
- BCP-38 guidelines for ingress filtering
- Null-route and black-hole any detected malicious traffic
- Netflow is primary method used for tracking traffic flows
- Unicast Reverse Path Forwarding is not consistently implemented
- Logging of Exceptions

Routing Control Plane



- MD-5 authentication
  - Some deploy at customer's request
- Route filters limit what routes are believed from a valid peer
- Packet filters limit which systems can appear as a valid peer
- Limiting propagation of invalid routing information
  - Prefix filters
  - AS-PATH filters (trend is leaning towards this)
  - Route dampening (latest consensus is that it causes more harm than good)
- Not yet possible to validate whether legitimate peer has authority to send routing update

Software Upgrade / Integrity

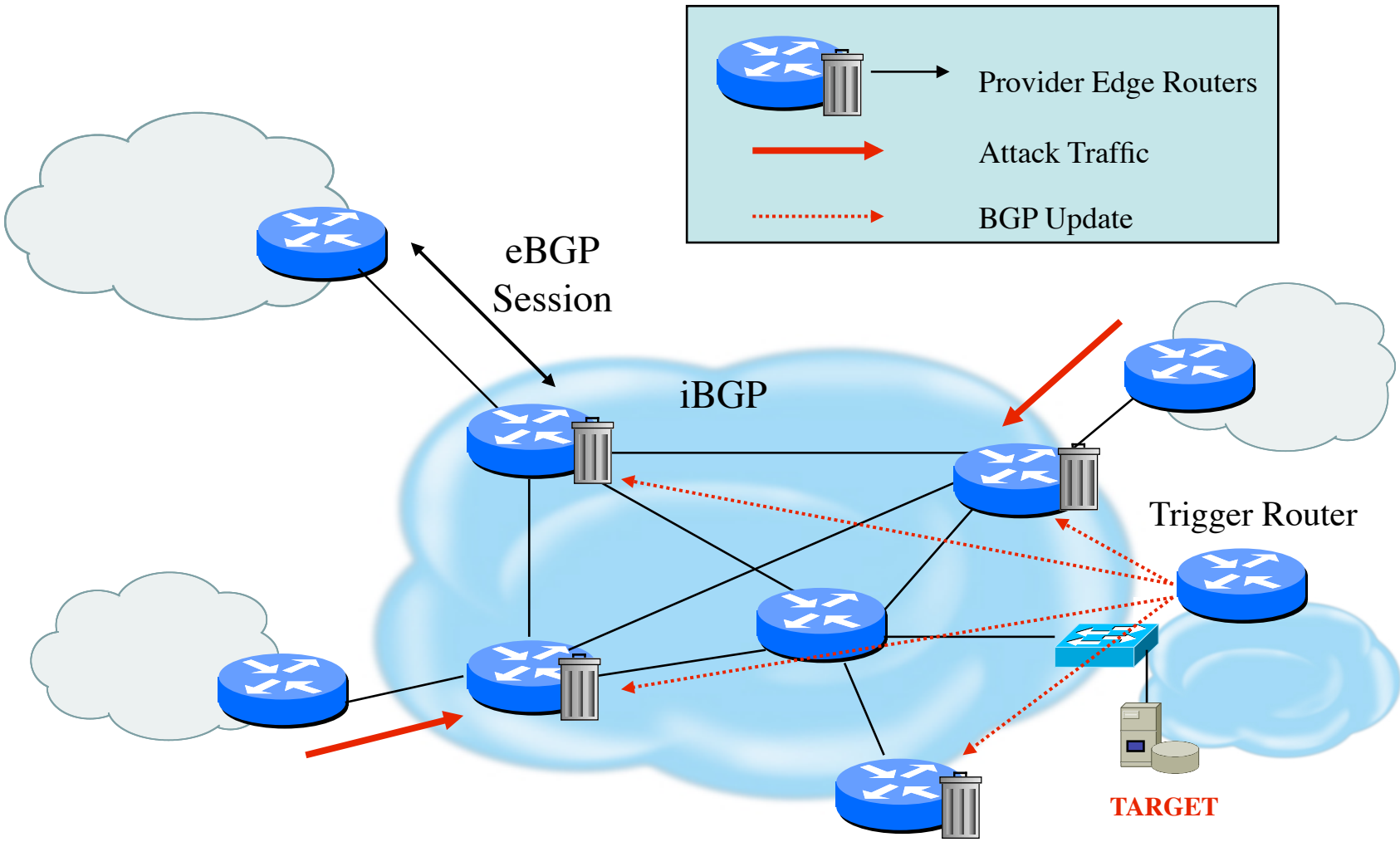


- Files stored on specific systems with limited access
- All access to these systems are authenticated and audited
- SCP is used where possible; FTP is NEVER used; TFTP still used
- Configuration files are polled and compared on an hourly basis
- Filters limit uploading / downloading of files to specific systems
- Many system binaries use MD-5 checks for integrity
- Configuration files are stored with obfuscated passwords

DoS Mitigation - RTBH Basics

- Use BGP routing protocol to trigger network wide response to an attack flow.
- Simple static route and BGP allows ISP to trigger network wide black holes as fast as iBGP can update the network.
- Unicast RPF allows for the black hole to include any packet whose source or destination address matches the prefix.
- Effective against spoofed and valid source addresses.

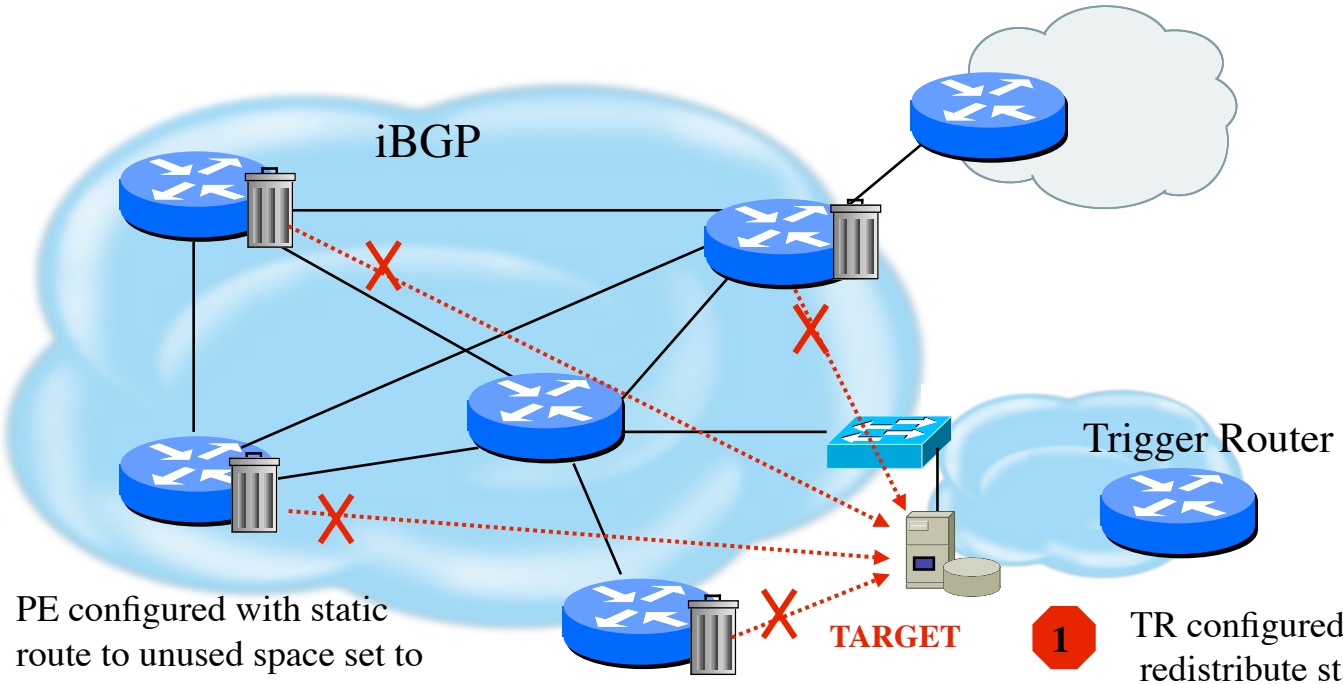
RTBH in the Network



Destination-Based RTBH

Steps:

1. Preparation
2. Trigger
3. Withdrawal



1. PE configured with static route to unused space set to Null0 (192.0.2.6/32 set to Null0)
2. Receives iBGP update which states next hop for target is 192.0.2.6/32
3. Installs new (valid) route to target

N OTE: All traffic to the target is dropped, even legitimate traffic

1. TR configured to redistribute static into every iBGP peer
2. Add static route which sets next hop to target destination (192.0.2.6)
3. Manually remove static route which causes BGP route withdrawl

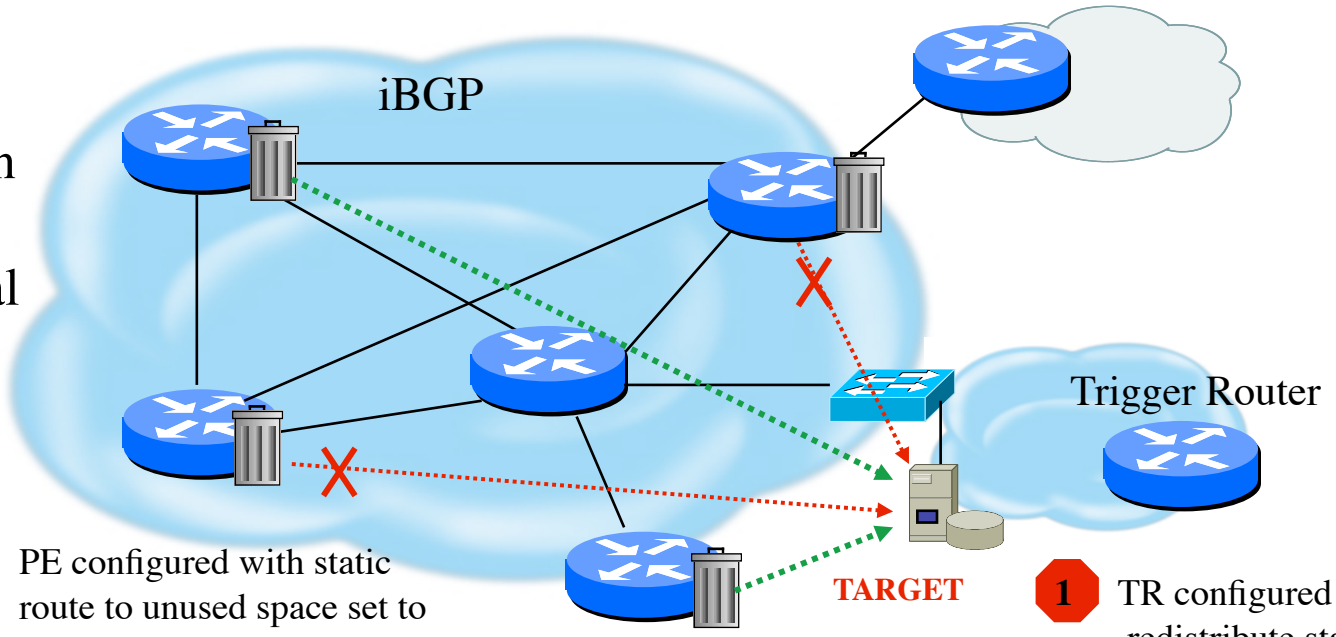
Source-Based RTBH

- Ability to drop packets at network edge based on specific source address
- Permits legitimate traffic from reaching target destination
- Depends on uRPF
- Packet dropped if:
  - If router has no entry for source IP address
  - If source IP address entry points to Null0

Source-Based RTBH

Steps:

1. Preparation
2. Trigger
3. Withdrawal

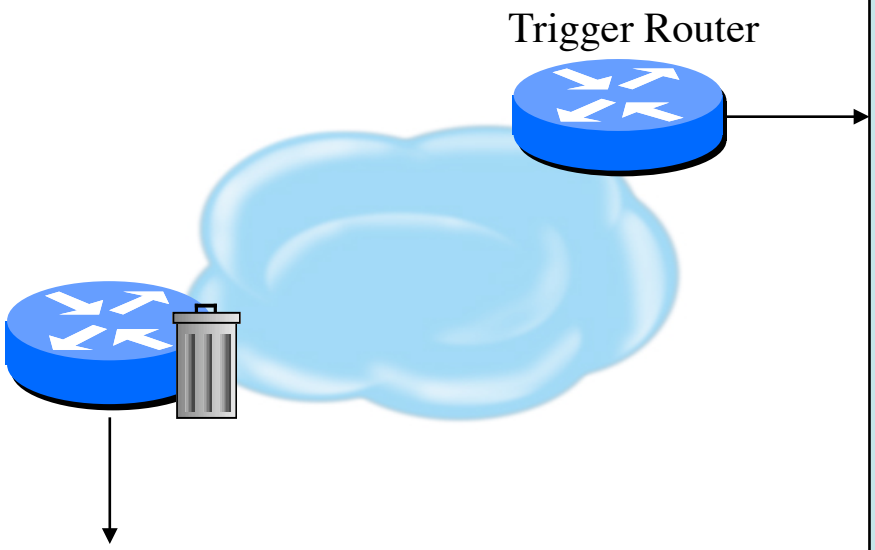


1. PE configured with static route to unused space set to Null0 (192.0.2.6/32 set to Null0) and loose mode uRPF on external interfaces
2. Receives iBGP update which states next hop for target is 192.0.2.6/32. All traffic from source IP will fail loose uRPF check.
3. Installs new (valid route to target

N OTE: Only traffic from the attack sources get dropped

1. TR configured to redistribute static into every iBGP peer
2. Add static route which sets next hop to target destination (192.0.2.6)
3. Manually remove static route which causes BGP route withdrawl

RTBH Configuration Example




```
interface Null0
no ip unreachables
!
ip route 192.0.2.1 255.255.255.255 null0
```

```
interface Null0
! avoid backscatter traffic
no ip unreachables
!
router bgp 6665
redestributre static route-map bh-trig
!
route map bh-trig permit 10
  match tag 66
  set ip next-hop 192.0.2.1
  set local-preference 200
  set origin igp
! ensure edge router does not readvertise
! prefix to any eBGP peer
  set community no-export
!
! make sure no other static routes affected
! by the bh-trig route map
route-map bh-trig deny 22
!
! the manually configured trigger
ip route 192.168.33.0 255.255.255.0 null0 tag66
```

Additional RTBH Considerations

- Avoid intentionally/unintentionally dropping legitimate traffic
- Deploy secure BGP features
  - Neighbor authentication
  - Prefix filters
  - 'TTL hack'
- Use prefix filters at edge and trigger routers to ensure essential services (e.g. DNS) not black-holed by mistake

The Changing Landscape in Network Level Security

- Peer-to-Peer Networking
  - Networks become pipes not enforcement points
  - Hosts require interaction with network elements
- Mobile Environments
  - Changing addresses
  - Location changes affect security enforcement
- What Impact Does IPv6 Have?
- Who Has Control?

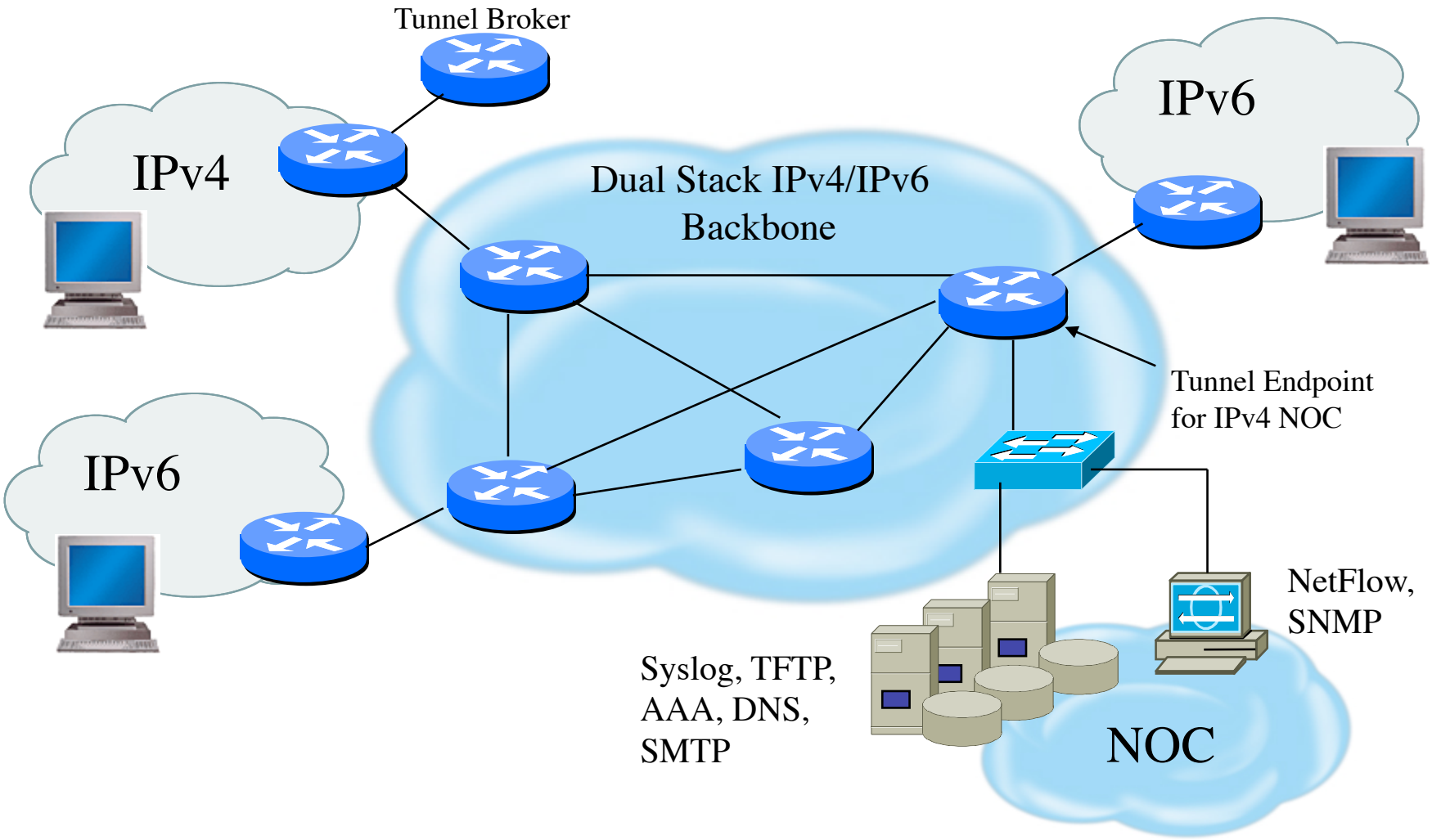
What Is The Same / What Is Different

- Same for IPv4 and IPv6
  - Security Properties
  - Security Services
- Different for IPv6 Architectures
  - Protocol Operation
  - More Automation
  - Scalable Mobile Hosts
  - Potential Application Integration

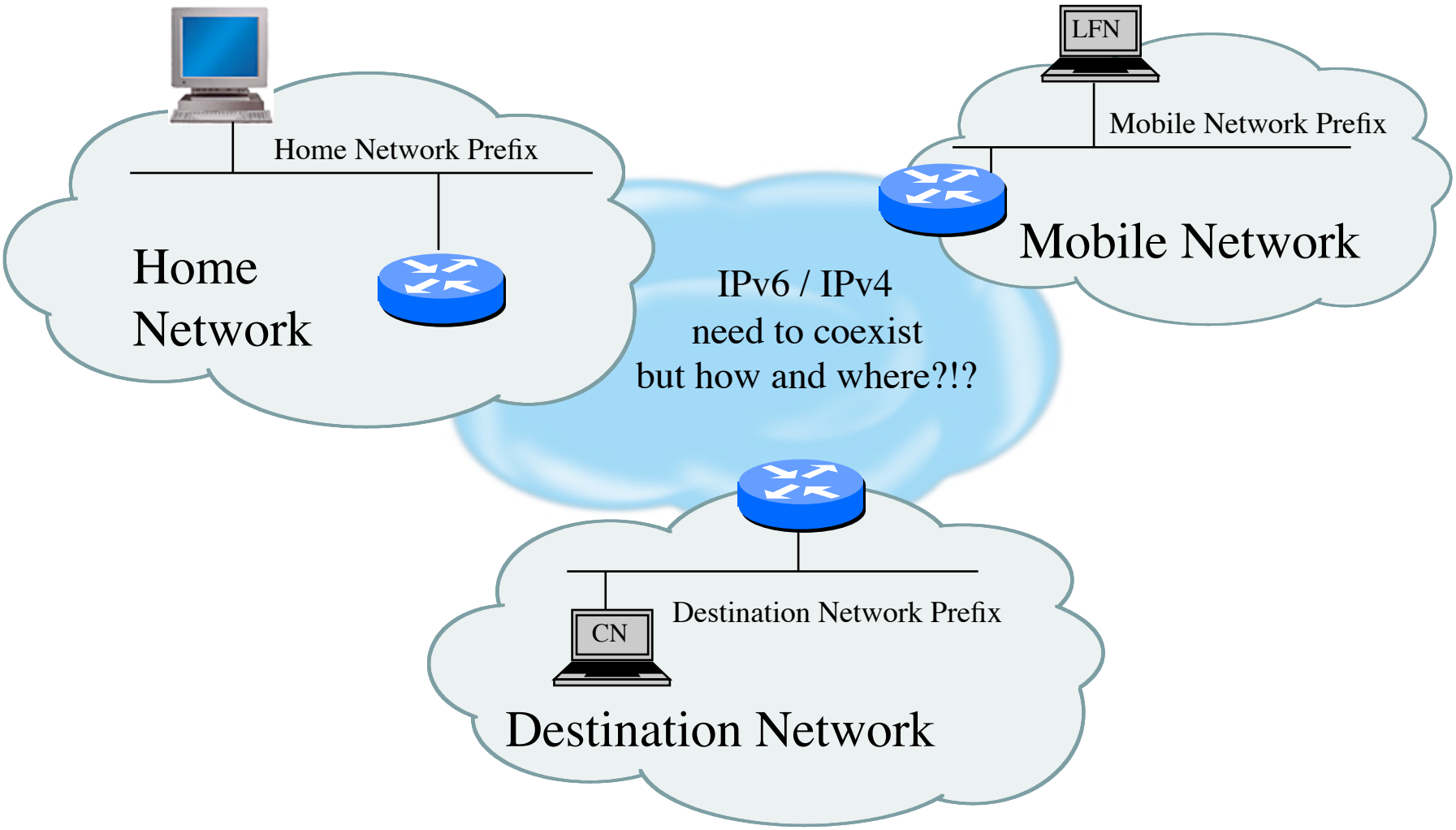
What Needs To Be Considered

- Where is automation advantageous versus a security risk?
- How will IPv4 content be accessible?
  - Is NAT a security feature or a simple way of getting access to the global Internet (without paying for it)?
  - Where is an address translation capability required?
- Where are network-based security mitigation techniques reliably advantageous versus a hindrance?
- What technologies need to be made easier to deploy to be operationally viable?
- What security services are being used to adhere to security policy requirements but are instantiations of IPv4 architecture limitations?

Sample IPv6 Architecture



Mobile Devices



IPv6 Automation

- Protocol Capabilities
  - Neighbor Discovery allows nodes to easily find one another
  - Router Advertisements enable nodes to automatically create their own globally reachable IPv6 address
- Security Issues
  - Redirect attacks
  - Denial-of-Service attacks
  - Neighbor solicitation spoofing
  - Neighbor advertisement spoofing
  - Neighbor Unreachability Detection failure
  - Duplicate Address Detection DoS attack

Architecture Considerations

- Addressing / Naming
  - What subnet boundaries make sense
    - your own network infrastructure
    - filtering considerations
  - Endpoint Identifier management
    - address automation vs obscurity vs auditability
  - DNS and DHCPv6 Considerations
- Native Routing vs Tunnels
- Management
- Security

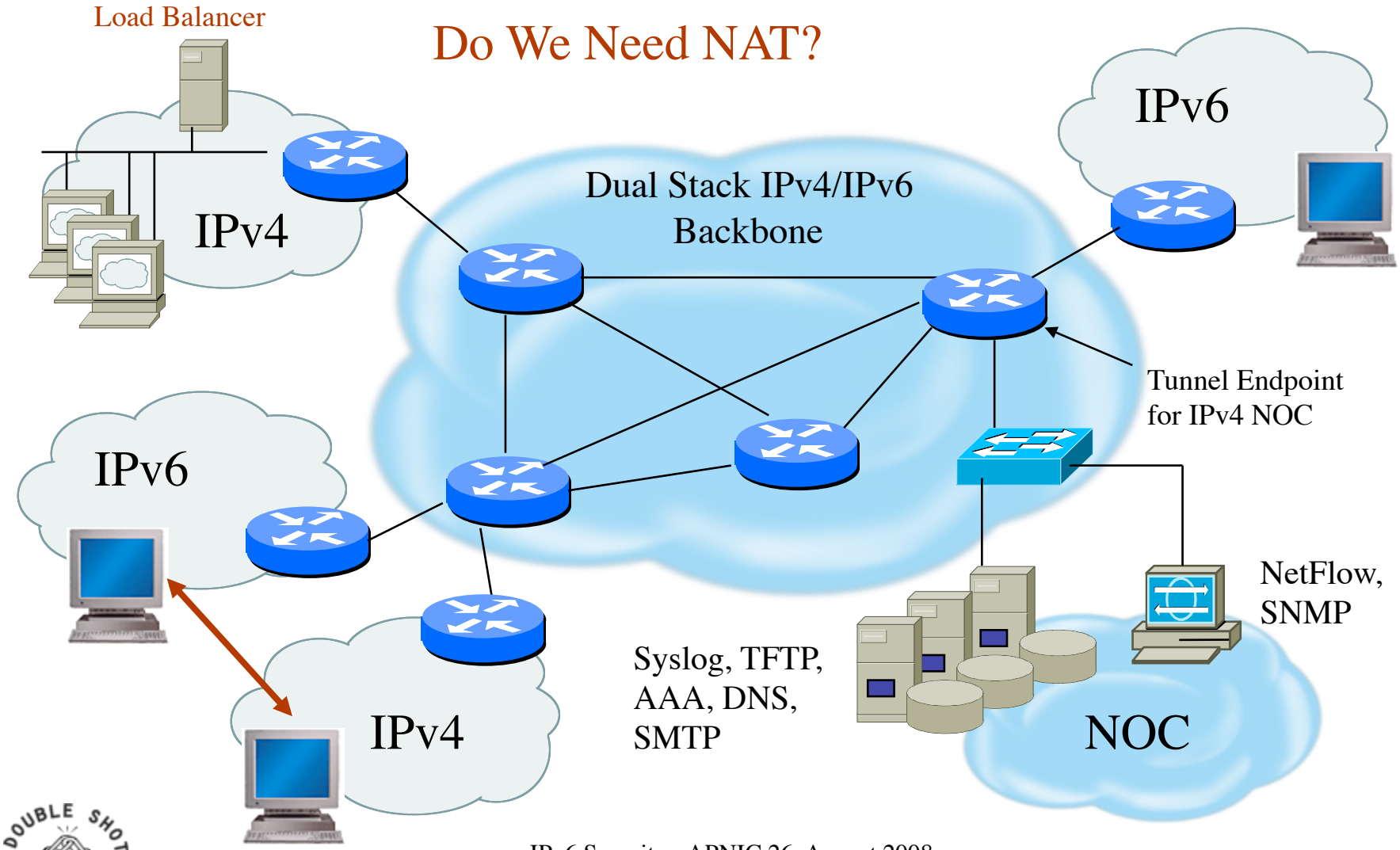
SeND Capabilities

- SeND protects against:
  - Spoofed Messages To Create False Entries In Neighbor Cache
  - Neighbor Unreachability Detection Failure
  - Duplicate Address Detection DoS Attack
  - Router Solicitation and Advertisement Attacks
  - Replay Attacks
  - Neighbor Discovery DoS Attacks
- SeND does NOT:
  - Protect statically configured addresses
  - Protect addresses configured using fixed identifiers (I.e.EUI-64)
  - Provide confidentiality
  - Compensate for unsecured link-layer
    - No guarantee that payload packets came from node that used SEND

Tunneling Considerations

- Manually configured tunnels are not scalable
- Automated tunnels require more diligence to provide effective security services
- Deployments of 6to4, ISATAP and Teredo all require layered security models
  - Perform ingress firewall sanity checks
  - Log and audit tunneled traffic
  - Provide authentication where possible
  - Use IPsec where appropriate

Network Address Translation



IPv6 Security Enhancements

- Fragmentation
  - Prohibited by intermediary devices
  - Overlapping fragments are not allowed
  - Devices must drop reassembled packets that are less than 1280 bytes
- Broadcasts
  - Removes concept of dedicated broadcasts
  - Specific language to avoid ICMPv6 broadcast amplification attacks
- IPsec
  - Defined into the base protocol spec

Hybrid Security Model

- Defense in Depth
  - Security services in network infrastructure
  - Security services on end host
- Provides gradual move to native v6
  - Add IPv6 capability in places that require dual-stack
  - If services can support native IPv6, deploy native
- Maintains existing policy controls
- Performance vs management tradeoff

End Host IPv6 Security Guidelines

- Basic Principles
  - Address assignment is performed in a reliable manner and cannot be spoofed
  - Traffic sourced from or destined to an end-host can be protected from modification, deletion or spoofing
  - Malicious behavior can be detected and mitigated
- Addressing recommendations
  - Use stateless auto-configuration when low probability that spoofing can occur
  - Use DHCPv6 if need to have control over addresses
  - Use standard but non-obvious static addresses for critical systems
- Hardening the host
  - Restrict access to the client or server to authenticated and authorized individuals
  - Monitor and audit access to the client and server
  - Turn off any unused services on the end node
  - Use host firewall capabilities to control traffic that gets processed by upper layer protocols
  - Use virus scanners to detect malicious programs
- Protecting traffic between hosts
  - Use IPsec

Conclusions

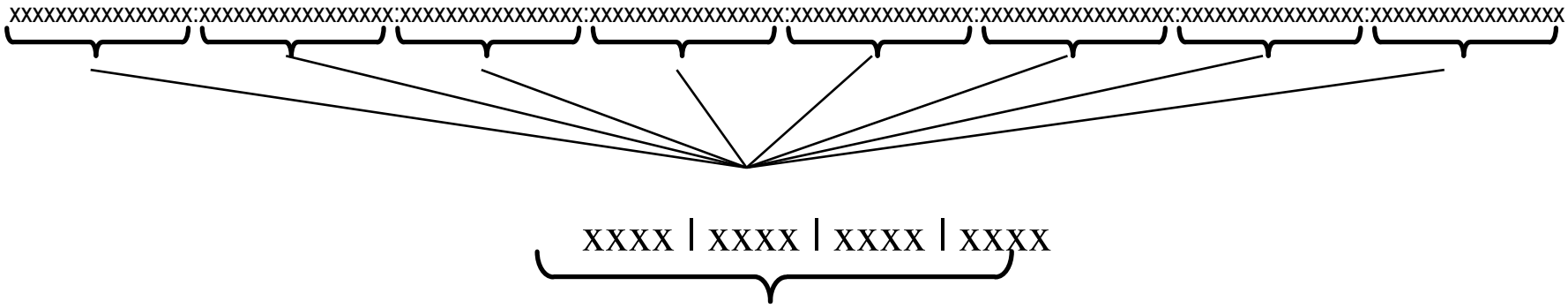
- Many similar issues for security regardless of IPv4/IPv6
- Security policies may need to be modified to enable end-to-end encryption
- Greater security efficiencies if IPv4 security architectures are NOT blindly mimicked (reduce use of NAT)
- Distributed security management is essential
- Layered defense enhanced with more effective end-host security services
- Identify actual versus perceived risks when deploying IPsec security services….i.e. use IPsec effectively!

Filtering / Firewalls

General Firewall BCP

- Explicitly deny all traffic and only allow what you need
- The default policy should be that if the firewall doesn't know what to do with the packet, deny/drop it
- Don't rely only on your firewall for all protection of your network
- Implement multiple layers of network protection
- Make sure all of the network traffic passes through the firewall
- Log all firewall exceptions (if possible)

IPv6 Addressing Review

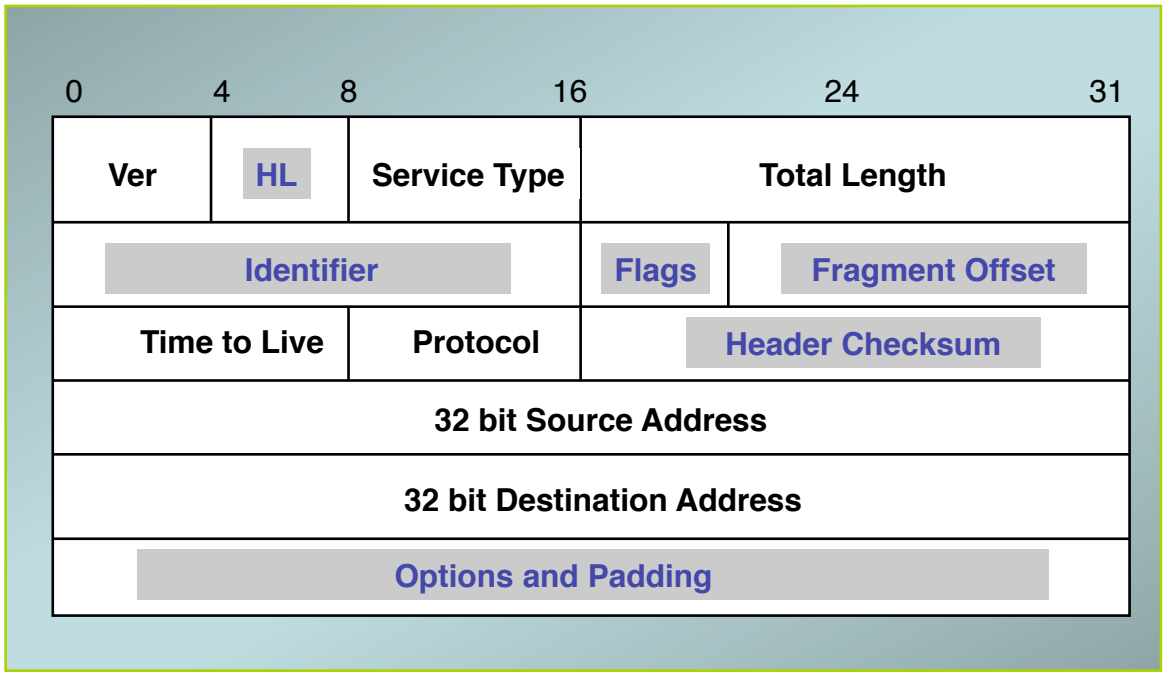


0000: 0	0100: 4	1000: 8	1100: C
0001: 1	0101: 5	1001: 9	1101: D
0010: 2	0110: 6	1010: A	1110: E
0011: 3	0111: 7	1011: B	1111: F

2001:DB8::/32
2001:DB8:0:0::/64
2001:DB8:0:0:0:0:FFFE::/112

IPv4 Header

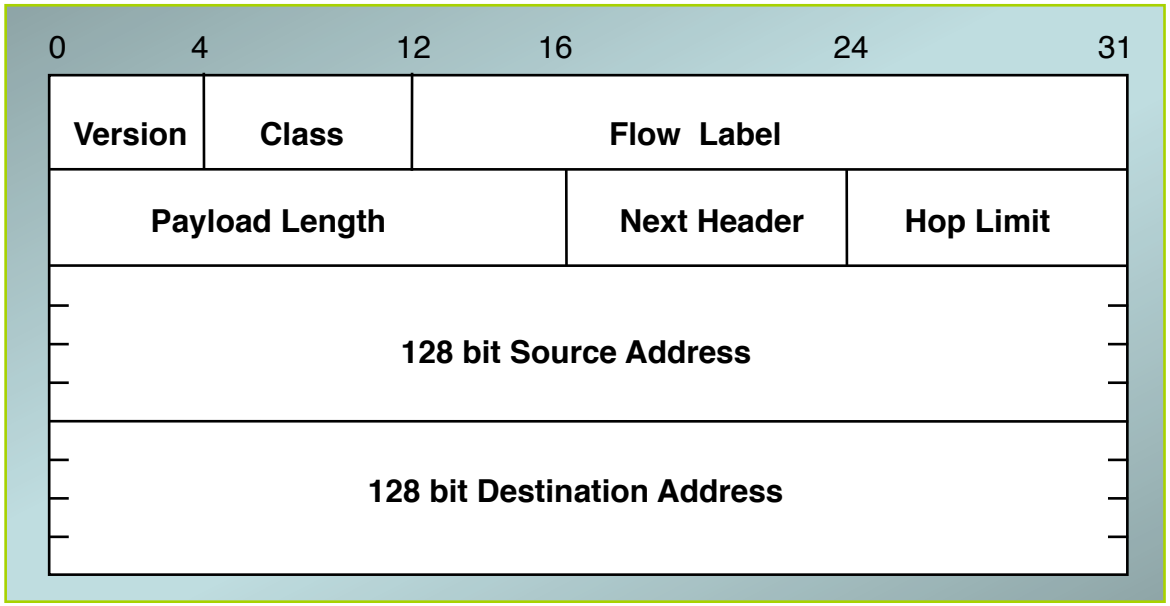
20 octets + options: 13 fields, including 3 flag bits



shaded fields are absent from IPv6 header

IPv6 Header

40 octets, 8 fields



Summary: IPv4/IPv6 Header Changes

- Streamlined
  - Fragmentation fields moved out of base header
  - IP options moved out of base header
  - Header Checksum eliminated
  - Header Length field eliminated
  - Length field excludes IPv6 header
- Revised
  - Time to Live = Hop Limit
  - Protocol = Next Header
  - Precedence & TOS = Traffic Class
  - Addresses increased from 32 bits to 128 bits
- Extended
  - Flow Label field added

IPv6 Extension Headers

- Carry the additional options and padding features that are part of the base IPv4 header
- Extension headers are optional and placed after the base header
- There can be zero, one, or more EH's between the v6 header and the upper-layer protocol header
  - Identified by the Next Header field
- Ordering is important
  - Must be processed in the order listed

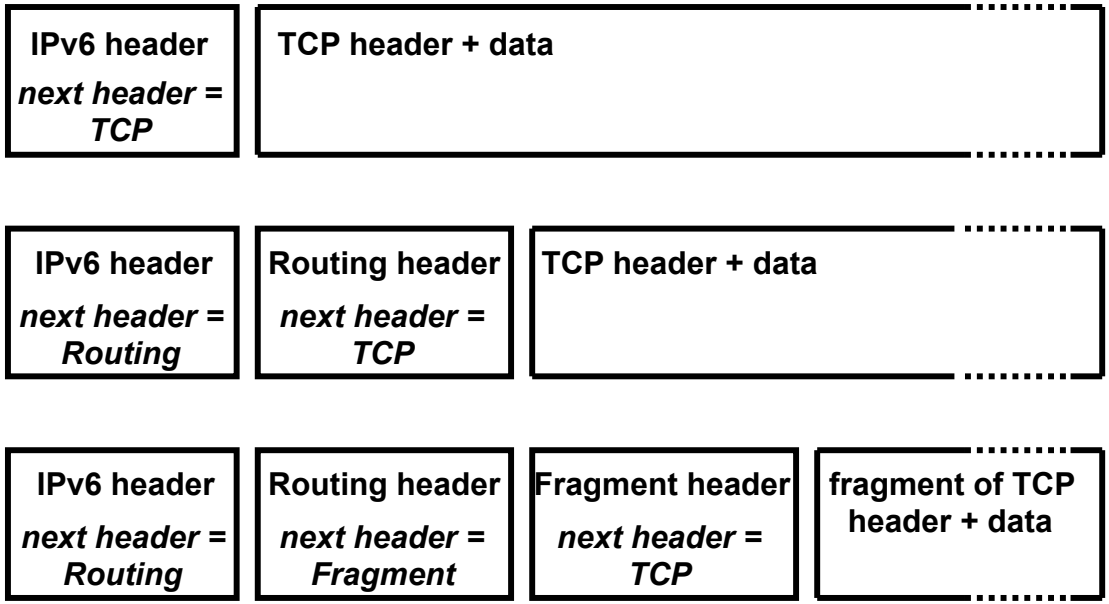
Currently Defined IPv6 Extension Headers:

- Hop-by-Hop Options (0)
- Routing Header (43)
- Fragment Header (44)
- ESP Header (50)
- Authentication Header (51)
- Destination Options (60)

Other Extension Header Values:

- TCP upper-layer (6)
- UDP upper-layer (17)
- ICMPv6 (58)
- No Next Header Present (59)

IPv6 Extension Header Chaining



Extension Header Ordering

- Hop-by-Hop
- Destination Options*
  - For options processed by the 1st destination address plus subsequent destinations listed in the routing header
- Routing
- Fragment
- Authentication
- Encapsulating Security Payload
- Destination Options
- Upper-Layer header

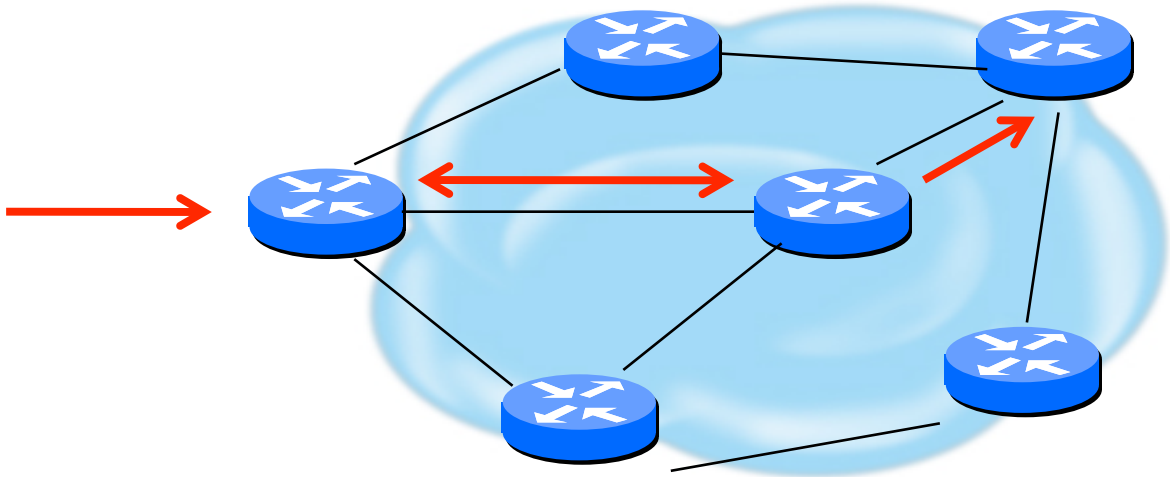
Routing Header: RFC 2460 Text

- The routing header is used by an IPv6 source to list one or more intermediate ***nodes*** to be "visited" on the way to packet's destination.
- Each extension header should occur at most once, except for the destination options header which should occur at most twice.
- IPv6 nodes must accept and attempt to process extension headers ***in any order*** and ***occurring any number of times*** in the same packet.

Routing Header Issue

A single RH of Type 0 may contain multiple intermediate node addresses, and the same address may be included more than once in the same RH0.

If the routing header contains a repetition of a pair of addresses of the form A B A B A B … If this A B pair were repeated 3 times then a single packet directed at A would traverse the path A B 3 times, and B A twice. If such packets were generated at a total rate of 1 Mbps then the path between A and B would experience a total of 5Mbps of traffic.

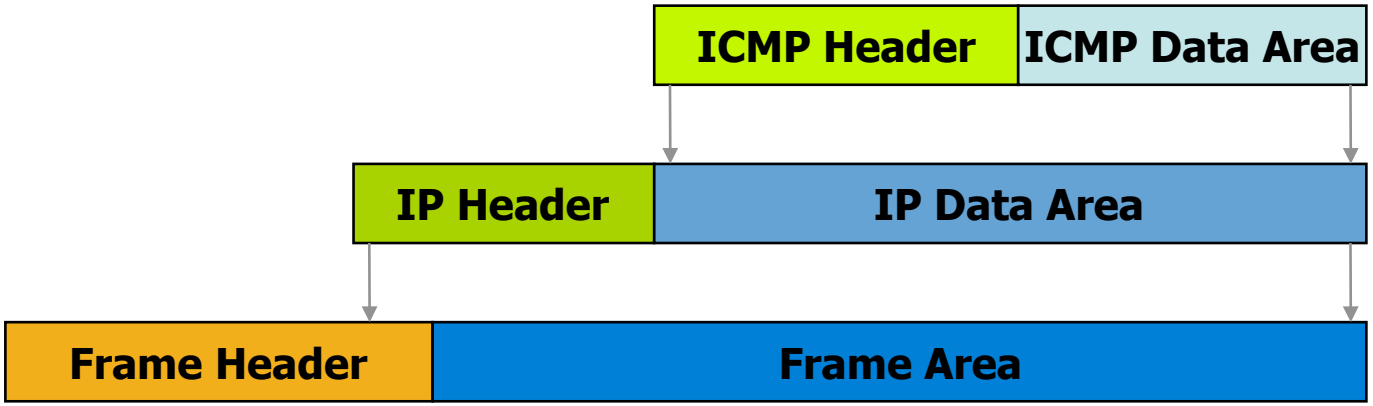


Routing Header Processing

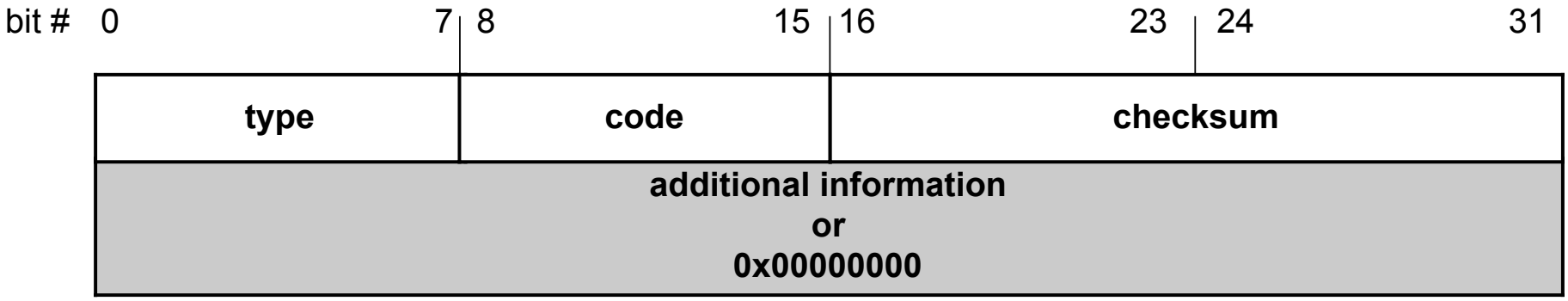
- Disabling processing still allows all other hosts to be used for attack
- Dropping is required for ISP's
- RFC 5095 – Deprecation of RH0
- Until rfc5095 implemented:
  - Use ingress filtering for RH0 traffic
  - RH Type 2 is required for mobility so have to ensure that only RH0 traffic is blocked

Internet Control Message Protocol

- Original specification in RFC 792
- Used to report problems with delivery of IP packets
- Supports Path MTU (PMTU) Discovery between a sender and a receiver, which helps to optimize performance of data delivery between pairs or hosts by avoiding fragmentation en route



ICMP Message Format



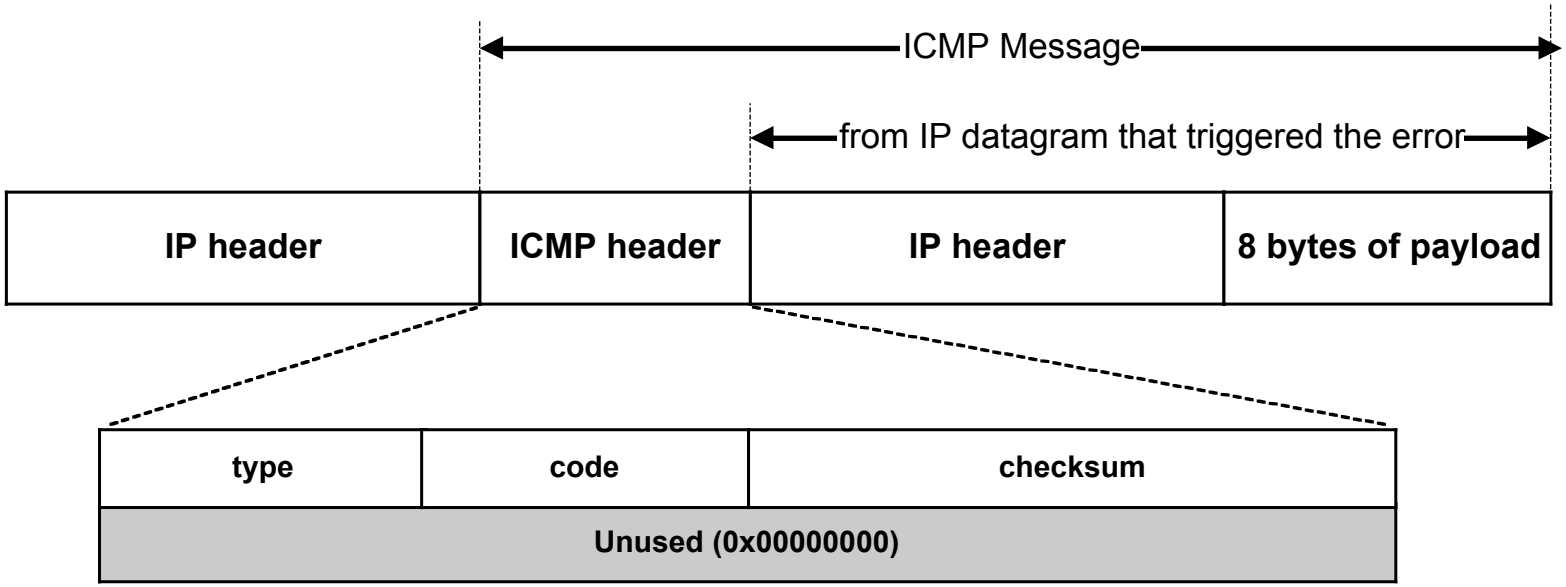
4 byte header:

- Type (1 byte): type of ICMP message
- Code (1 byte): subtype of ICMP message
- Checksum (2 bytes): similar to IP header checksum. Checksum is calculated over entire ICMP message

If there is no additional data, there are 4 bytes set to zero.
Each ICMP messages is at least 8 bytes long

ICMP Error Messages

- ICMP error messages report error conditions
- Typically sent when a datagram is discarded
- ICMP error messages include the complete IP header and the first 8 bytes of the payload (typically: UDP, TCP)



ICMP Message Types

Type	Message Type	Description
3	Destination Unreachable	Packet could not be delivered
11	Time Exceeded	Time to live field hit 0
12	Parameter Problem	Invalid header field
4	Source Quench	Tell host to slow down transmission due to congestion
5	Redirect	Notification that packet seems to be routed wrong
8	Echo	Ask a machine if it is alive and reachable
0	Echo Reply	Yes, I am alive
13	Timestamp Request	Same as Echo request, but with timestamp
14	Timestamp Reply	Same as Echo reply, but with timestamp

Destination Unreachable Codes

Code	Definition
0	Network Unreachable [no routing table entry available for destination network]
1	Host Unreachable [destination host should be directly reachable but does not respond to ARP requests]
2	Protocol Unreachable [protocol in protocol field of IP header is not supported at destination]
3	Port Unreachable [transport protocol at destination host cannot pass datagram to an application]
4	Fragmentation needed & Don't Fragment was set
5	Source Route failed
6	Destination Network Unknown
7	Destination Host Unknown
8	Source Host Isolated
9	Communication Destination Network is Administratively Prohibited
10	Communication Destination Host is Administratively Prohibited
11	Destination Network Unreachable for Type of Service
12	Destination Host Unreachable for Type of Service
13	Communication Administratively Prohibited
14	Host Precedence Violation
15	Precedence Cutoff Violation

Additional Codes for ICMP Types

Redirect Codes

Code	Definition
0	Redirect Datagram for the Network (or subnet)
1	Redirect Datagram for the Host
2	Redirect Datagram for the Type of Service & Network
3	Redirect Datagram for the Type of Service & Host

Time Exceeded Codes

Code	Definition
0	Time to Live Exceeded in Transit
1	Fragment Reassembly Time Exceeded

Parameter Problem Codes

Code	Definition
0	Pointer Indicates the Error
1	Missing a Required Option
2	Bad Length

ICMPv6

- Is similar to IPv4 ICMP, with a few differences:
  - ICMP is carried in an IPv6 datagram
  - A checksum is computed since ICMP is a transport protocol, relative to IPv6
  - New messages are defined for the IPv6 specification
  - In an error message, the original datagram is included in the error packet for easier recovery by the source
- Identified by the Next Header value = 58
- ICMP header contains Type and Code fields to identify and qualify the message specifics
- Two defined ICMP classes in IPv6:
  - Error Messages
  - Informational

ICMPv6 Error Messages

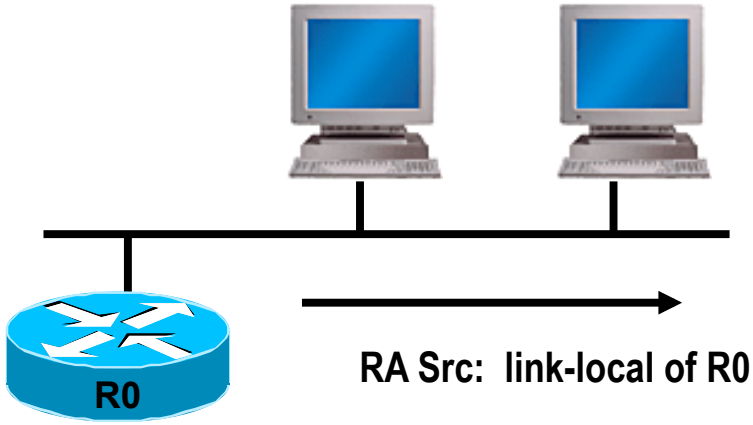
- Identified by a Type field value between 0 and 127
- Message Types:
  - destination unreachable
    - no route
    - administratively prohibited (i.e. firewalls)
    - address unreachable
    - port unreachable
  - packet too big
  - time exceeded
  - parameter problem
    - erroneous header field
    - unrecognized next header type
    - unrecognized option

ICMPv6 Informational Messages
(Type: 128-255)

ICMP Number	*Message Type*
128	Echo request
129	Echo reply
130	Multicast group membership query
131	Multicast group membership report
132	Multicast group membership termination
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
137	Redirect
138	Router Renumbering
139	Node Information query
140	Node Information reply
141	Inverse Neighbor Solicitation
142	Inverse Neighbor Advertisement

IPv6 Router Advertisement

- Sent periodically or in response to a Router Solicitation message
- Periodic RA's are sent to the all-nodes multicast address "ff02::1"
- RA messages contain information that inform the hosts about link information needed for auto-configuration



RA Src: link-local of R0 interface

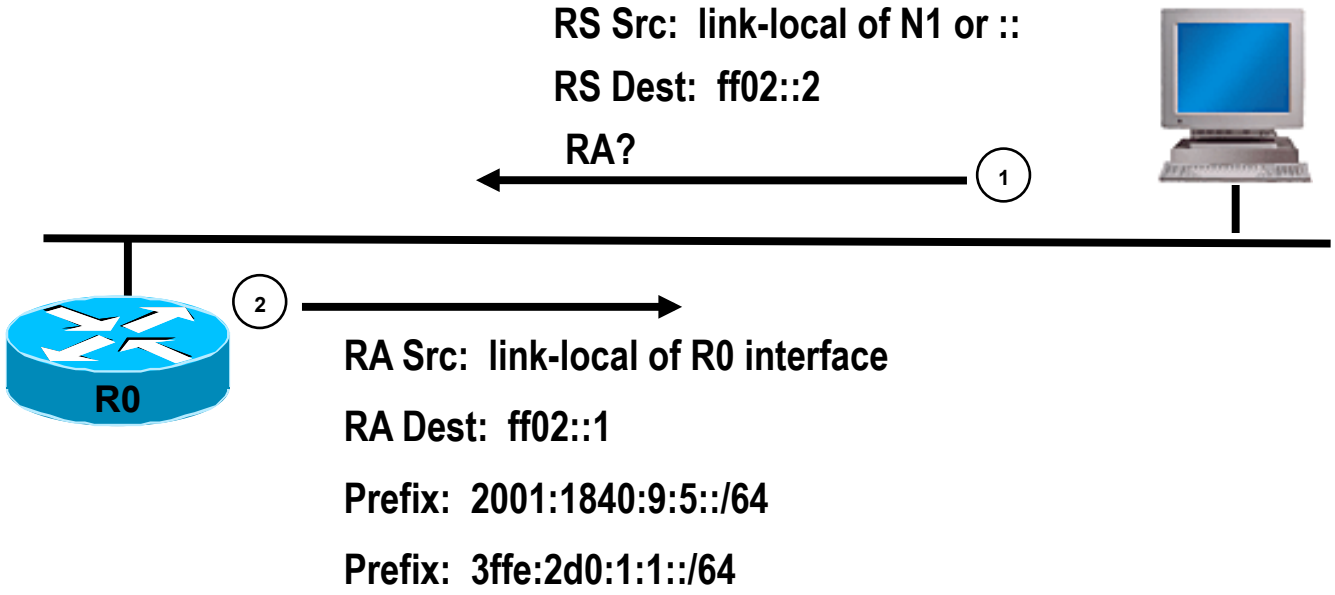
RA Dest: ff02::1

Prefix: 2001:1840:9:5::/64

Prefix: 3ffe:2d0:1:1::/64

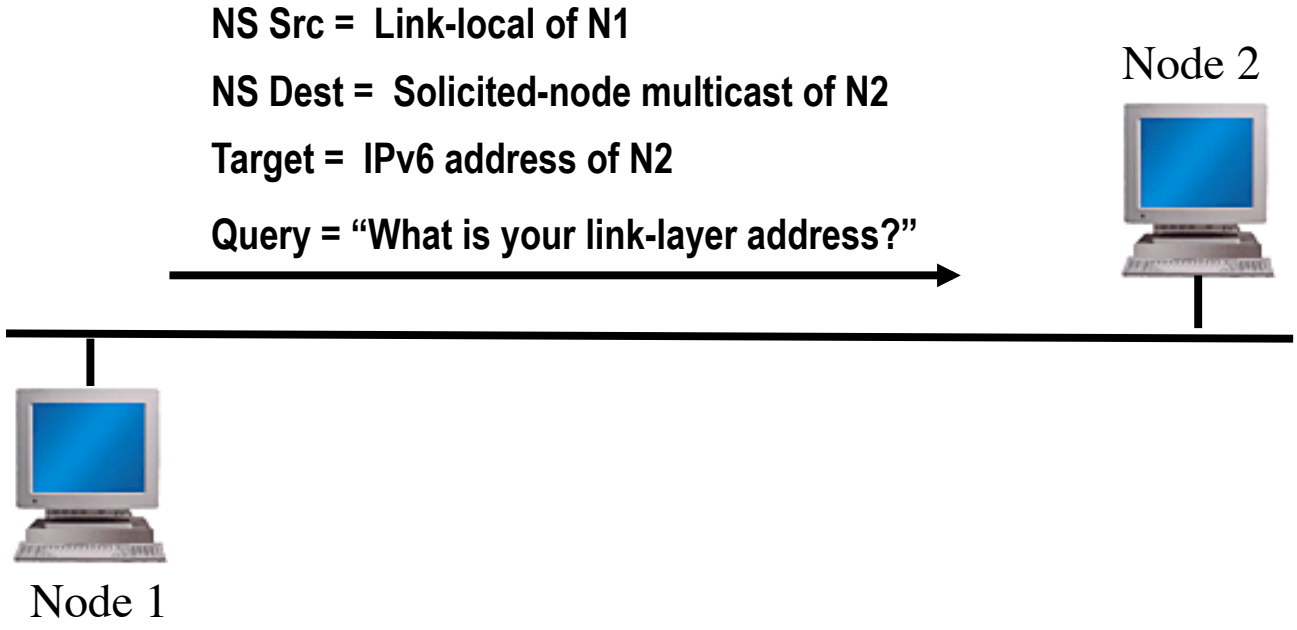
IPv6 Router Solicitation

- Sent at host start-up or to solicit a Router Advertisement immediately
- RS messages are usually sent to the all-routers multicast address "ff02::2"
- RS source address could be the link-local address of the sending node, or the unspecified "::" address



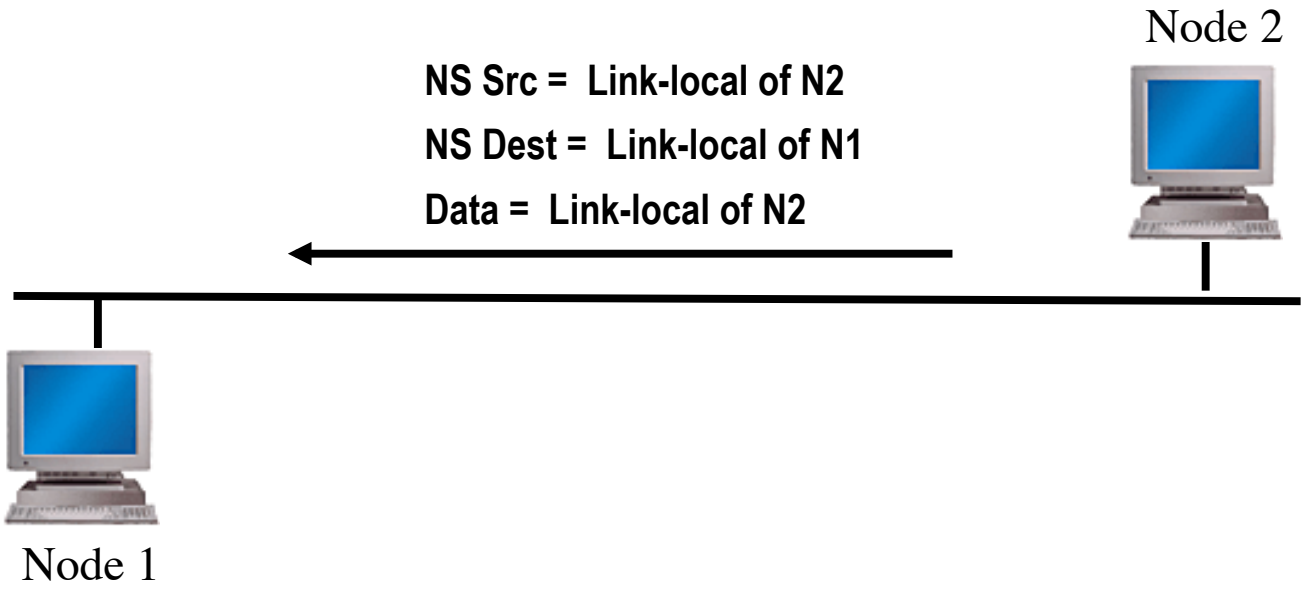
IPv6 Neighbor Solicitation

- Used by nodes for link-layer to IP-layer address resolution
- For link-layer address resolution, the solicited-node multicast address is used as the destination of the request (vs. broadcast in IPv4 ARP)
- Also used in the Duplicate Address Detection (DAD) and Neighbor Unreachability Detection (NUD) processes



IPv6 Neighbor Advertisement

- Sent in response to an NS or unsolicited to propagate new information
- Neighbor Advertisements contain:
  - Router flag: to indicate whether this neighbor is a router
  - Solicited flag: to indicate whether this NA is in response to a NS
  - Override flag: to indicate whether this information should override an existing neighbor cache entry
- NA's in response to an address resolution request are unicast to the solicitor



IPv6 Filtering Considerations

- IPv6 addressing architecture will simplify or complicate filters….carefully think about it.
- Routing filters are usually more optimal than packet filters but have less granularity
  - Routing filters affect the routes that are accepted and sent between routers and therefore forward or drop traffic based on reachability information
  - Packet filters are used to allow or deny data packets from being processed or forwarded by a device based on the IP header information.
- Best policy is to deploy filtering mechanisms that will drop any unwanted traffic as close to source as possible

Ingress Packet Filters To Consider

- Accept all ICMPv6 packets for Neighbor Discovery and Path MTU Discovery that is a function necessary for the communication with IPv6
- Reject the packets which contain relevant special-use prefix in the ***source*** address field
  - ::1/128 : loop back address
  - ::/128 : unspecified address
  - ::/96 : IETF reserved address;IPv4-compatible IPv6 address
  - ::ffff:0:0/96 : IPv4-mapped IPv6 address
  - ::/8 : reserved
  - fc00::/7 : unique-local address
  - ff00::/8 : multicast address
  - 2001:db8::/3 : documentation addresses

Ingress Packet Filters To Consider(cont.)

- Reject the packets which contain relevant special-use prefix in the ***destination*** address field
  - ::1/128 : loop back address
  - ::/128 : unspecified address
  - ::/96 : IETF reserved address;IPv4-compatible IPv6 address
  - ::ffff:0:0/96 : IPv4-mapped IPv6 address
  - ::/8 : reserved
  - fc00::/7 : unique-local [fc00::/16] and site-local [fc00::/10] address
  - 2001:db8::/32 : documentation address
- Reject the packets which have your own prefix in the source address field
- Reject packets that use the routing header Care must be taken not to reject ICMPv6 packets whose source address used with Duplicate Address Detection is the unspecified address (::/128). If all of ICMPv6 is accepted, then there is no problem although ordering of the filters needs to be carefully thought through.

Egress Packet Filters To Consider

- Permit sending all ICMPv6 packets for Neighbor Discovery and Path MTU Discovery that is a function necessary for the communication with IPv6
- Deny sending the packets which contain special-use prefix in the source address field
  - ::1/128 : loop back address
  - ::/128 : unspecified address
  - ::/96 : IETF reserved address;IPv4-compatible IPv6 address
  - ::ffff:0:0/96 : IPv4-mapped IPv6 address
  - ::/8 : reserved
  - fc00::/7 : unique-local address
  - ff00::/8 : multicast address
  - 2001:db8::/32 : documentation address
- Deny sending packets that use the routing header [unless using mobility features]
- Deny sending packets with destination address in the 6to4 reserved address range (2202::/16) if not supporting 6to4 services (i.e. relays) and not providing transit services
- Deny sending packets with destination address in the Teredo address range (2001::/32) if not running a Teredo relay or offering a Teredo transit service
- Multicast address should only be in source address field.

Allow Following ICMPv6 Through A Firewall

- ICMPv6 type 1 code 0: no route to destination
- ICMPv6 type 2: packet too big (required for PMTUD)
- ICMPv6 type 3: time exceeded
- ICMPv6 type 4: parameter problem (informational when IPv6 node has problem identifying a field in the IPv6 header or in an extension header)
- ICMPv6 type 128: echo request
- ICMPv6 type 129: echo reply

Allow Following ICMPv6 To/From A Firewall

- ICMPv6 type 2: packet too big – firewall device is not allowed to fragment IPv6 packets going through it and must be able to generate this message for correct PMTUD behavior
- ICMPv6 type 4: parameter problem
- ICMPv6 type 130-132: multicast listener messages – in IPv6 a routing device must accept these messages to participate in multicast routing
- ICMPv6 type 133-134: router solicitation and advertisement – needed for IPv6 autoconfiguration
- ICMPv6 type 135-136: neighbor solicitation and advertisement – used for duplicate address detection and layer2-to-IPv6 address resolution

Cisco Filters (Access-Lists)

- IPv6 access-lists (ACL) are used to filter traffic and restrict access to the router
- IPv6 extended access lists add support for option header and upper layer filtering
- Cisco specific filtering characteristics
  - A reference to an empty ACL will permit any any
  - Implicit permit rule for neighbor discovery
  - Implicit deny any any as final rule in each ACL

Configuring Cisco IPv6 ACLs

- Creating the IPv6 ACL

  [no] ipv6 access-list <name>

- Defining the IPv6 ACL entry

  [no] permit | deny ipv6 | <*protocol*> any | host <*src*> | src/len [sport] any | host <*dest*> | dest/len [dport] [reflect <*name*> [timeout <*secs*>]] [fragments] [routing] [dscp <*val*>] [flow-label <*val*>] [time-range <*name*>] [log | log-input] [sequence <*num*>]

- Applying an ACL to an interface

  interface s0/0

  ipv6 traffic-filter ipv6_in in

  ipv6 traffic-filter ipv6_out out

- Restricting access to the router

  line vty 0 4

  ipv6 access-class vty-filter in

Monitoring Cisco IPv6 ACLs

- Show the IPv6 ACL configuration

  show ipv6 access-list [name]

- Clearing the IPv6 ACL match count

  clear ipv6 access-list [name]

As with any filter configuration, ordering is important since all entries are checked sequentially. Ensure that most frequent 'hits' are on top of the list.

Cisco and RH0 Filtering

- To disable processing of all types routing headers on 12.2(15)T and up one can use:

```
no ipv6 source-route
```

Note that this will still forward these packets on to other hosts which can be vulnerable. This statement also affects perfectly valid Routing Headers of Type 2 which are used by Mobile IPv6.

- If possible upgrade to 12.4(2)T or higher and block only the Type 0 Routing Header (note interface specific config):

```
Router(config)#ipv6 access-list deny-sourcerouted
Router(config-ipv6-acl)#deny ipv6 any any routing-type 0
Router(config-ipv6-acl)#permit ipv6 any any
Router(config)#interface Ethernet0
Router(config-if)#ipv6 source-route
Router(config-if)#ipv6 traffic-filter deny-sourcerouted in
```

Cisco IPv6 NetFlow

- Netflow IPv6 support from 12.4 IOS releases
- Uses Netflow v9
- Activate per interface

  ipv6 flow ingress

  ipv6 flow egress
- Show status

  show ipv6 flow cache

BGP Prefix Filters To Consider

- Special-use prefixes
  - ::/0 exact : default route
  - ::1/128 : loop back address
  - ::/128 : unspecified address
  - ::/96 : IPv4-compatible IPv6 address
  - ::ffff:0:0/96 : IPv4-mapped IPv6 address
  - ::/8 or longer : reserved
  - fe80::/10 or longer : link-local address
  - fc00::/7 or longer : unique-local address
  - ff00::/8 or longer : multicast range (RFC3513)
  - fe00::/9 or longer : multicast range (RFC3513)
  - 2001:db8::/32or longer : documentation address
- Your own prefix
- The 6bone prefix (3ffe::/16)
- The 6to4 reserved address range (2002::/16) if not supporting 6to4 services (i.e. relays) and not providing transit services
- The Teredo address range (2001::/32) if not running a Teredo relay or offering a Teredo transit service

BGP Prefix Filters (RIR Allocations)

- APNIC
  - ftp://ftp.apnic.net/stats/apnic/delegated-apnic-latest
- RIPE NCC
  - ftp://ftp.ripe.net/pub/stats/ripencc/delegated-ripencc-latest
- ARIN
  - ftp://ftp.arin.net/pub/stats/arin/delegated-arin-latest
- LACNIC
  - ftp://ftp.lacnic.net/pub/stats/lacnic/delegated-lacnic-latest
- AfriNIC
  - ftp://ftp.afrinic.net/pub/stats/afrinic/delegated-afrinic-latest

BGP Simple Bogon Prefix Filter Example

- *ipv6 prefix-list ipv6-special-use-pfx deny 0::/0 le 128*
- *ipv6 prefix-list ipv6-special-use-pfx deny 0::1/128 le 128*
- *ipv6 prefix-list ipv6-special-use-pfx deny 0::/128*
- *ipv6 prefix-list ipv6-special-use-pfx deny 0::/96*
- *ipv6 prefix-list ipv6-special-use-pfx deny 0::ffff:0:0/96*
- *ipv6 prefix-list ipv6-special-use-pfx deny 0::/8 le 128*
- *ipv6 prefix-list ipv6-special-use-pfx deny fe80::/10 le 128*
- *ipv6 prefix-list ipv6-special-use-pfx deny fc00::/7 le 128*
- *ipv6 prefix-list ipv6-special-use-pfx deny fe00::/9 le 128*
- *ipv6 prefix-list ipv6-special-use-pfx deny ff00::/8 le 128*
- *ipv6 prefix-list ipv6-special-use-pfx deny 2001:db8::/32 le 128*
- *ipv6 prefix-list ipv6-special-use-pfx deny 3ffe::/16 le 128*

BGP RIR Allocation Prefix Filter Example (Needs Constant Updating)

- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2001:0500::/30 ge 48 le 48*
- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2001:0678::/29 ge 48 le 48*
- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2001::/16 ge 35 le 35*
- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2001::/16 ge 19 le 32*
- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2003::/18 ge 19 le 32*
- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2400::/12 ge 13 le 32*
- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2600::/12 ge 13 le 32*
- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2610::/23 ge 24 le 32*
- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2620::/23 ge 40 le 48*
- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2800::/12 ge 13 le 32*
- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2A00::/12 ge 13 le 32*
- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2C00::/12 ge 13 le 32*
- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2001:0DF0::/29 ge 40 le 48*
- *ipv6 prefix-list ipv6-RIR-allocations-pfx permit 2001:43F8::/29 ge 40 le 48*

IPv6 Filtering References

- RFC 4890 ‘Recommendations for Filtering ICMPv6 Messages in Firewalls’
- RFC 5156 ‘Special-Use IPv6 Addresses’
- http://www.space.net/~gert/RIPE/ipv6-filters.html
- http://www.cymru.com/Bogons/v6top.html
- NSA Router Security Configuration Guide Supplement – Security for IPv6 Routers

Many filtering recommendations are not uniform and that while similarities exist, a definitive list of what to deny and what to permit does not exist. Any environment will need to determine what is most suitable for them by using these references as guidelines.

Crypto Fundamentals

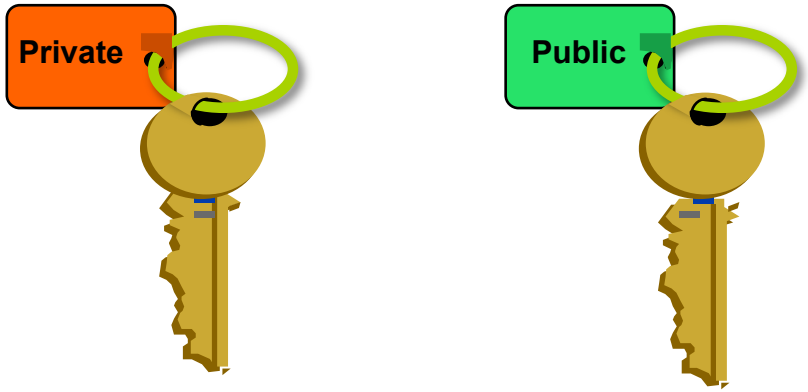
Cryptography Is Used For?

- Authentication Protocols
- Data Origin Authentication
- Data Integrity
- Data Confidentiality

Public Key Encryption

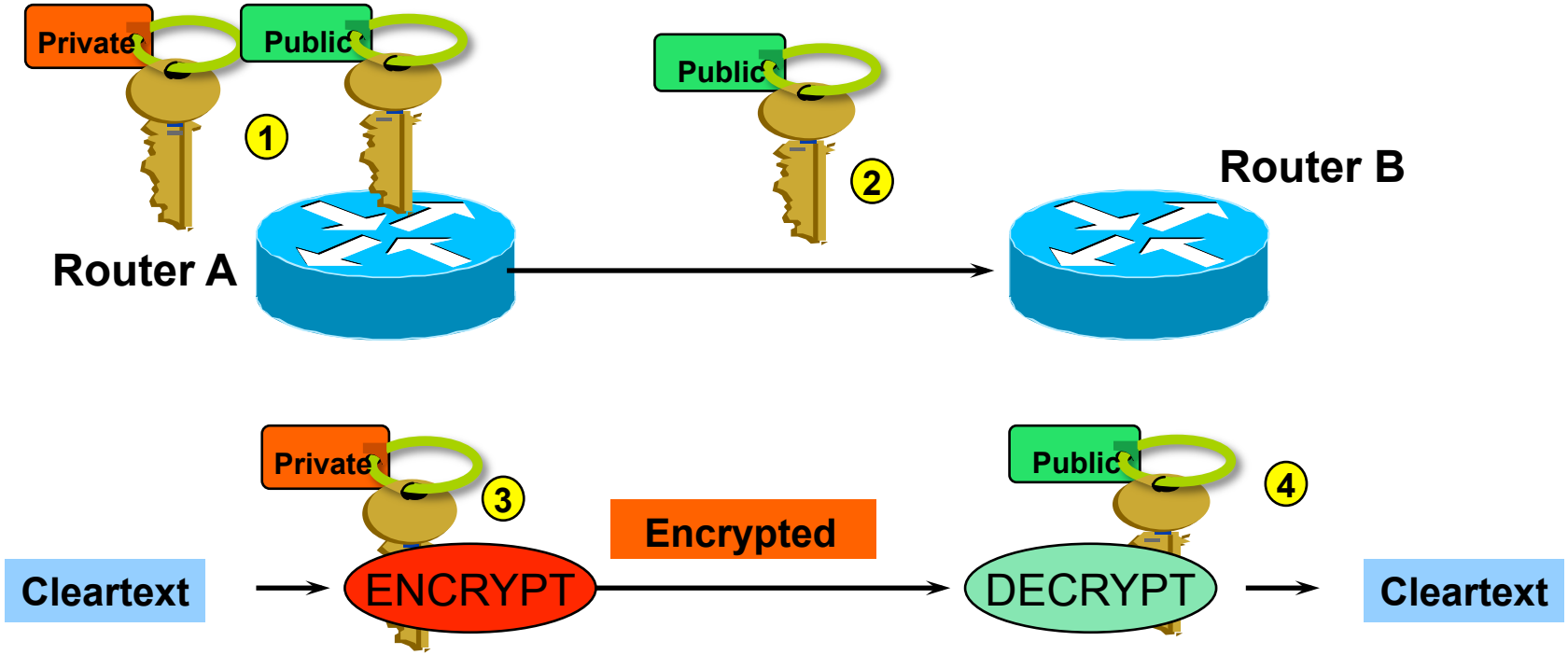
Uses public/private keys

- One key is mathematical inverse of the other
- Private key is only known by owner of the pair
- Public keys are stored in public servers



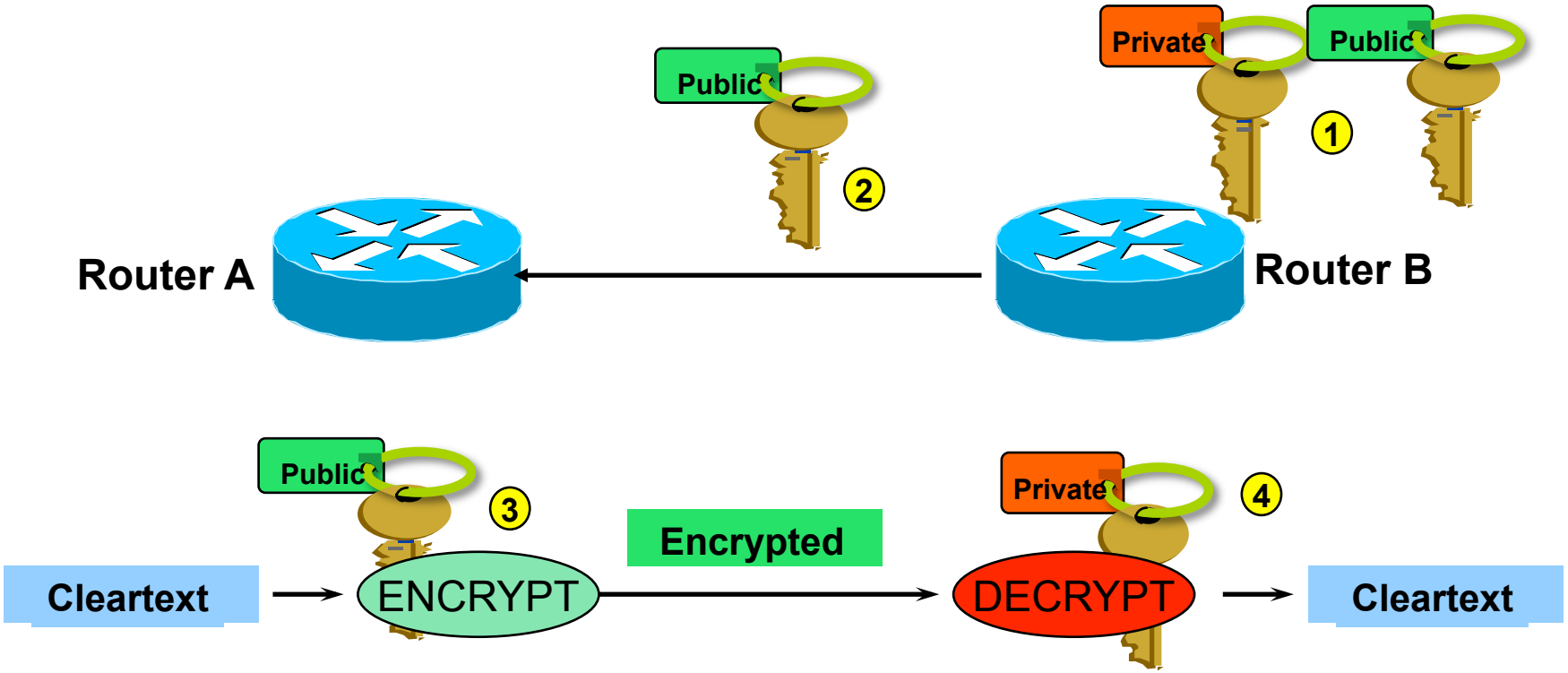
Computing Key pair is computationally expensive!!
Common Algorithms: RSA, El Gamal, DSS, ECC

Authentication and Integrity



1. Router A generates public/private key pair
2. Router A sends its public key to Router B
3. Router A encrypts packet with its private key and sends encrypted packet to Router B
4. Router B receives encrypted packet and decrypts with Router A's public key

Data Confidentiality

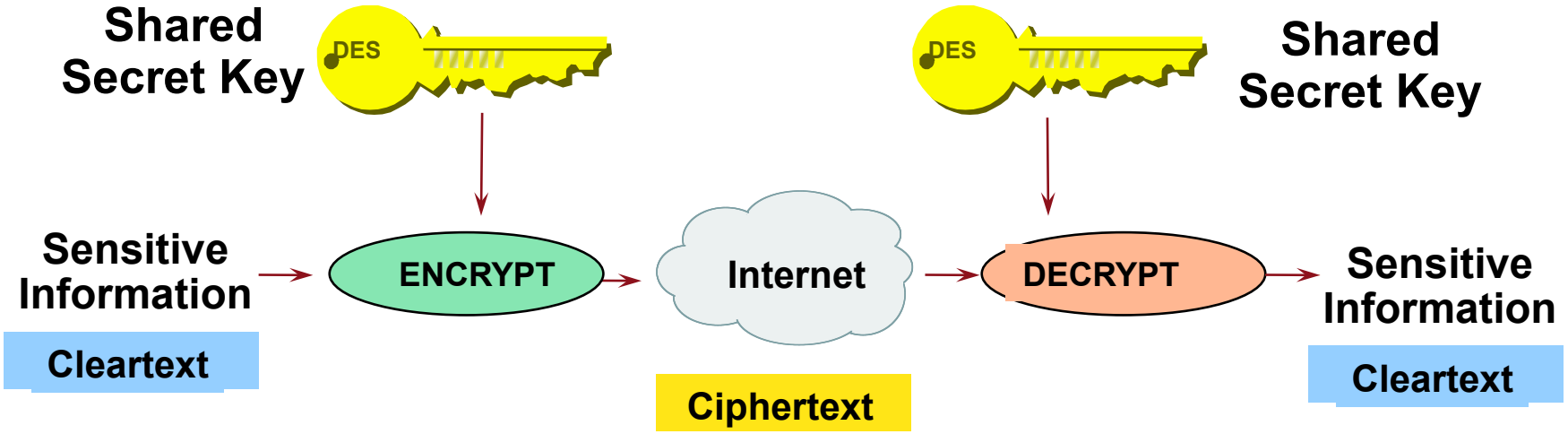


1. Router B generates public/private key pair
2. Router B sends its public key to Router A
3. Router A encrypts packet with router B's public key and sends encrypted packet to Router B
4. Router B receives encrypted packet and decrypts with its' private key

Secret Key Encryption

Uses symmetric keys

- Two parties share the same secret key
- Problem is securely distributing the key



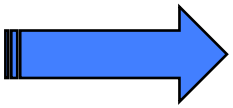
Common Algorithms: DES, 3DES, AES, IDEA

Key Length

Key Length (in bits)	Number of Combinations
40	$2^{40} = 1{,}099{,}511{,}627{,}776$
56	$2^{56} = 7.2 \times 10^{16}$
64	$2^{64} = 1.8 \times 10^{19}$
112	$2^{112} = 5.2 \times 10^{33}$
128	$2^{128} = 3.4 \times 10^{38}$
192	$2^{192} = 6.2 \times 10^{57}$
256	$2^{256} = 1.1 \times 10^{77}$

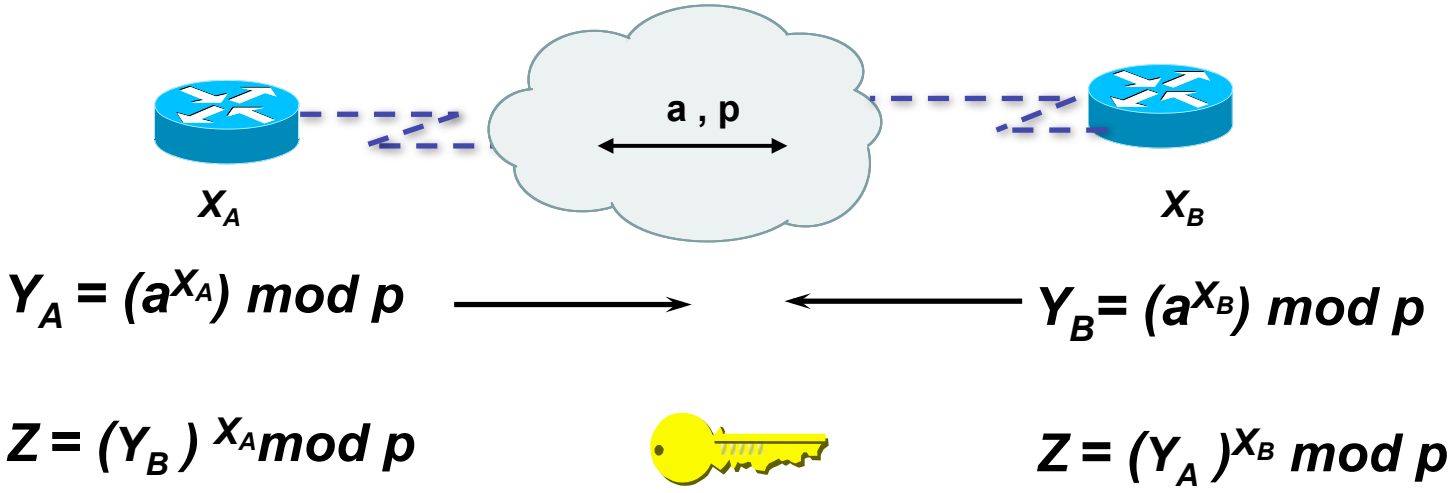
Secret Key Scalability

Configuring shared secret keys easily becomes an administrative nightmare so need automated mechanism to securely derive secret keys in scalable manner



Diffie-Hellman Algorithm

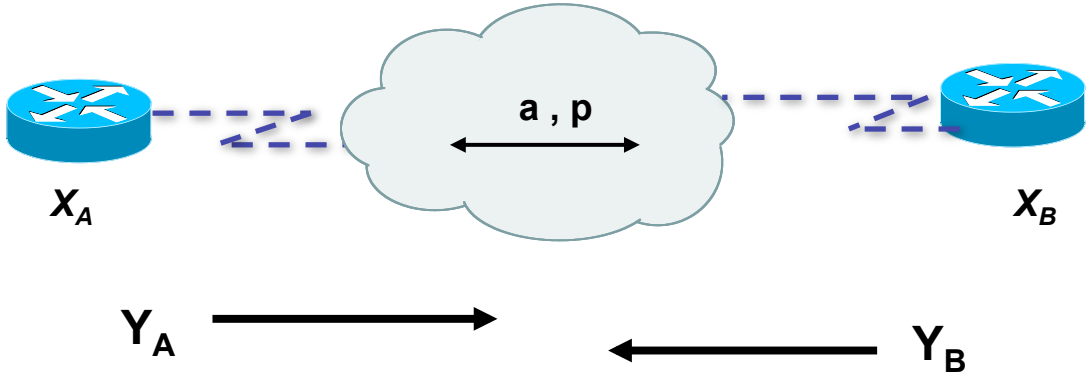
Deriving Secret Keys Using Public Key Technology (e.g. Diffie-Hellman)



By exchanging numbers in the clear, two entities can determine a new unique number (Z), known only to them

DH Man-in-the Middle Attack

- Diffie-Hellman is subject to a man-in-the-middle attack
- Digital signatures of the 'public values' can enable each party to verify that the other party actually generated the value



=> DH exchanges need to be authenticated!!

Hash Algorithm

- Reduces a variable-length input to a fixed-length output
  - Output is called a *hash* or *message digest* or *fingerprint*
  - Output length is 128 bits for MD5 and 160 bits for SHA-1
- Requirements
  - Can't deduce input from output
  - Can't generate a given output
  - Can't find two inputs which produce the same output
- Used to
  - Create data checksum to detect data modification
  - Create fixed-length encryption keys from passwords

X-OR Function

1 xor 1 = 0 0 xor 0 = 0

1 xor 0 = 1 0 xor 1 = 1

Example 1: **0 1 1 0 0 1 0 1 xor'ed with**
1 1 0 1 0 0 1 1

RESULT: **1 0 1 1 0 1 1 0**

Example 2: **1 0 1 1 0 1 1 0 xor'ed with**
1 1 0 1 0 0 1 1

RESULT: **0 1 1 0 0 1 0 1**

Computing a Keyed-MAC

- Message Authentication Code (MAC) creates a hash value dependant on the key (password)
  - Message broken down into n blocks of 512-bits
  - Shared secret key is xor'ed with specified array to produce K1
  - Shared secret key is xor'ed a 2nd time with another specified array to produce K2

Hash1 = $(1^{st}$ block of message + K1$)_{MD5}$

Hash2 = $(hash1 + K2)_{MD5}$

Hash3 = $(2^{nd}$ block of message + hash2$)_{MD5}$

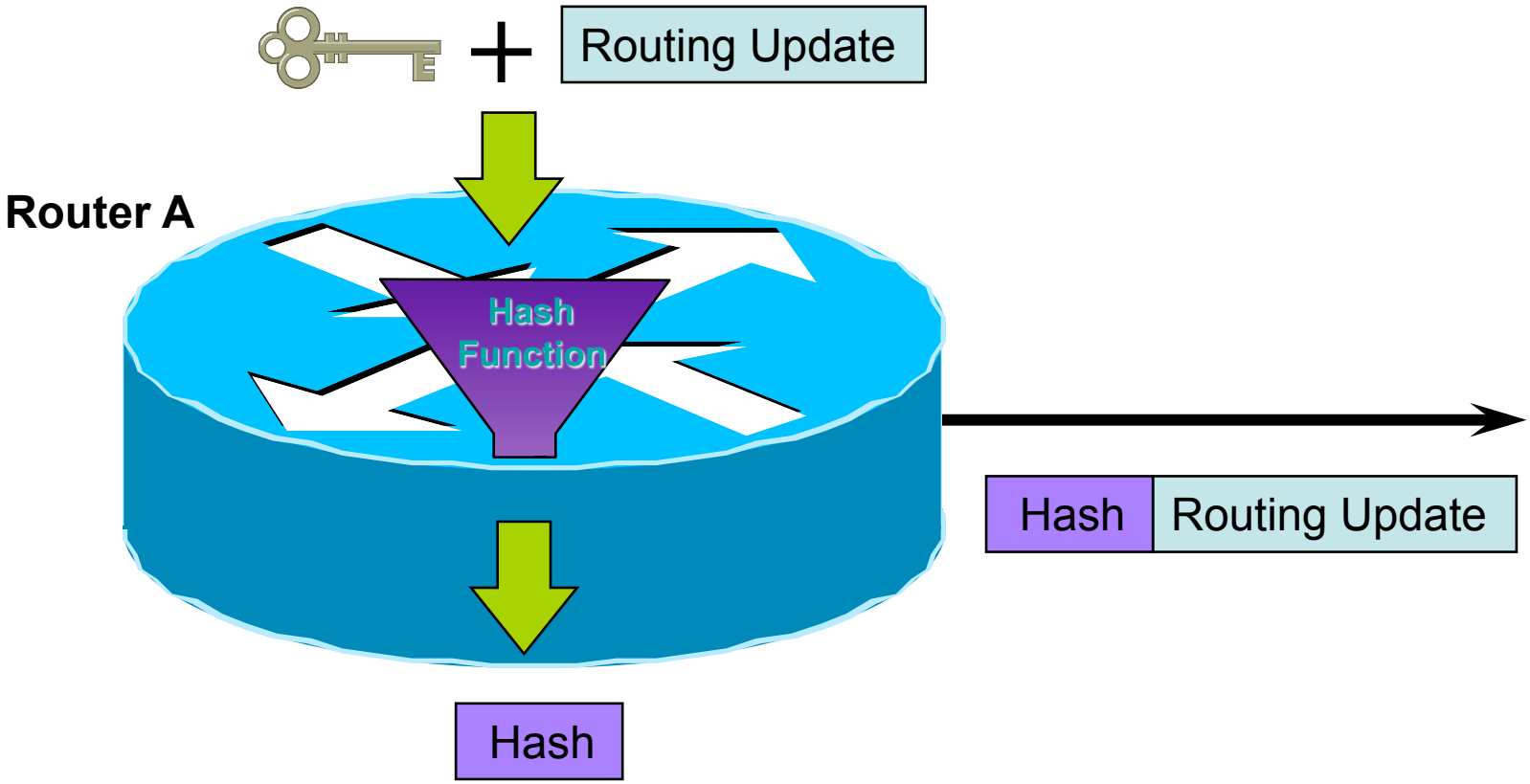
Hash(n+1) = $(n^{th}$ block of message + hashn$)_{MD5}$

HMAC-MD5-96 / HMAC-SHA-96 -> last hash truncated to 96 bits!!

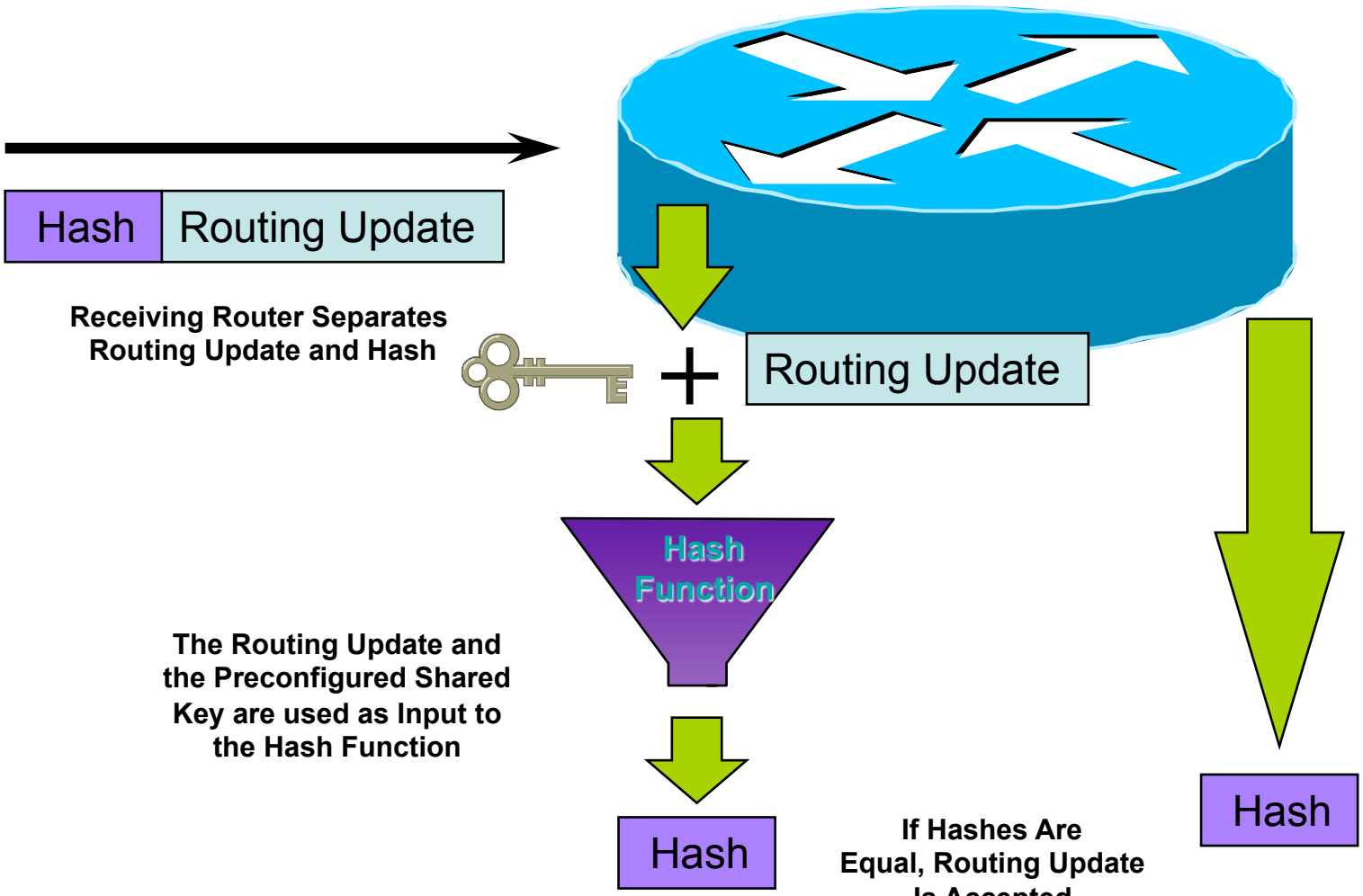
Digital Signatures

- Combines a hash with a digital signature algorithm
- Used to prove the identity of the sender and the integrity of the packet
- Two common public-key digital signature techniques:
  - RSA (Rivest, Shamir, Adelman)
  - DSS (Digital Signature Standard)
- A sender uses its private key to sign a packet.
- The receiver of the packet uses the sender's public key to verify the signature.
- Successful verification assures:
  - The packet has not been altered
  - The identity of the sender

MD-5 Based Authentication



MD-5 Based Authentication



IPsec Technology Details

An Underutilized Technology

- What attacks would not exist if ALL packets were authenticated and integrity protected?
  - IPsec without encryption
  - Requires better products - easier configuration and interoperable defaults
- Can end-users be held more responsible / liable for traffic sourced from their machines?
  - Better auditing capabilities

IPsec Components

- AH (Authentication Header)
  - Authentication is applied to the entire packet, with the mutable fields in the IP header zeroed out
  - If both ESP and AH are applied to a packet, AH follows ESP
  - Standard requires HMAC-MD5-96 and HMAC-SHA1-96….older implementations also support keyed MD5
- ESP (Encapsulating Security Payload)
  - Must encrypt and/or authenticate in each packet
  - Encryption occurs before authentication
  - Authentication is applied to data in the IPsec header as well as the data contained as payload
  - Standard requires DES 56-bit CBC and Triple DES. Can also use RC5, IDEA, Blowfish, CAST, RC4, NULL
- IKE (Internet Key Exchange)
  - Automated SA (Security Association) creation and key management

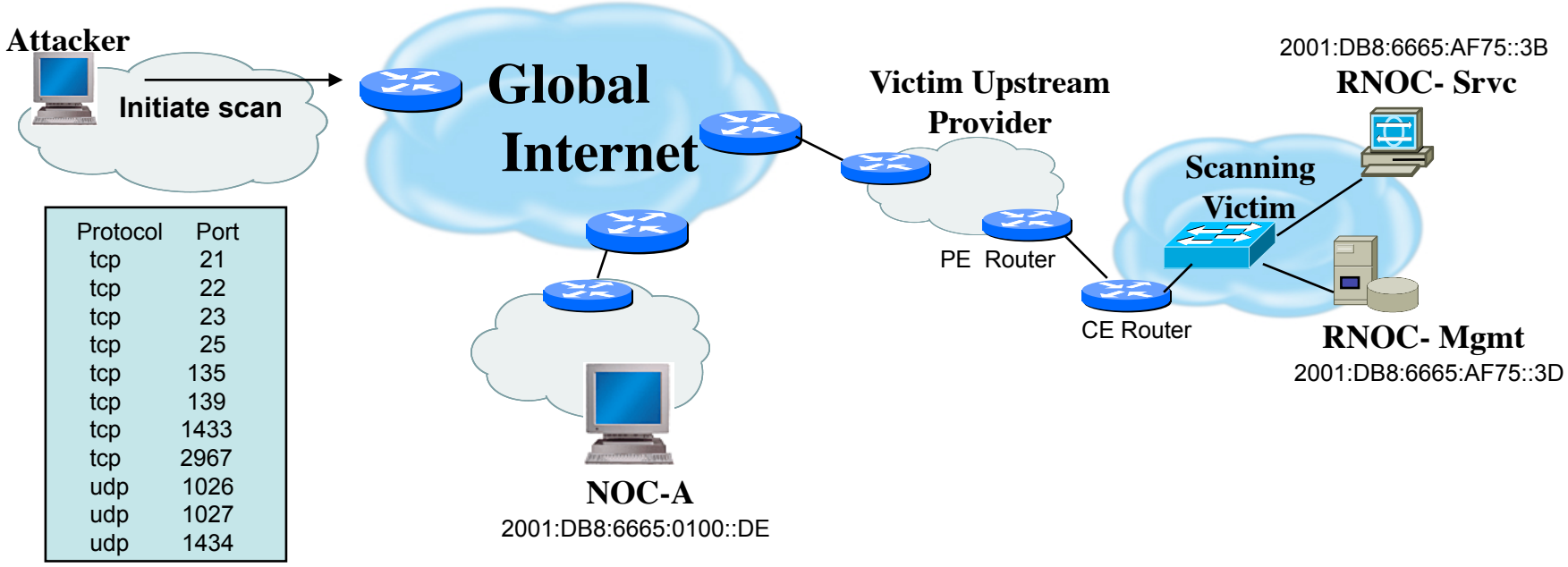
Why Use IPsec ?

- Confidentiality….although not the only reason…….
- Data integrity and source authentication
  - Data “signed” by sender and “signature” verified by the recipient
  - Modification of data can be detected by signature “verification”
  - Because “signature” based on a shared secret, it gives source authentication
    - The shared secret is cryptographically derived
- Anti-replay protection
  - Optional : the sender must provide it but the recipient may ignore
- Key Management
  - IKE – session negotiation and establishment
  - Sessions are rekeyed or deleted automatically
  - Secret keys are securely established and authenticated
  - Remote peer is authenticated through varying options

Considerations For Using IPsec

- Security Services
  - Data origin authentication
  - Data integrity
  - Replay protection
  - Confidentiality
- Size of network
- How trusted are end hosts
- Vendor support
- What other mechanisms can accomplish similar attack risk mitigation

Protecting Against Scanning Attacks



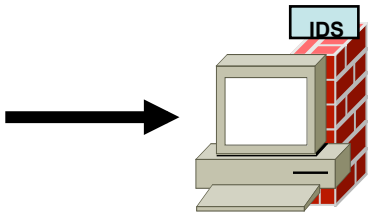
Protocol	Port
tcp	21
tcp	22
tcp	23
tcp	25
tcp	135
tcp	139
tcp	1433
tcp	2967
udp	1026
udp	1027
udp	1434

IPsec Security Policy Database

From	To	Protocol	Dst Port	Policy
2001:DB8:6665:0100::DE	2001:DB8:6665:01C8::3B	TCP / UDP	53 (DNS)	ESP: SHA1, AES-256
2001:DB8:6665:0100::DE	2001:DB8:6665:AF75::3B	TCP	25 (SNMP)	ESP: SHA1, AES-256
2001:DB8:6665:0100::DE	2001:DB8:6665:AF75::3D	UDP	1812/1813 (RADIUS)	ESP: SHA1, AES-128
2001:DB8:6665:0100::DE	2001:DB8:6665:AF75::3D	UDP	514 (Syslog)	ESP: SHA1, 3DES
2001:DB8:6665:0100::DE	2001:DB8:6665:AF75::/48	TCP / UDP	ANY	ESP: SHA1

DOUBLE SHOT SECURITY

Inbound IPsec processing



Device Receives IPsec Protected Packet

Consult SAD Using <SPI, DST IP, AH/ESP>

SA Match

Process The IPsec Packet

No SA Match

Do Traffic Selectors Adhered To SPD?

NO

Drop Packet & Log Error Message

YES

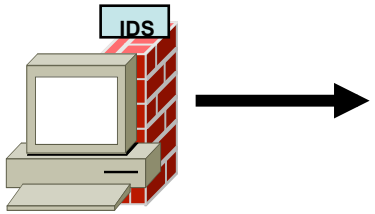
Forward Packet to Upper Layer Processing

Traffic Selectors classify the traffic to be protected and can be a combination of:

- Source Address
- Destination IP Address
- Protocol Type
- Upper Layer Port Numbers
- User ID*
- Application Type*

* Only on host-based transport SAs

Outbound IPsec processing



Application Generates Packet

Consult SPD Outbound Cache Using Traffic Selectors

DROP

FORWARD **Unprotected**

Data Link Layer Processing

PROTECT

Is SA Established In SAD?

YES

FORWARD **IPsec Protected**

Apply IPsec Protection

NO

Create SA (Manually or Automatically)

Populate SAD

Traffic Selectors classify the traffic to be protected and can be a combination of:

- Source Address
- Destination IP Address
- Protocol Type
- Upper Layer Port Numbers
- User ID*
- Application Type*

* Only on host-based transport SAs

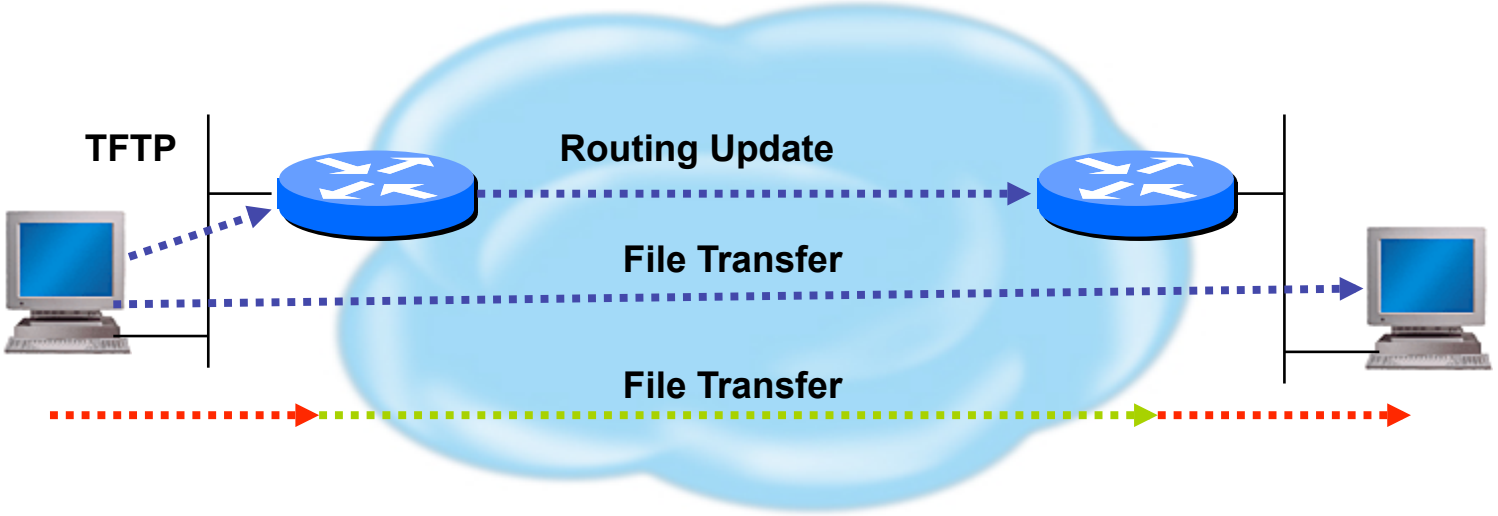
Architectures using IPsec

- Protect all traffic using IPsec for data origin authentication and integrity
  - ***AH versus ESP/Null Encryption***

- Add confidentiality as dictated by security policy
  - ***ESP***

Need to dispel myth that using IPsec mandates the demise of network layer defense mechanisms

Transport vs Tunnel Mode

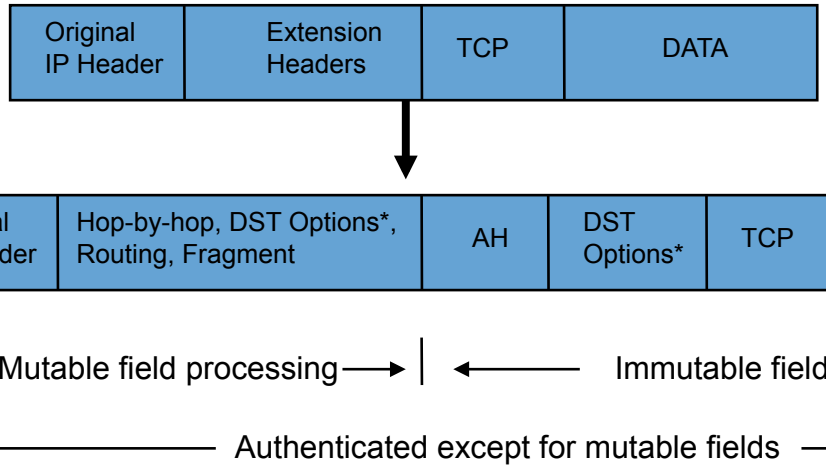


Transport Mode: End systems are the initiator and recipient of protected traffic

Tunnel Mode: Gateways act on behalf of hosts to protect traffic

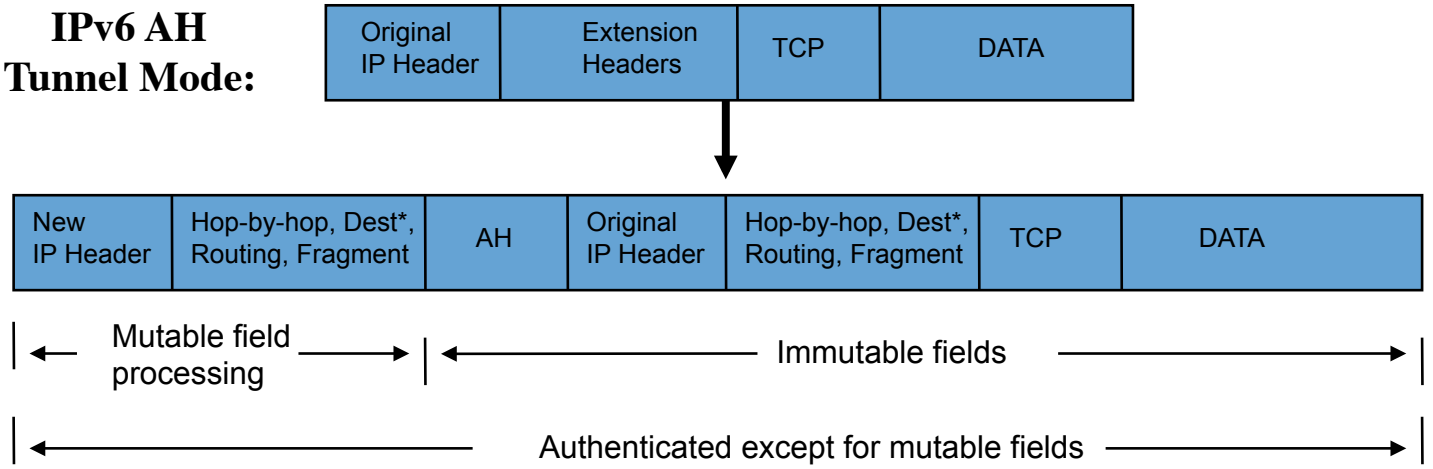
IPv6 IPsec AH

IPv6 AH Transport Mode:



Mutable Fields:
- DSCP
- ECN
- Flow Label
- Hop Limit

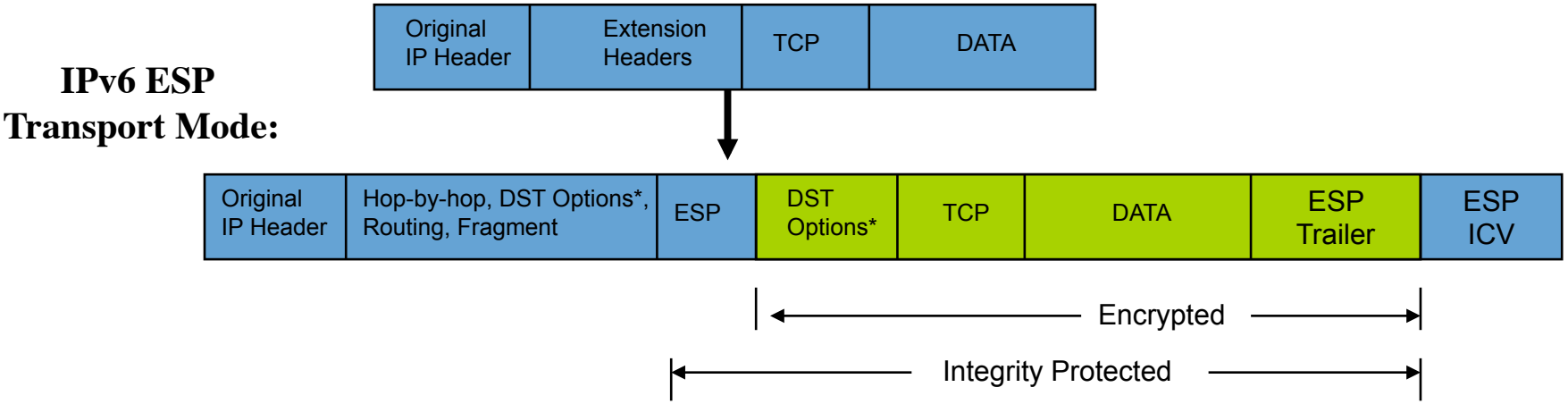
IPv6 AH Tunnel Mode:



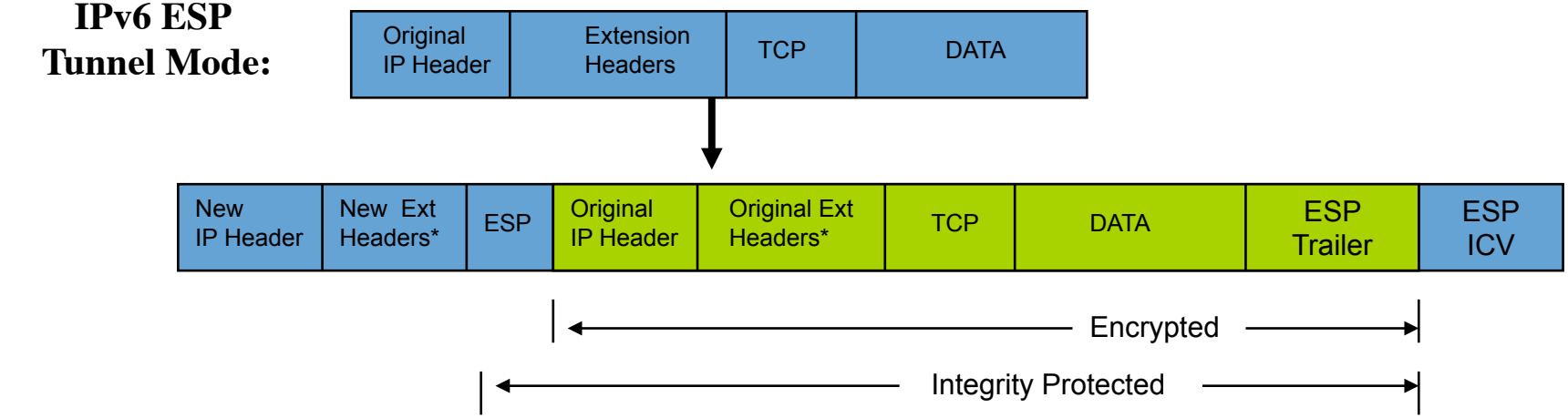
Mutable Fields:
- DSCP
- ECN
- Flow Label
- Hop Limit

IPv6 IPsec ESP

IPv6 ESP Transport Mode:

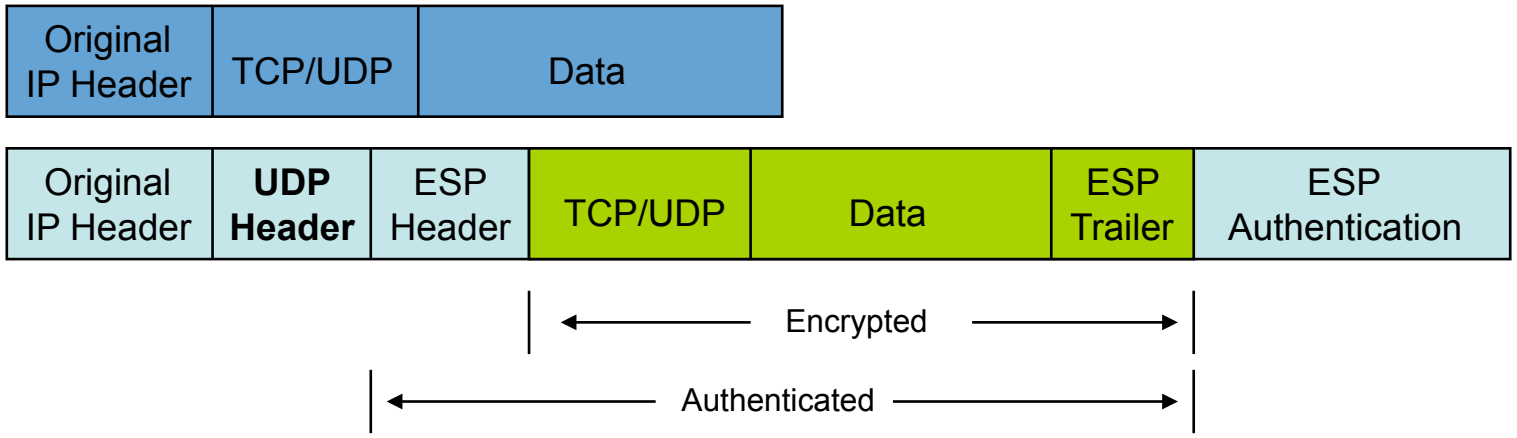


IPv6 ESP Tunnel Mode:

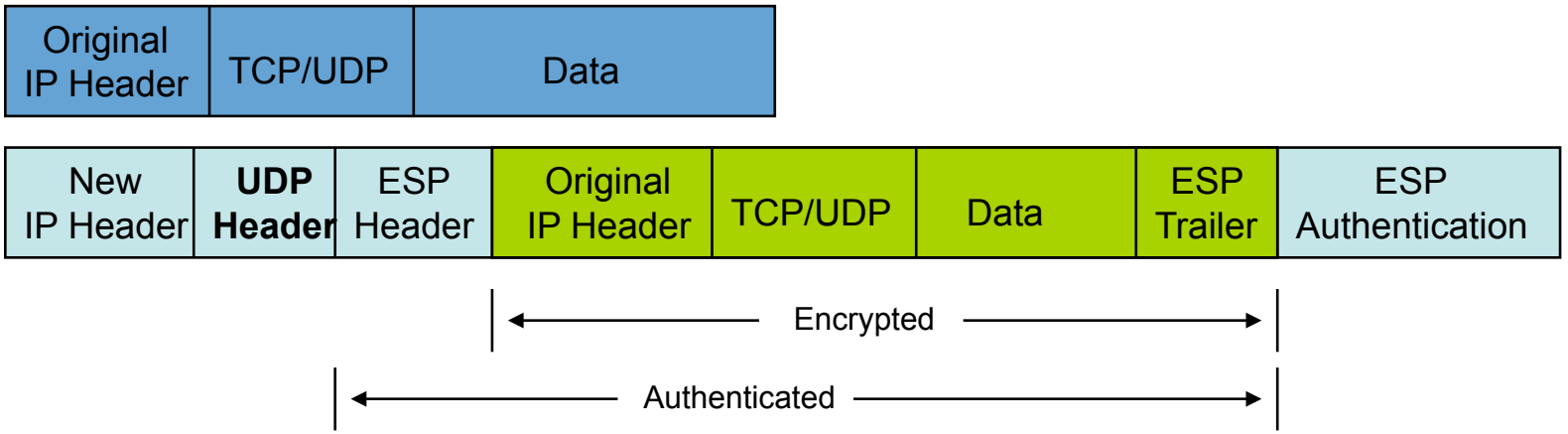


NAT Traversal [Requires IPsec ESP]

Transport Mode



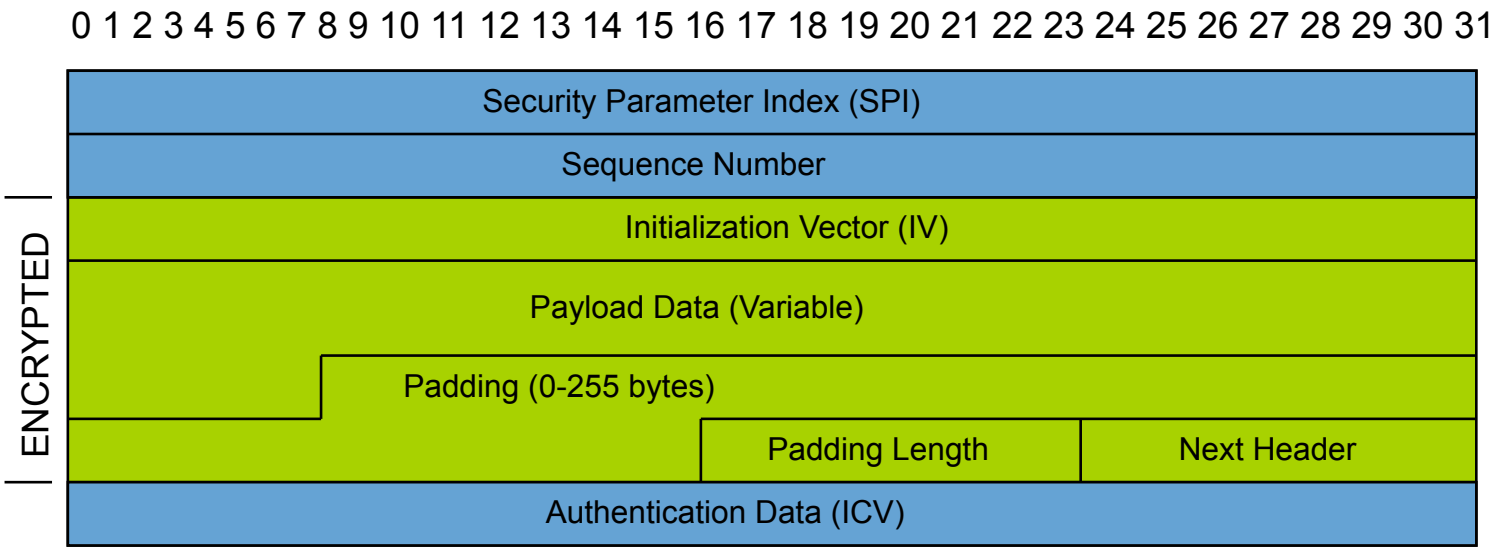
Tunnel Mode



IPv6 Enhancements Needed

- Standards Modifications
  - Need to take into consideration Stateless Autoconfiguration where Router Advertisement sends network prefix
  - Need to be able to differentiate between encrypted versus integrity protected traffic
- Usability
  - Interoperable defaults
  - Consistent terminology

ESP Header Format



SPI: Arbitrary 32-bit number that specifies SA to the receiving device
Seq #: Start at 1 and must never repeat; receiver may choose to ignore
IV: Used to initialize CBC mode of an encryption algorithm
Payload Data: Encrypted IP header, TCP or UDP header and data
Padding: Used for encryption algorithms which operate in CBC mode
Padding Length: Number of bytes added to the data stream (may be 0)
Next Header: The type of protocol from the original header which appears in the encrypted part of the packet
Auth Data: ICV is a digital signature over the packet and it varies in length depending on the algorithm used (SHA-1, MD5)

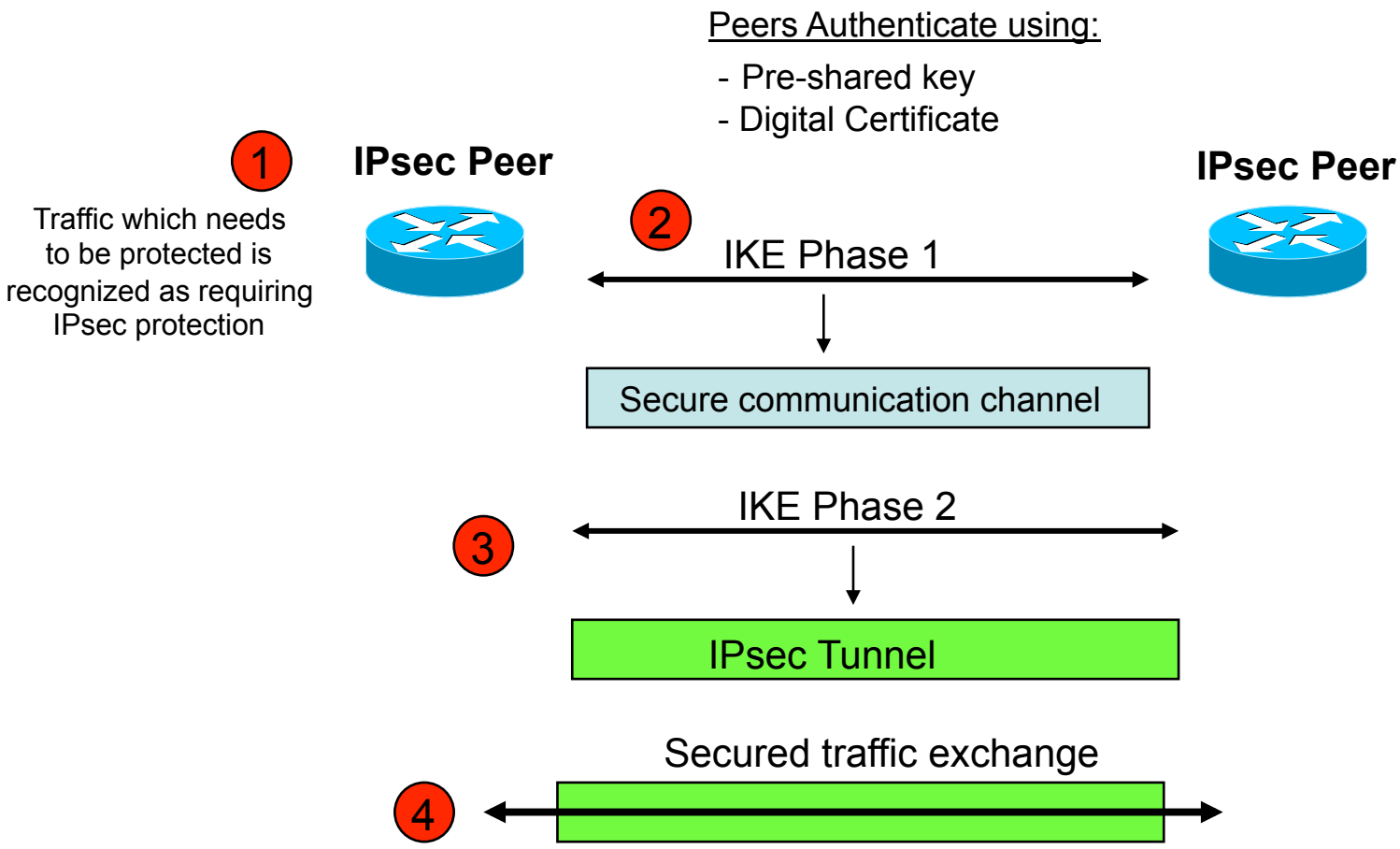
IKE - Internet Key Exchange

- Automatically establishes SAs and creates/deletes cryptographic material
- Authenticates communicating peers
- IKE1 Works in 2 Phases
  - Phase I
    - Establish a secure channel (ISAKMP/IKE SA) to negotiate the data protection cryptographic material
    - Results in a single ISAKMP/IKE SA
  - Phase II
    - Establishes the secure channel for the transmission of data
    - Results in a pair of IPsec SAs

IKEv1

- Phase I
  - Establish a secure communication channel (ISAKMP/IKE SA)
  - Main Mode
    - Negotiates an ISAKMP SA which will be used to create IPsec SAs
    - SA negotiation (encryption algorithm, hash algorithm, authentication method, which DF group to use)
    - Do a Diffie-Hellman exchange
    - Provide authentication information
    - Authenticate the peer
  - Aggressive Mode
    - Uses 3 (vs 6) messages to establish IKE SA
    - No denial of service protection
    - Does not have identity protection
    - Optional exchange and not widely implemented
- Phase II
  - Establishes a secure channel for actual data (IPsec SA)
  - Quick mode
    - All traffic is encrypted using the ISAKMP/IKE Security Association
    - Each quick mode negotiation results in two IPsec Security Associations
    - Creates/refreshes keys

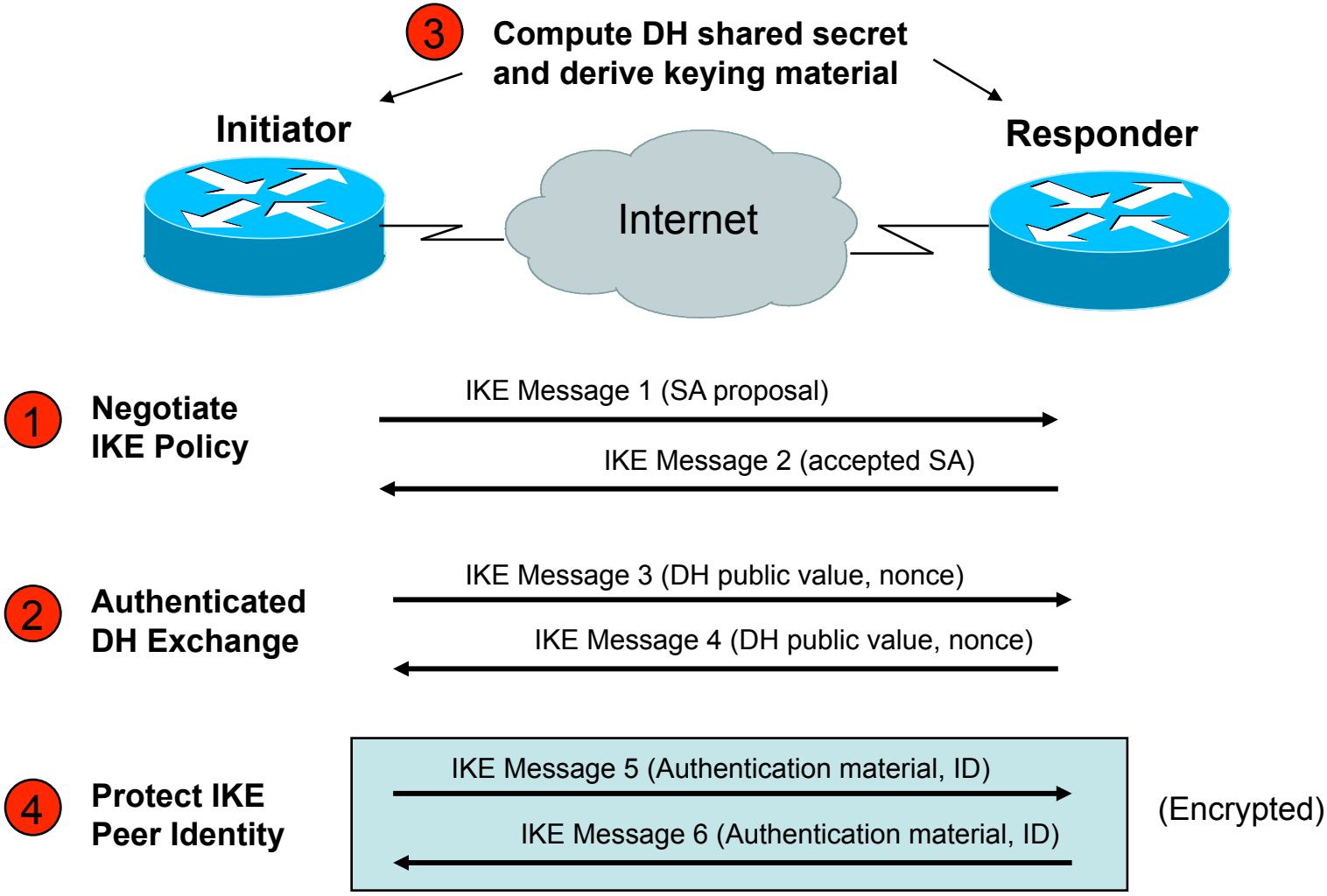
IPsec with IKE



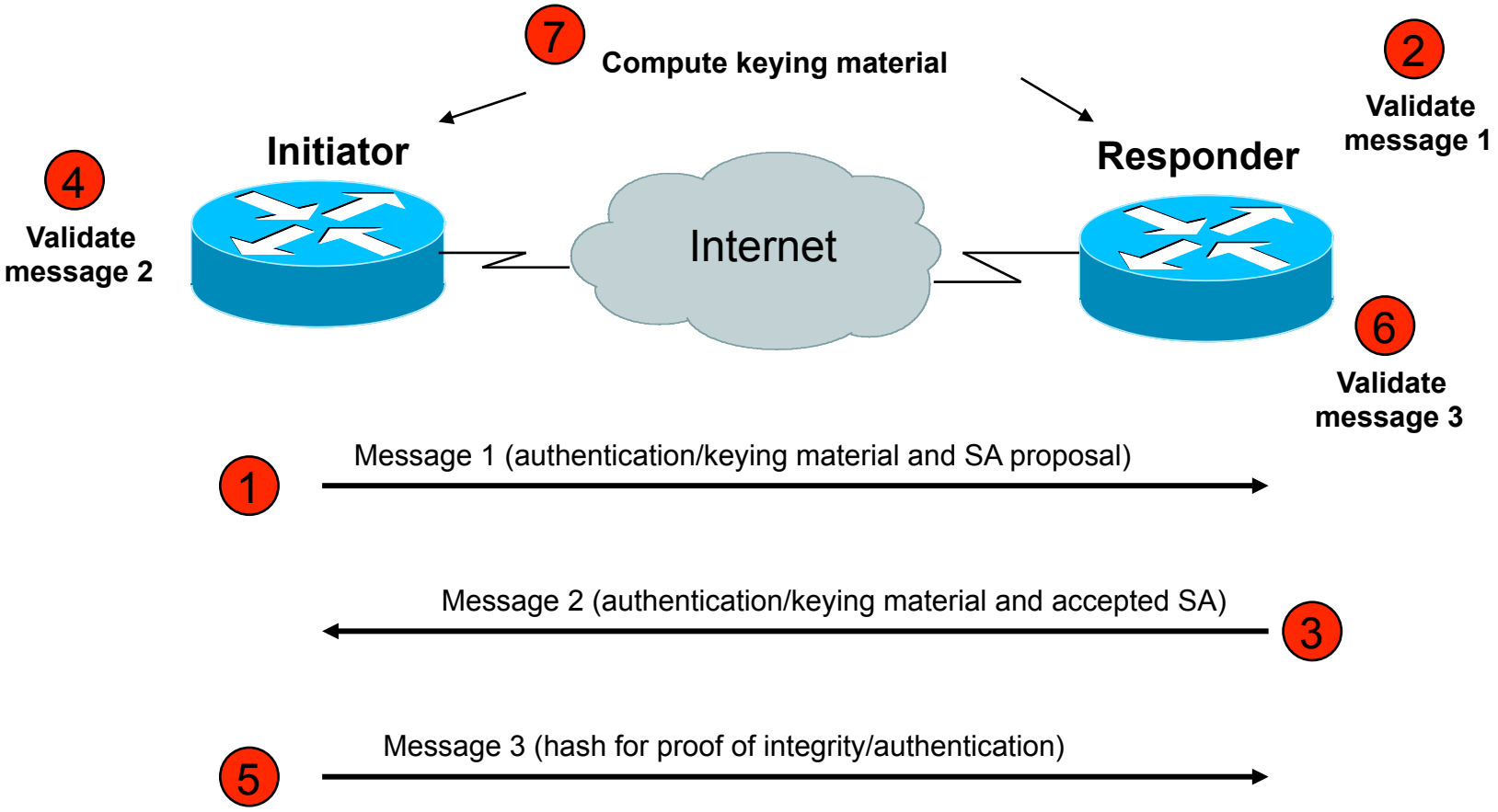
IPsec IKE Phase 1 Uses DH Exchange

- First public key algorithm (1976)
- Diffie Hellman is a key establishment algorithm
  - Two parties in a DF exchange can generate a shared secret
  - There can even be N-party DF changes where N peers can all establish the same secret key
- Diffie Hellman can be done over an insecure channel
- IKE authenticates a Diffie-Hellman exchange
  - Pre-shared secret
  - Nonce (RSA signature)
  - Digital signature

IKE Phase 1 Main Mode



IKE Phase 2 Quick Mode



PFS- what is it?

- Perfect Forward Secrecy
- Doing new DH exchange to derive keying material (instead of using the shared key derived from previous phase 1 and not doing another phase 1)
- Only relevant for re-keying a phase 2 IKE (I.e. automatically establishing new keys for integrity and confidentiality of the traffic you want to protect)

(DH used to derive shared secret which is used to derive keying material for IPsec security services)

Vendor Specific Deployment Issues

- Lack of interoperable defaults
  - A default does NOT mandate a specific security policy
  - Defaults can be modified by end users
- Configuration complexity
  - Too many knobs
  - Vendor-specific terminology
- Good News: IPv6 support in most current implementations

Default Issues

Vendor A

IKE Phase 1
- -SHA1
- -RSA-SIG
- -Group 1
- -Lifetime 86400 Sec
- -Main Mode

IKE Phase 2
- -PFS
- -Group 1

Vendor B

IKE Phase 1
- -MD5
- -Pre-Share Key
- -Group 5
- -Lifetime 86400 Sec
- -Main Mode

IKE Phase 2
- -PFS
- -Group 5

Vendor C

IKE Phase 1
- -SHA1
- -Pre-Share Key
- -Group 2
- -Lifetime 86400 Sec
- -Aggressive Mode

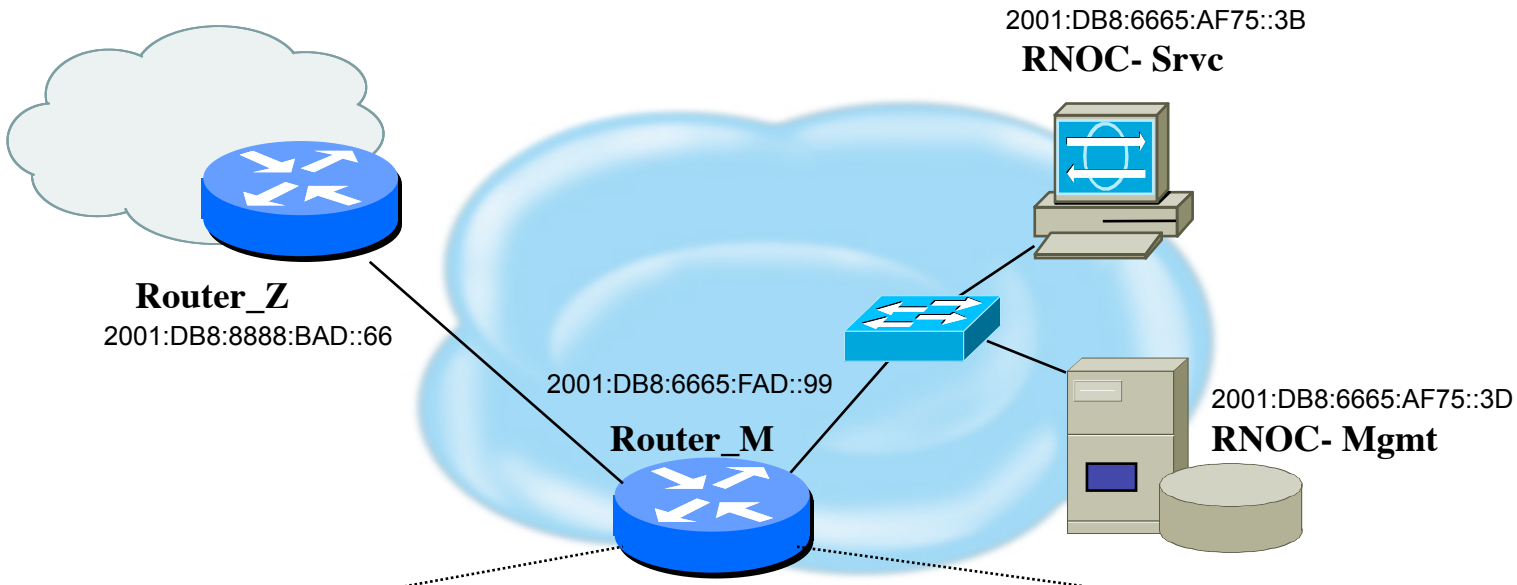
IKE Phase 2
- -PFS
- -Group 2

Terminology Issues

IKE Phase 1	DH Key Length	IKE Phase 2
IKE Phase 1 SA	DH Group	IKE Phase 2 SA
IKE SA	Modp #	IPsec SA
ISAKMP SA	Group #	Quick Mode
Main Mode		

Configuration complexity increased with vendor -specific configuration terms

Potentially Easy Configuration



Syslog server 2001:DB8:6665:AF75::3D authenticate esp-null sha1 pre-share ‘secret4syslog’

TFTP server 2001:DB8:6665:AF75::3D authenticate esp-null aes128 pre-share ‘secret4tftp’

BGP peer 2001:DB8:8888:BAD::66 authenticate esp-null aes128 pre-share ‘secret4AS#XXX’

DOUBLE SHOT SECURITY

Interoperable Defaults For SAs

- Security Association groups elements of a conversation together
  - AH authentication algorithm and keys
  - ESP encryption algorithm and key(s)
  - Cryptographic synchronization
  - SA lifetime
  - SA source address
  - Mode (transport or tunnel)

How Do We Communicate Securely ?

Do we want integrity protection of data ?
Do we want to keep data confidential ?
Which algorithms do we use ?
What are the key lengths ?
When do we want to create new keys ?
Are we providing security end-to-end ?

Pretty Good IPsec Policy

- IKE Phase 1 (aka ISAKMP SA or IKE SA or Main Mode)
  - 3DES (AES-192 if both ends support it)
  - Lifetime (480 min = 28800 sec)
  - SHA-1
  - DH Group 14 (aka MODP# 14)
- IKE Phase 2 (aka IPsec SA or Quick Mode)
  - 3DES (AES-192 if both ends support it)
  - Lifetime (60 min = 3600 sec)
  - SHA-1
  - PFS 2
  - DH Group 14 (aka MODP# 14)

Routers: Configuring IPsec

- For IPv6, consider using transport mode between routers and syslog servers, tftp servers, snmp servers, etc.
- Document for Cisco IPv6 IPsec configuration:
  - http://www.lseltd.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6_c/v6_ipsec.pdf
- Document for Juniper IPsec configuration:
  - http://www.pacificbroadband.com/techpubs/software/junos/junos83/feature-guide-83/html/fg-ipsec13.html#1139838

Cisco: Configuring IPsec

STEP 1 *Configure the IKE Phase 1 Policy (ISAKMP Policy)*

Cisco literature refers to IKE Phase 1 as the ISAKMP policy.
It is configured using the command:

crypto isakmp policy *priority*

Multiple policies can be configured and the priority number, which ranges from 1 to 10,000, denotes the order of preference that a given policy will be negotiated with an ISAKMP peer. The lower value has the higher priority. Once in the ISAKMP configuration mode, the following parameters can be specified are:

Encryption Algorithm
Hash Algorithm
Authentication Method
Group Lifetime

Cisco: Configuring IPsec

STEP 2 *Set the ISAKMP Identity*

The ISAKMP identity specifies how the IKE Phase 1 peer is identified, which can be either by IP address or host name.
The command to use is:

crypto isakmp *identity* {*IP address* | *hostname*}

By default, a peer's ISAKMP identity is the peer's IP address.
If you decide to change the default just keep in mind that it is best to always be consistent across your entire IPsec-protected network in the way you choose to define a peer's identity.

Cisco: Configuring IPsec

STEP 3 *Configure the IPsec AH and ESP Parameters*

The AH and ESP parameters are configured with the following commands:

crypto ipsec transform-set *transform-set-name* <transform 1> <transform 2> mode [tunnel | transport]
crypto ipsec security-association lifetime seconds *seconds*

STEP 4 *Configure the IPsec Traffic Selectors*

The traffic selectors are configured by defining extended access-lists. The *permit* keyword causes all IP traffic that matches the specified conditions to be protected by IPsec

Cisco: Configuring IPsec

STEP 5 *Configure the IKE Phase 2 (IPsec SA) Policy*

This step sets up a crypto map which specifies all the necessary parameters to negotiate the IPsec SA policy. The following commands are required:

crypto map *crypto-map-name seq-num* ipsec-isakmp
match address *access-list-id*
set peer [*IP address* | *hostname*]
set transform-set *transform-set-name*
set security-association lifetime seconds *seconds*
set pfs [group1 | group 2]

Cisco: Configuring IPsec

STEP 6 *Apply the IPsec Policy to an Interface*

The configured crypto map is then applied to the appropriate interface using the crypto map *crypto-map-name* command.
It is possible to apply the same crypto map to multiple interfaces.
This case would require the use of the command:

crypto map *crypto-map-name* local-address *interface-id*

Using this command, the identifying interface will be used as the local address for IPsec traffic originating from or destined to those interfaces sharing the same crypto map. A loopback interface should be used as the identifying interface.

Unix IPsec IKE Daemons

- Racoon2 (IKEv1 and IKEv2 and KINK)
  - http://www.racoon2.wide.ad.jp/w/
- Ipsec-tools (IKEv1)
  - port of KAME's IPsec utilities to the Linux-2.6 IPsec implementation; it supports NetBSD and FreeBSD as well
  - http://ipsec-tools.sourceforge.net/
- Strongswan (IKEv1 and IKEv2)
  - http://www.strongswan.org/
- Openikev2 (IKEv2)
  - http://openikev2.sourceforge.net/

LINUX and MACOSX machines

- Type command ‘ *man racoon* ’
  - Read how to set-up racoon, the name for this particular IKE software
- Type command ‘ *man setkey* ’
  - This command is used to set up the SA database
- The following files are located in ***/etc/raccoon***:
  - ***psk.txt*** – file which contains the shared secrets
  - ***raccoon.conf*** – file which configures IKE phase 1 and IKE phase 2 parameters

Set Up Security Policy Database

- Create a file named '***ipsec.conf***' which will be used with *setkey* to establish the correct security associations. The file should have the following information:

  - `flush;`
  - `spdflush;`
  - `spdadd 2001:DB8:6665:AF75::3D/128 2001:DB8:8888:BAD::66/128 any -P out ipsec esp/ transport//require ;`
  - `spdadd 2001:DB8:8888:BAD::66/128 2001:DB8:6665:AF75::3D/128 any -P in ipsec esp/ transport//require ;`

Creating SA Database

- Test to see what happens when you try and create an SA database:
- Type the following:
  - `setkey -f /etc/racoon/ipsec.conf`
- Use the '`setkey -P -D`' command to see if appropriate entries have been created

Pre-Shared Key Configuration

- Edit the psk.txt file to add the peer IP address and the pre-shared secret key:
  - `# file for pre-shared keys used for IKE authentication`
  - `# format is:  'identifier' 'key'`
  - `# For example:`
  - `#  10.1.1.1              flibbertigibbet`
  - `#  www.example.com       12345`
  - `#  foo@www.example.com   micropachycephalosaurus`
  - `<peer IPv6 address>      <shared secret>`
- Since the psk.txt file contains sensitive information make sure that the file is appropriately protected:
  - `chmod 600 /etc/raccoon/psk.txt`

Racoon.conf file

```
# Racoon IKE daemon configuration file.
# See 'man racoon.conf' for a description of the format
      and entries.
path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";
log debug;
remote anonymous
{
    {
        exchange_mode main;
        lifetime time 480 min;
        proposal {
            encryption_algorithm 3des;
            hash_algorithm sha1;
            authentication_method pre_shared_key;
            dh_group 14;
        }
    }
```

```
sainfo anonymous
{
        pfs_group 2;
        lifetime time 60 min ;
        encryption_algorithm 3des, blowfish
         448, rijndael ;
        authentication_algorithm hmac_sha1,
         hmac_md5 ;
        compression_algorithm deflate ;
}
```

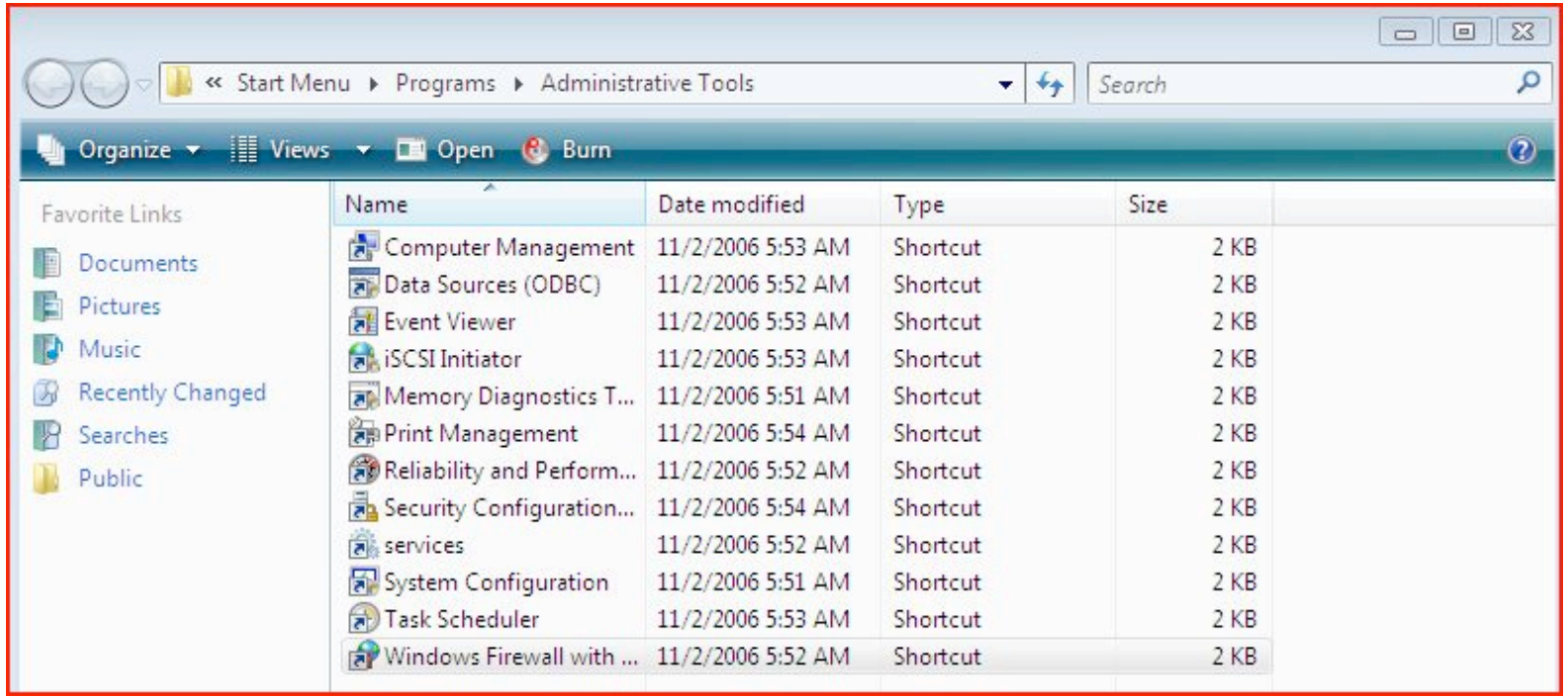
Testing Racoon

- Test racoon with the following command:
  - `racoon -v -f /etc/racoon/racoon.conf -l /etc/racoon/test.log`

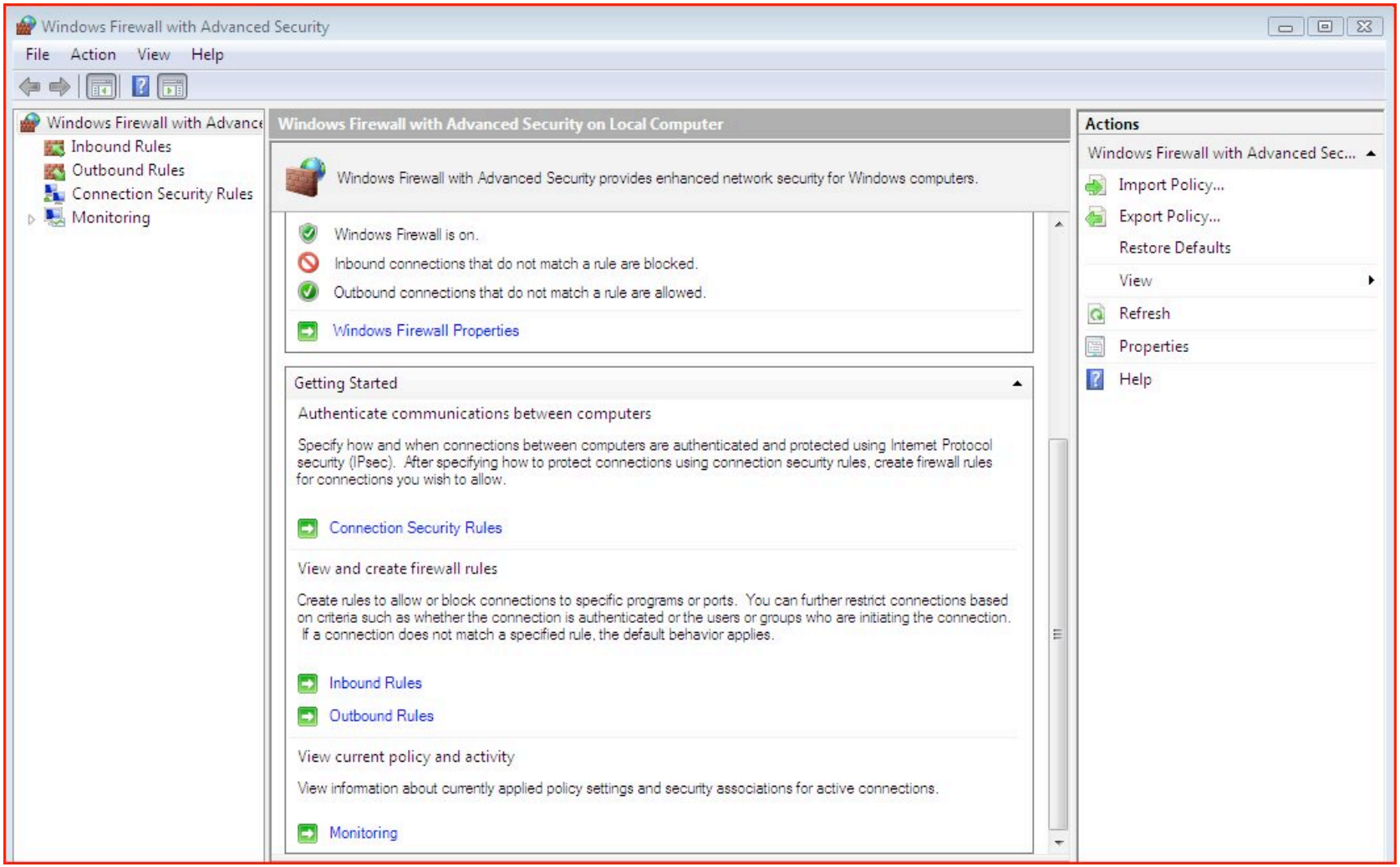
- The '`-l /etc/racoon/test.log`' file is used to write any debug information in the event that there are problems.

Vista: Configuring IPsec

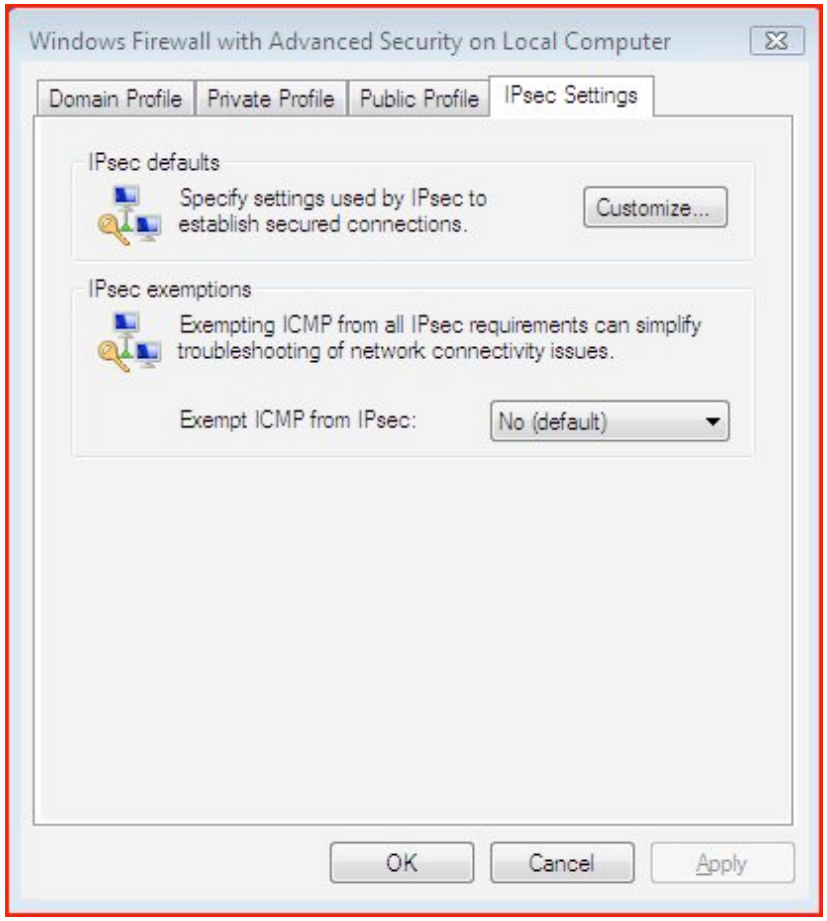
- Defaults work great in a MS-only environment
- Need to edit firewall (wf.mmc) otherwise

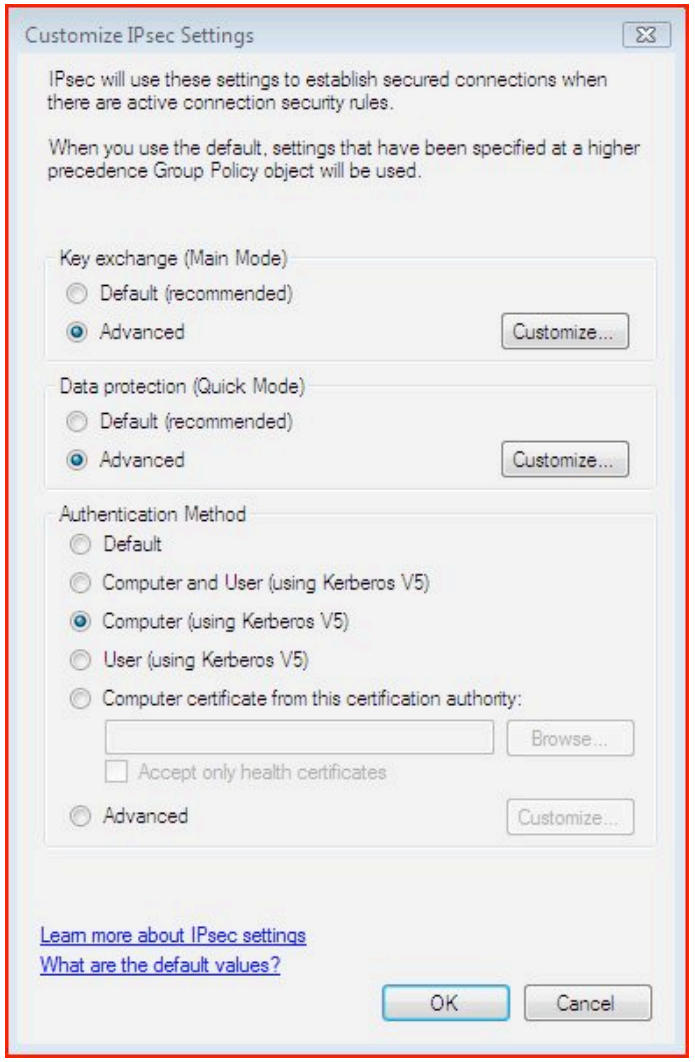


Vista: Configuring IPsec

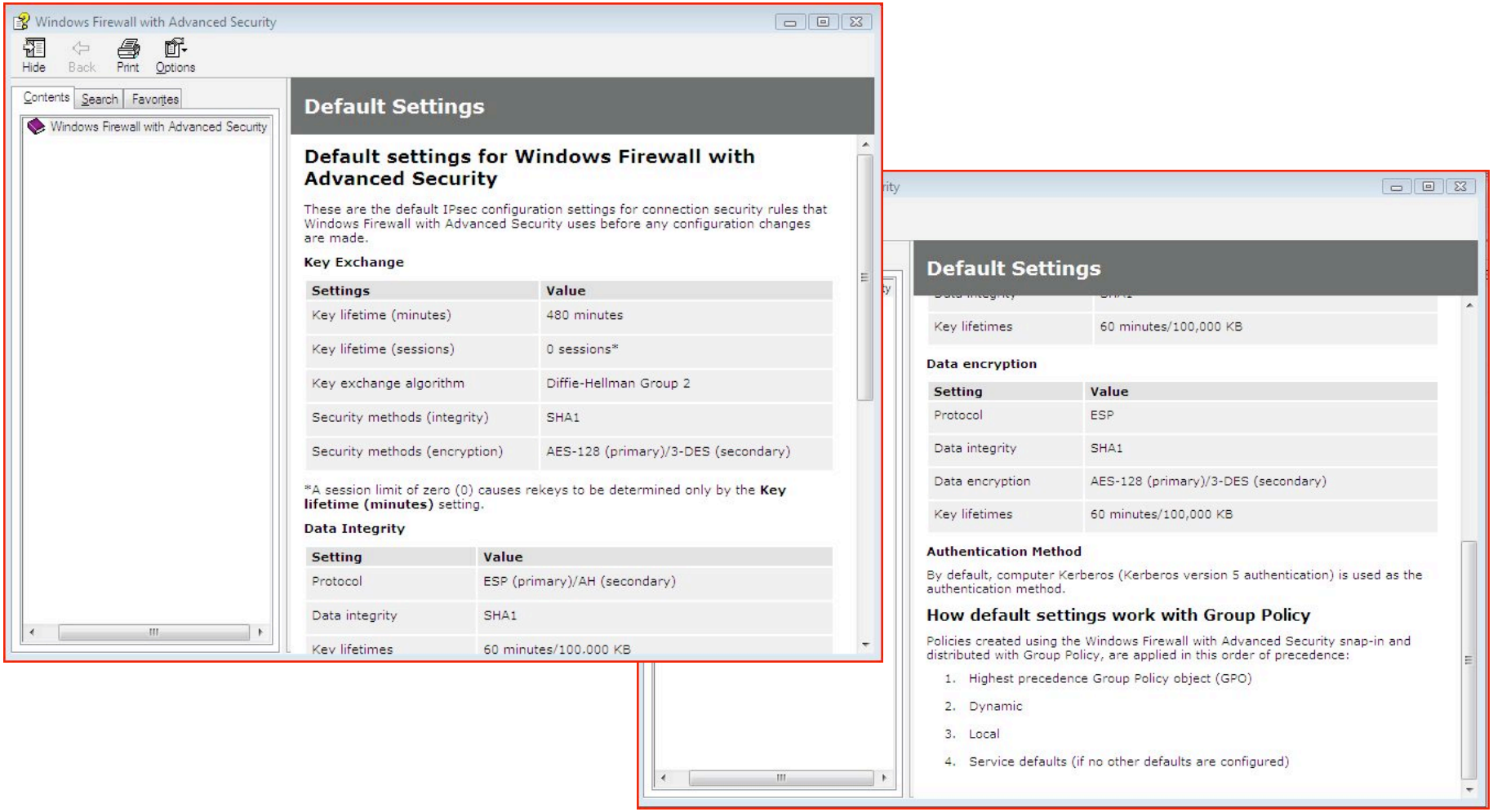


Vista: Customizing IPsec Settings

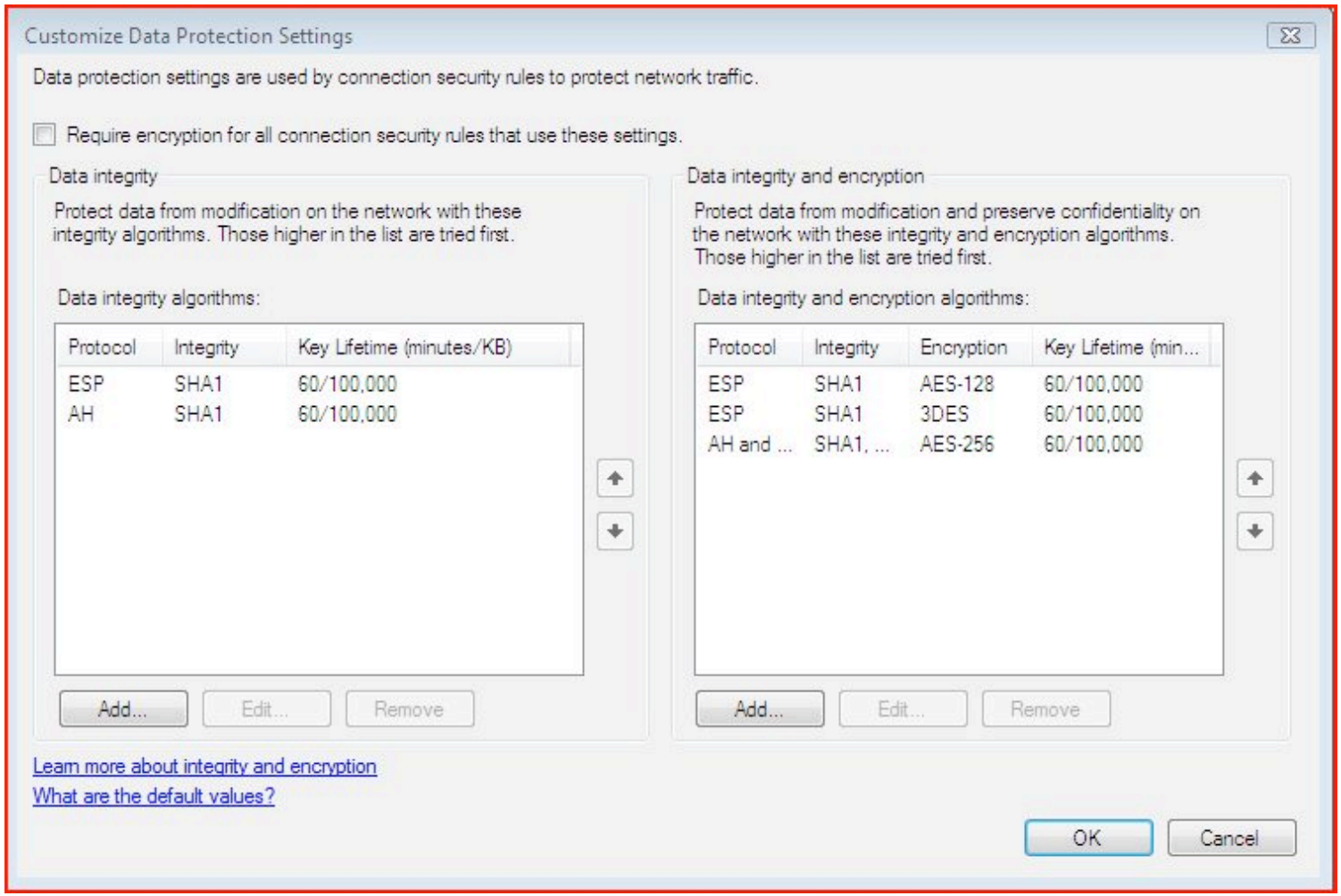




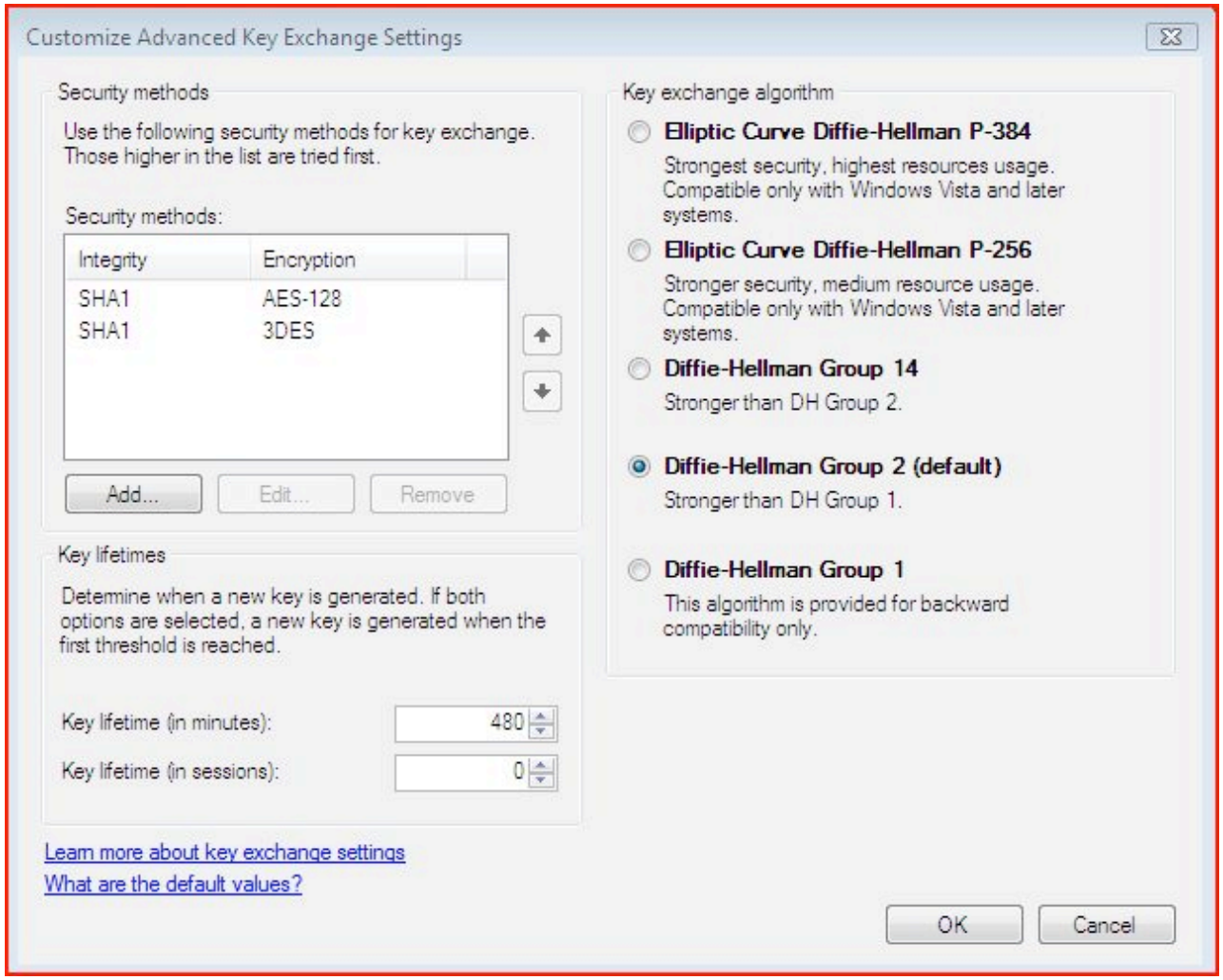
Vista IPsec Defaults



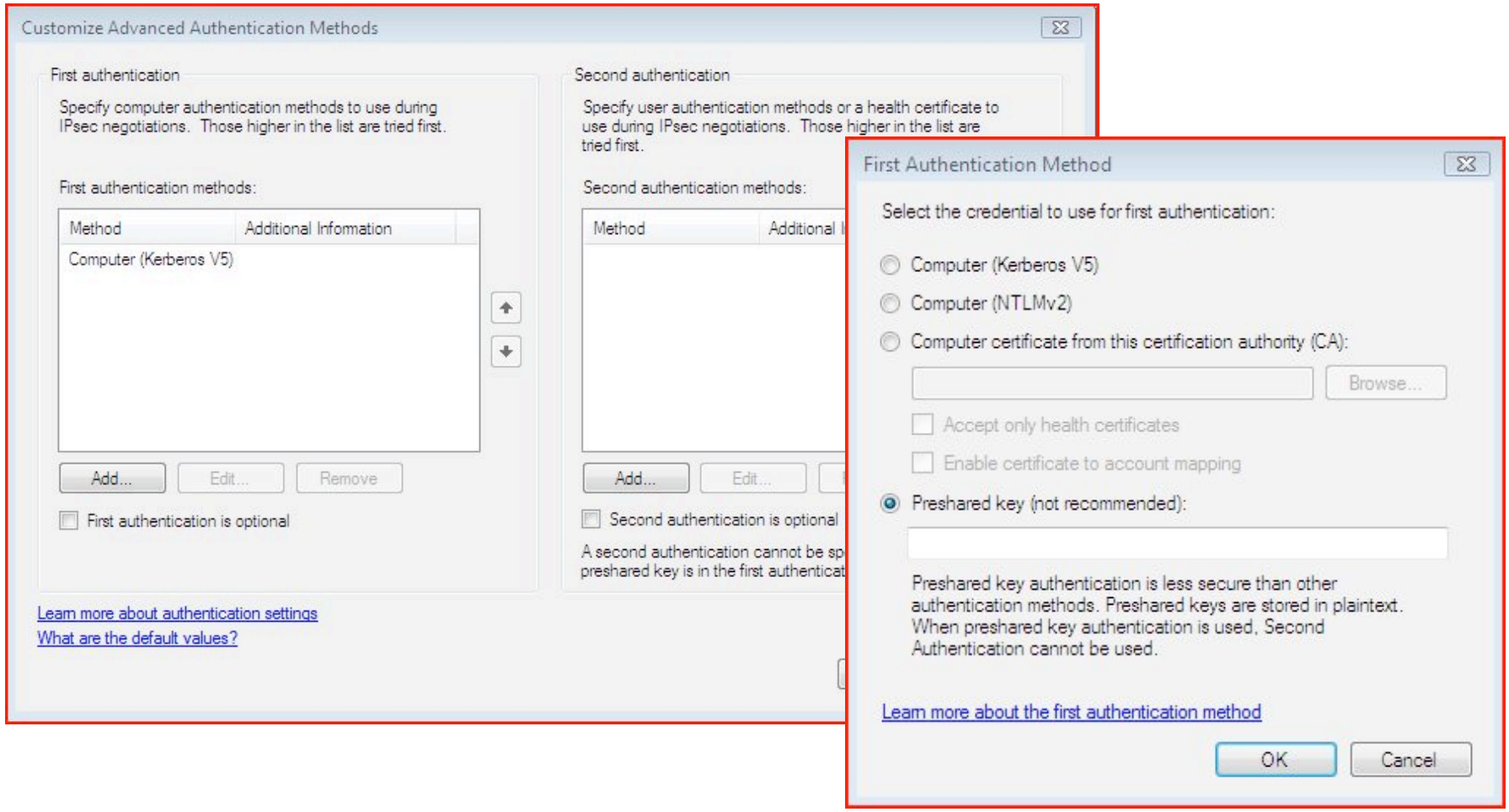
Vista: Customizing Data Protection



Vista: Customizing Key Exchange



Vista: Customizing Authentication



Conclusions

- IPsec is a complex standard but user configurations shouldn't be
- Using IPsec does NOT mean you have to encrypt the data (providing traffic integrity can be useful too)
- Don't leave IPsec out when you are trying to gain experience with IPv6 - time to fix usability issues is NOW