

Network Infrastructure Security

APRICOT 2005 Workshop

February 18-20, 2005

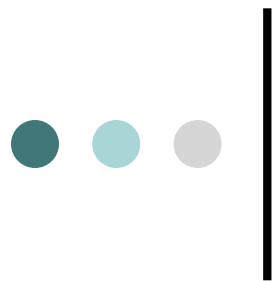
Merike Kaeo

merike@doubleshotsecurity.com



Agenda (Day 2)

- Securing Data Traffic
 - Packet Filters
 - Encryption (IPsec vs SSL)
- Logging Information
 - What to Log
 - Storing Logs
- LAB
 - Ingress / Egress Filtering
 - IPsec configurations

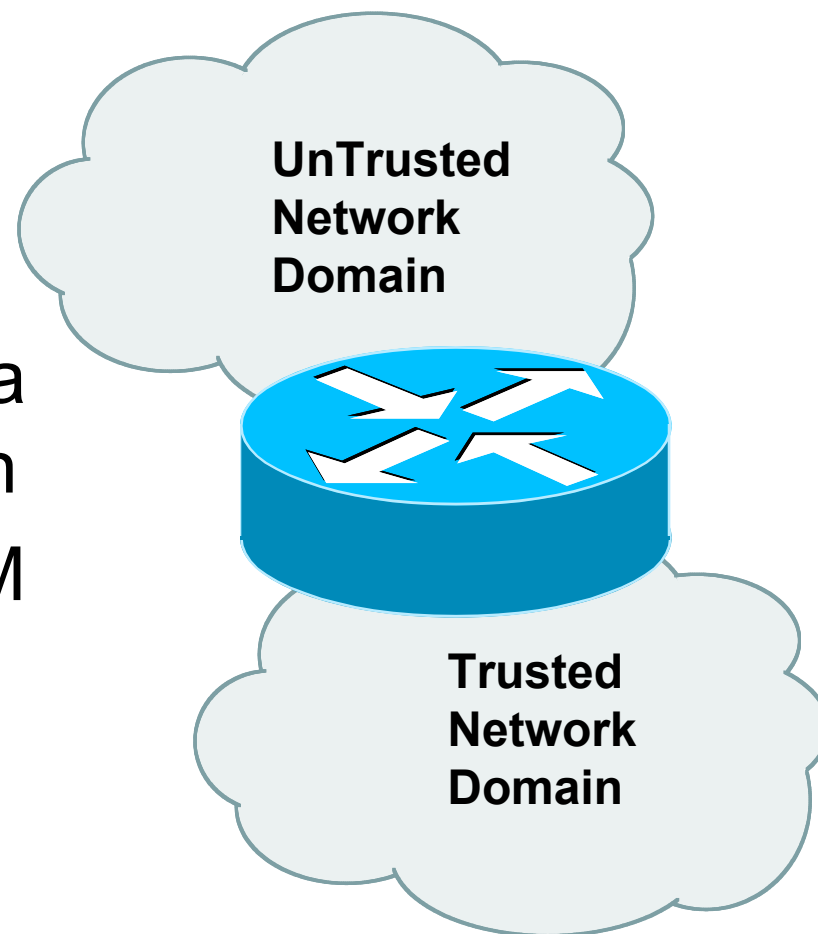


Agenda (Day 3)

- Securing Routing Protocols
 - Route Authentication (MD5)
 - Filtering Policies
 - Flap Damping
 - Prefix Limits
- Auditing Tools
 - Sniffers and Traffic Analyzers
 - Vulnerability Assessment (Nessus, NMAP)
- Mitigating DoS Attacks
 - Blackhole /Sinkhole Routing
 - Rate Limiting
- LAB

● ● ● | Role of the Router

- Forwards packets at network layer
- First point of entry TO a trusted network domain
- Last point of exit FROM a trusted network domain





RFC2827 – Ingress Filtering

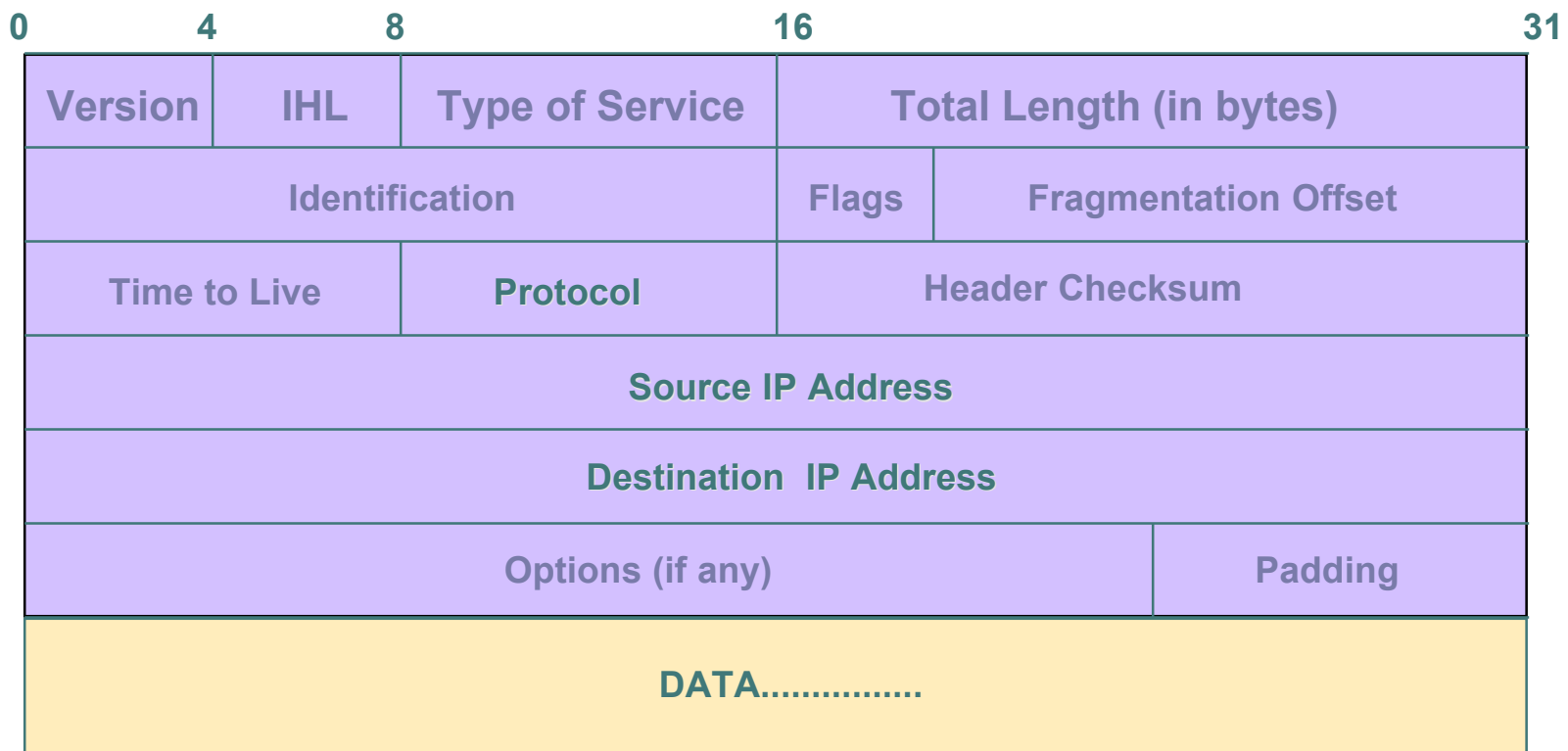
If an ISP is aggregating routing announcements for multiple downstream networks, strict traffic filtering should be used to prohibit traffic which claims to have originated from outside of these aggregated announcements.

The ONLY valid source IP address for packets originating from a customer network is the one assigned by the ISP (whether statically or dynamically assigned).

An edge router could check every packet on ingress to ensure the user is not spoofing the source address on the packets which he is originating.



IP Header Format



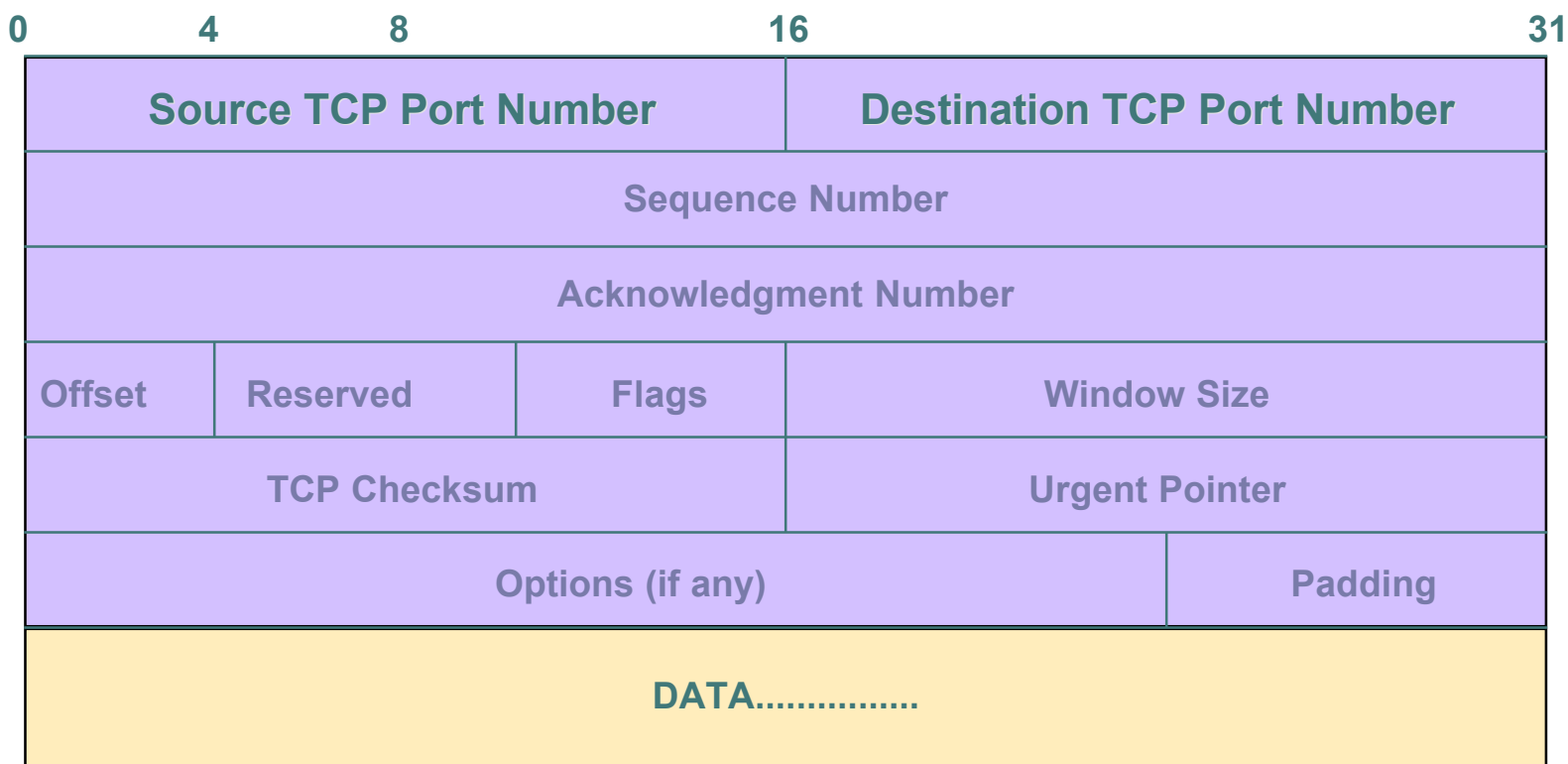


TCP (Transport Control Protocol)

- Provides reliable virtual circuits to user processes
- Lost or damaged packets are resent
- Sequence numbers maintain ordering
- All packets except first contain ACK #
(ACK# = sequence number of last sequential byte successfully received)



TCP Header Format



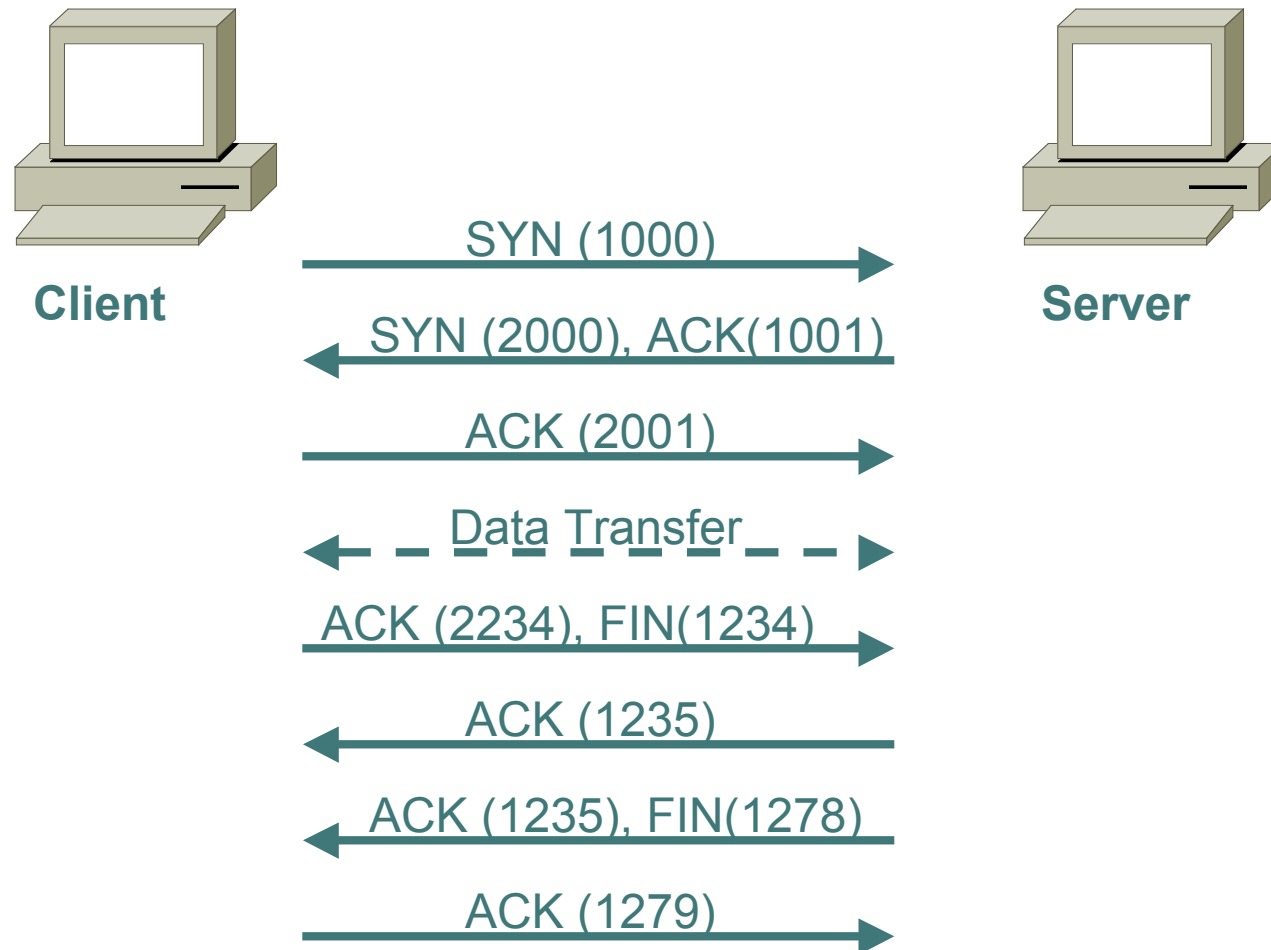


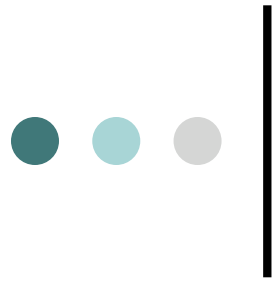
TCP Control Flags

URG	ACK	PSH	RST	SYN	FIN
-----	-----	-----	-----	-----	-----

- URG: indicates urgent data in data stream
- ACK: acknowledgement of earlier packet
- PSH: flush packet and not queue for later delivery
- RST: reset connection due to error or other interruption
- SYN: used during session establishment to synchronize sequence numbers
- FIN: used to tear down a session

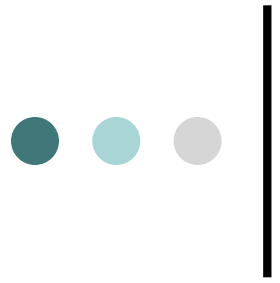
TCP Session





TCP Port Numbers

- Port numbers < 1024 are privileged ports
- Destination port is fixed
- Source port is randomly generated

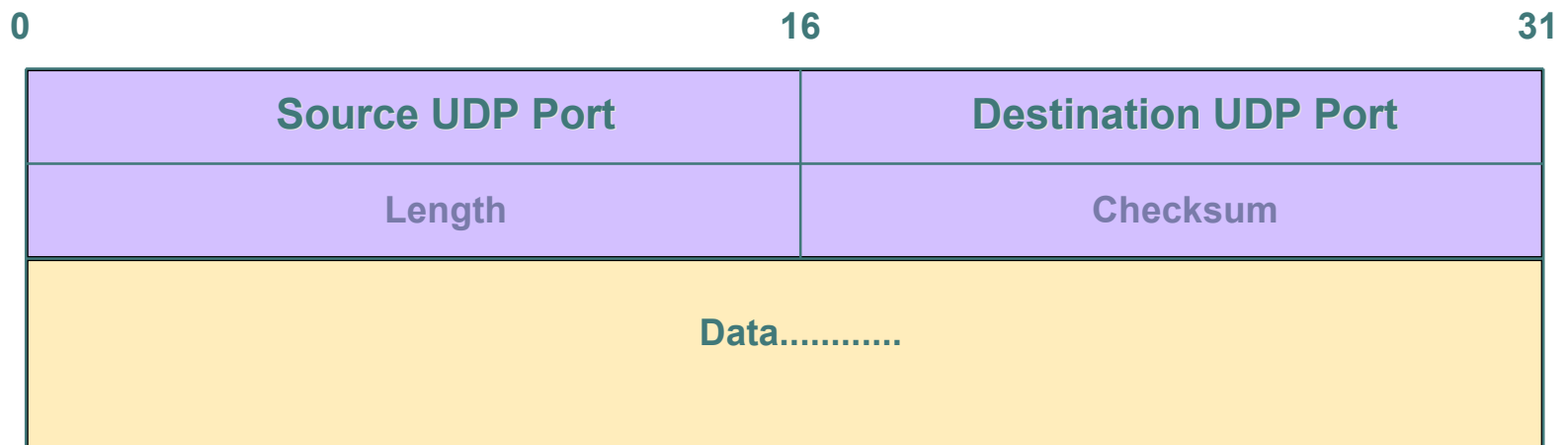


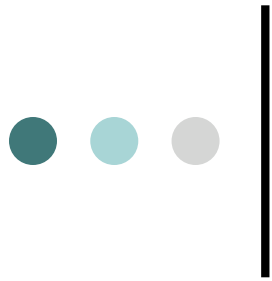
UDP (User Datagram Protocol)

- Delivery is on a best-effort basis
 - No error correction
 - No retransmission
 - No lost, duplicate, re-ordered packet detection
- Easier to spoof than TCP packets
 - no handshake
 - no sequence numbers



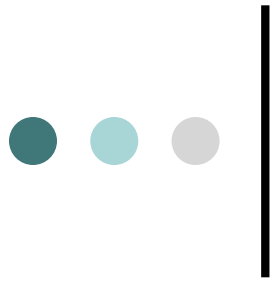
UDP Header Format





ICMP

- Transmits command and control information
 - ICMP Echo
 - determines whether another system is alive
 - ICMP Destination Unreachable
 - No route to destination
 - ICMP Source Quench
 - Slow down number of packets sent



ICMP

- IP Hdr and first 64 bits of transport header
 - included in ICMP Message
 - limits scope of changes dictated by ICMP
 - older implementations do not use this info
 - Destination Unreachable messages can affect all connections between a pair of hosts
 - Redirect messages should only be obeyed by hosts (from router or directly connected network)



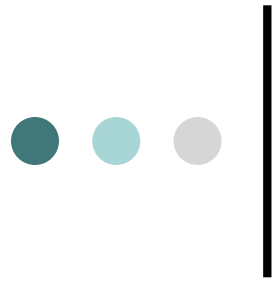
ICMP Message Types

Message Type	Value	Description
Echo Reply	0	Ping response if system alive
Destination Unreachable	3	Earlier IP message not deliverable
Source Quench	4	Packets received too fast to process
Redirect	5	Traffic should be directed to another router
Echo	8	Send a ping
Time Exceeded	11	Max # of hops in TTL field is exceeded
Parameter Problem	12	Bad parameter in header field
Timestamp	13	Includes time on sending machine and requests time on destination machine
Timestamp Reply	14	Timestamp response
Information Request	15	Used by host to determine which network it is on
Information Reply	16	Contains response to information request



IP Fragments

- Only first fragmented packet contains port number information
- Firewall should have capability of fragment reassembly



How Do We Control Traffic ?

- Firewalls
 - Simple Rule-Based
 - Proxy
 - Stateful
- Which One Is Needed ?
- Where Do I Put It ?
- What Do I Configure ?



Firewall Cost Tradeoff

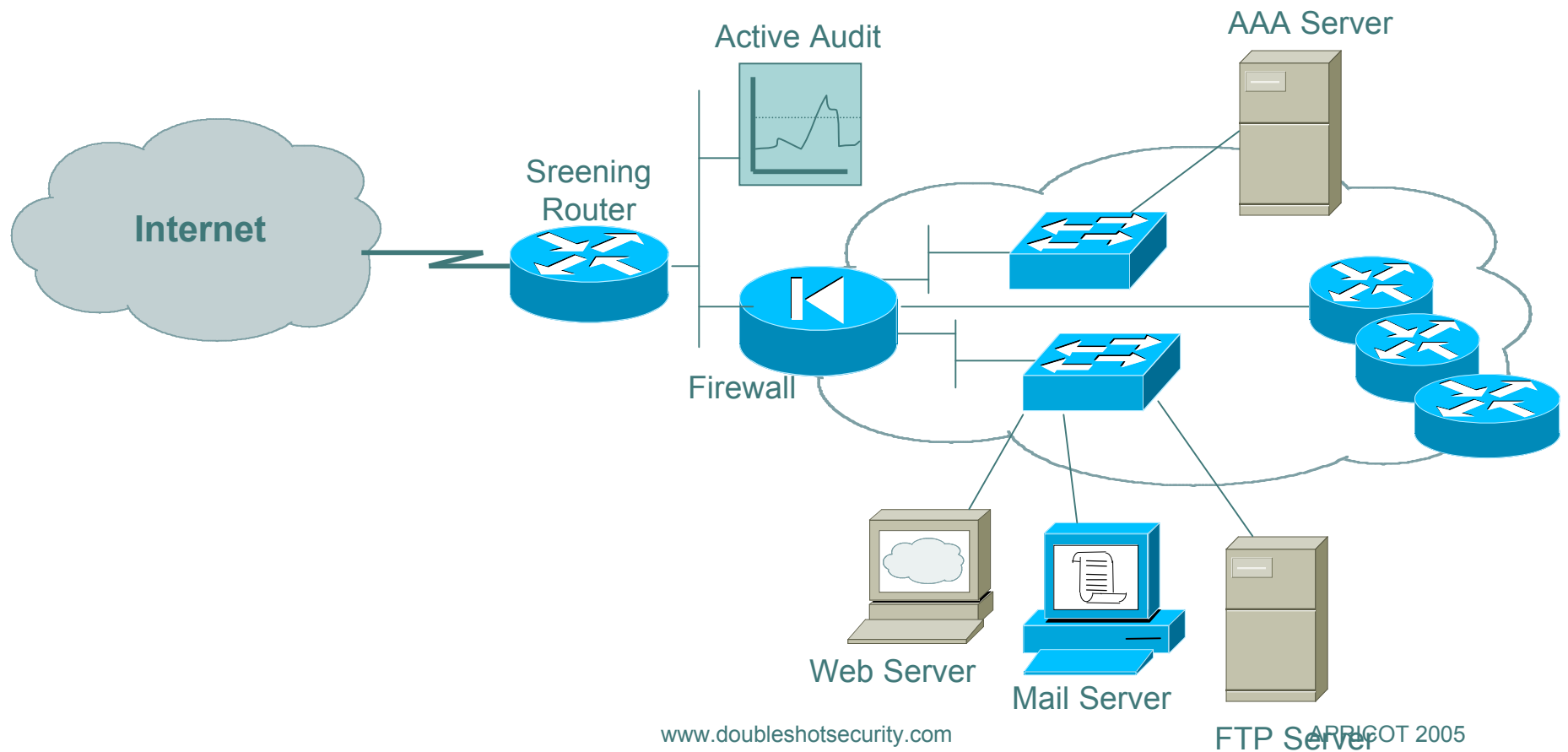
USING A FIREWALL

- Hardware cost and maintenance
- Software purchase and updates
- Administrative setup and training
- Lost business from blocked service
- Loss of some service

NOT USING A FIREWALL

- Effort spent dealing with break-ins
- Legal costs

Typical Secure Infrastructure Architecture





Filtering Recommendations

- Log filter port messages properly
- Allow only internal addresses to enter the router from the internal interface
- Block packets from outside (untrusted) that are obviously fake or commonly used for attacks
- Block packets that claim to have a source address of any internal (trusted) network.



Filtering Recommendations

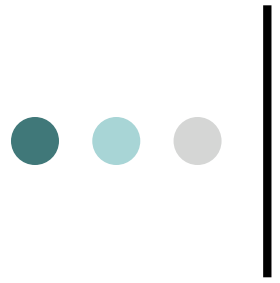
- Block incoming loopback packets and RFC 1918 networks
 - 127.0.0.0
 - 10.0.0.0 – 10.255.255.255
 - 172.16.0.0 – 172.31.0.0
 - 192.168.0.0 – 192.168.255.255
- Block multicast packets (if NOT using multicast)
- Block broadcast packets (careful of DHCP and BOOTP users)
- Block incoming packets that claim to have same destination and source address



DoS Filtering

(* these networks may be reallocated)

Description	Network
default	0.0.0.0 /8
loopback	127.0.0.0 /8
RFC 1918	10.0.0.0 /8
RFC 1918	172.16.0.0 /12
RFC 1918	192.168.0.0 /16
Net Test	192.0.2.0 /24
Testing devices *	192.18.0.0 /15
IPv6 to IPv4 relay *	192.88.99.0 /24
RFC 1918 nameservers *	192.175.48.0 /24
End-node auto configuration *	169.254.0.0 /16



Email Spam Sources

- Open relays and proxies
- Compromised machines
- Direct Spam sources
- Insecure Webmail interfaces / Perl scripts



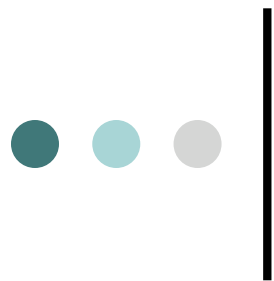
Preventing Outbound SPAM

- Scan network for open relays and proxies
- Block compromised hosts until fixed
- Block outbound port 25 for dynamic IP addresses
- Filter inbound access to known proxy ports



Filtering Inbound SPAM

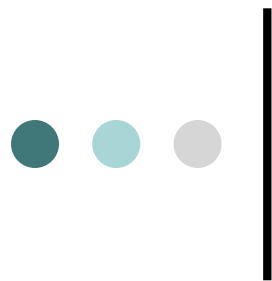
- Check SMTP headers
- Build DNS block lists (DNSBLs)
- HELO filtering
- Use SPAM filters (Spamassassin, Razor)
- Block routes to major spammers



EMAIL (SMTP) Filtering

○ Sample SMTP Filtering

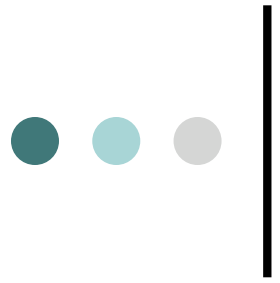
- Permit outgoing traffic to port 25
- Permit incoming traffic from port 25
- Permit our trusted hosts with dst port 25
- Permit all other traffic with src port 25 and ACK flag set (the reply)



Defining Filtering Rules (SMTP)

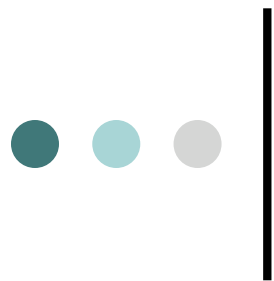
Direction	SRC IP address	DST IP address	Protocol	SRC Port	DST Port	ACK set	Description
In	External	Internal	TCP	>1023	25	*	Incoming mail Sender to recipient
Out	Internal	External	TCP	25	>1023	Yes	Incoming mail Recipient to sender
Out	Internal	External	TCP	>1023	25	*	Outgoing mail Sender to recipient
In	External	internal	TCP	25	>1023	Yes	Outgoing mail Recipient to sender

* ACK not set on first packet but set on all subsequent packets

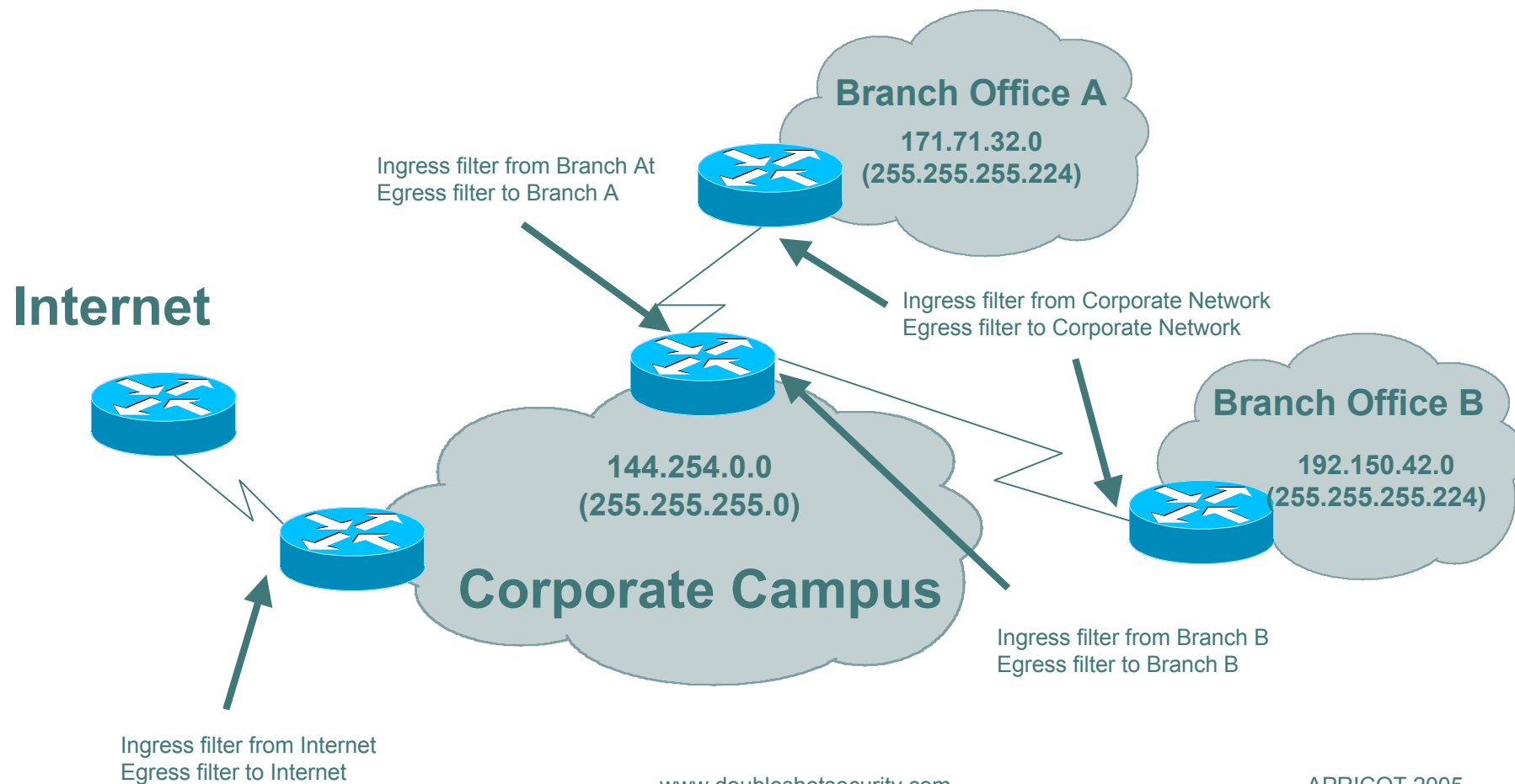


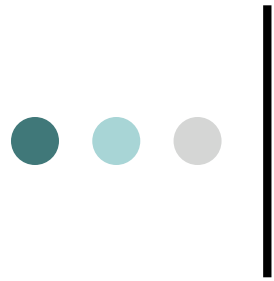
Filtering Issues

- Ordering
 - What sequence is packet inspected in?
- Performance
 - Are there any limitations?
- Logging
 - Get appropriate information
 - Timestamps



Simple Filtering Example





Branch Router Policy

Ingress filtering:

- deny all rfc 1918 and special use addresses from entering the branch network
- deny all traffic with an IP source address that matches the branch network address allocation
- permit all other traffic

Egress filtering:

- permit only traffic with an IP source address that matches the branch network
- deny all other traffic



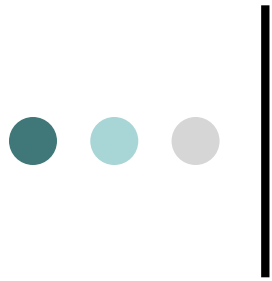
Branch Router Configuration

The configuration is as follows: (for branch A router)

```
access-list 133 deny ip host 0.0.0.0 any
access-list 133 deny ip 127.0.0.0 0.255.255.255 any
access-list 133 deny ip 10.0.0.0 0.255.255.255 any
access-list 133 deny ip 172.16.0.0 0.15.255.255 any
access-list 133 deny ip 192.168.0.0 0.0.255.255 any
access-list 133 deny ip 192.0.2.0 0.0.0.255 any
access-list 133 deny ip 169.254.0.0 0.0.255.255 any
access-list 133 deny ip 240.0.0.0 15.255.255.255 any
access-list 133 deny ip 171.71.32.0 0.0.0.31 any
access-list 133 permit ip any any
```

```
access-list 144 permit ip 171.71.32.0 0.0.0.31 any
access-list 144 deny ip any any
```

```
interface BRI0
description To Corporate Network
ip access-group 133 in
ip access-group 144 out
```

NAS Router Policy

Ingress filtering:

- permit only traffic with an IP source address of branch networks
- deny all other traffic

Egress filtering:

- deny all rfc 1918 and special use addresses from propagating to branch networks
- deny all traffic with an IP source address that matches the branch network address allocation
- permit all other traffic

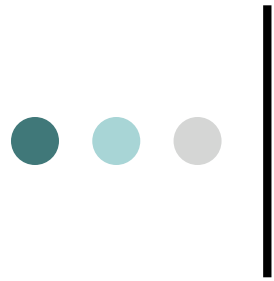


NAS Router Configuration

```
access-list 133 permit ip 171.71.32.0 0.0.0.31 any
access-list 133 permit ip 192.150.42.0 0.0.0.31 any
access-list 133 deny ip any any
```

```
access-list 144 deny ip host 0.0.0.0 any
access-list 144 deny ip 127.0.0.0 0.255.255.255 any
access-list 144 deny ip 10.0.0.0 0.255.255.255 any
access-list 144 deny ip 172.16.0.0 0.15.255.255 any
access-list 144 deny ip 192.168.0.0 0.0.255.255 any
access-list 144 deny ip 192.0.2.0 0.0.0.255 any
access-list 144 deny ip 169.254.0.0 0.0.255.255 any
access-list 144 deny ip 240.0.0.0 15.255.255.255 any
access-list 144 deny ip 171.71.32.0 0.0.0.31 any
access-list 144 deny ip 192.150.42.0 0.0.0.31 any
access-list 144 permit ip any any
```

```
interface Serial 0:23
description To Branch Offices
ip access-group 133 in
ip access-group 144 out www.doubleshotsecurity.com
```



Internet Router Policy

Ingress filtering:

- deny all rfc 1918 and special use addresses from entering the corporate network
- deny all traffic with an IP source address of the corporate network or branch networks
- permit all other traffic

Egress filtering:

- permit only traffic with an IP source address of the corporate network and branch networks
- deny all other traffic



Internet Router Configuration

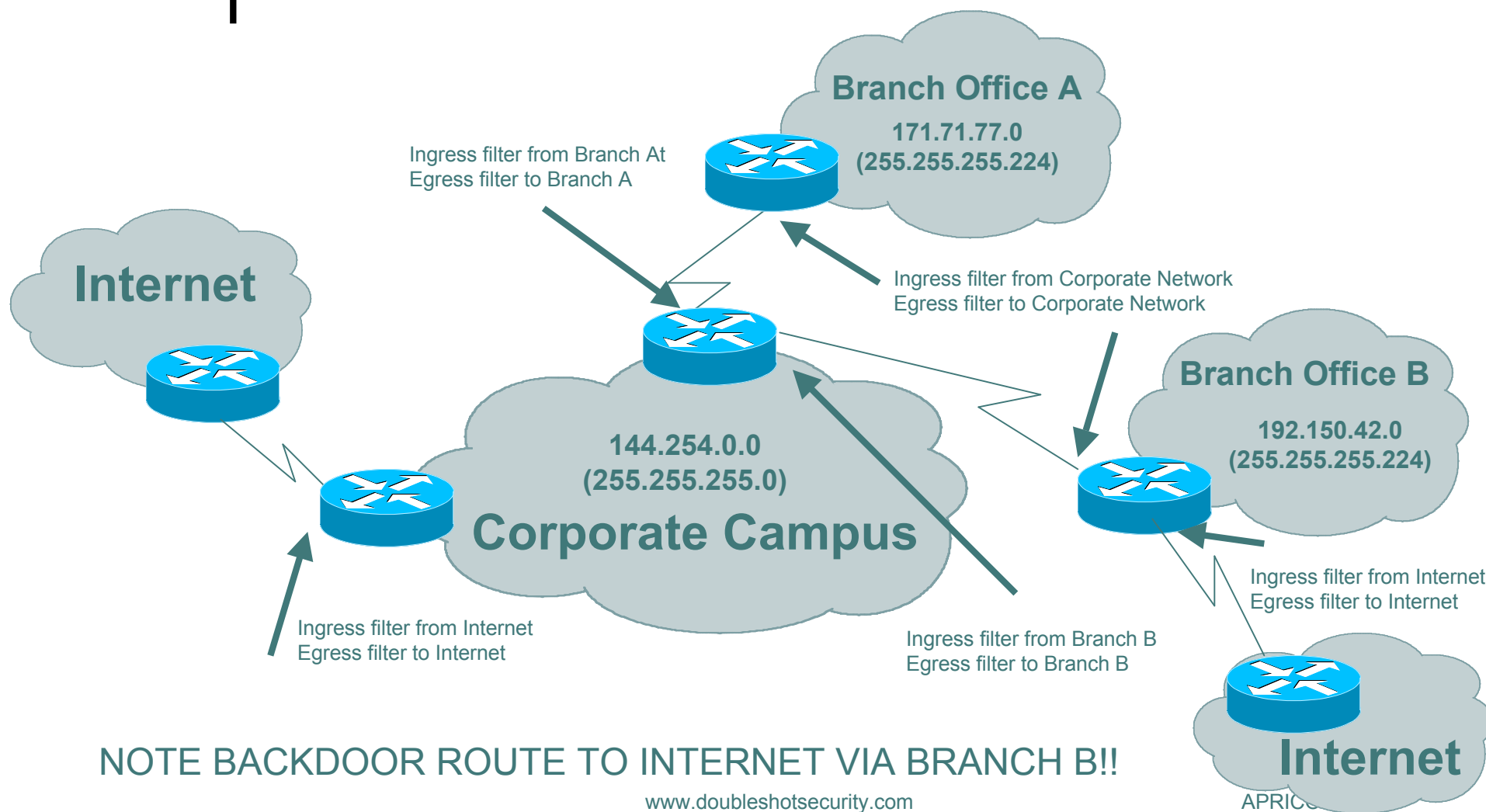
```
access-list 133 deny ip host 0.0.0.0 any
access-list 133 deny ip 127.0.0.0 0.255.255.255 any
access-list 133 deny ip 10.0.0.0 0.255.255.255 any
access-list 133 deny ip 172.16.0.0 0.15.255.255 any
access-list 133 deny ip 192.168.0.0 0.0.255.255 any
access-list 133 deny ip 192.0.2.0 0.0.0.255 any
access-list 133 deny ip 169.254.0.0 0.0.255.255 any
access-list 133 deny ip 240.0.0.0 15.255.255.255 any
access-list 133 deny ip 144.254.0.0 0.0.255.255 any
access-list 133 deny ip 171.71.32.0 0.0.0.31 any
access-list 133 deny ip 192.150.42.0 0.0.0.31 any
access-list 133 permit ip any any
```

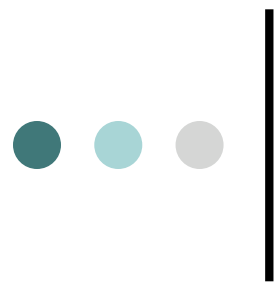
```
access-list 144 permit ip 144.254.0.0 0.0.255.255 any
access-list 144 permit ip 171.71.32.0 0.0.0.31 any
access-list 144 permit ip 192.150.42.0 0.0.0.31 any
access-list 144 deny ip any any
```

```
interface Serial 0/0
description To Internet
ip access-group 133 in
ip access-group 144 out
```



Advanced Filtering Example

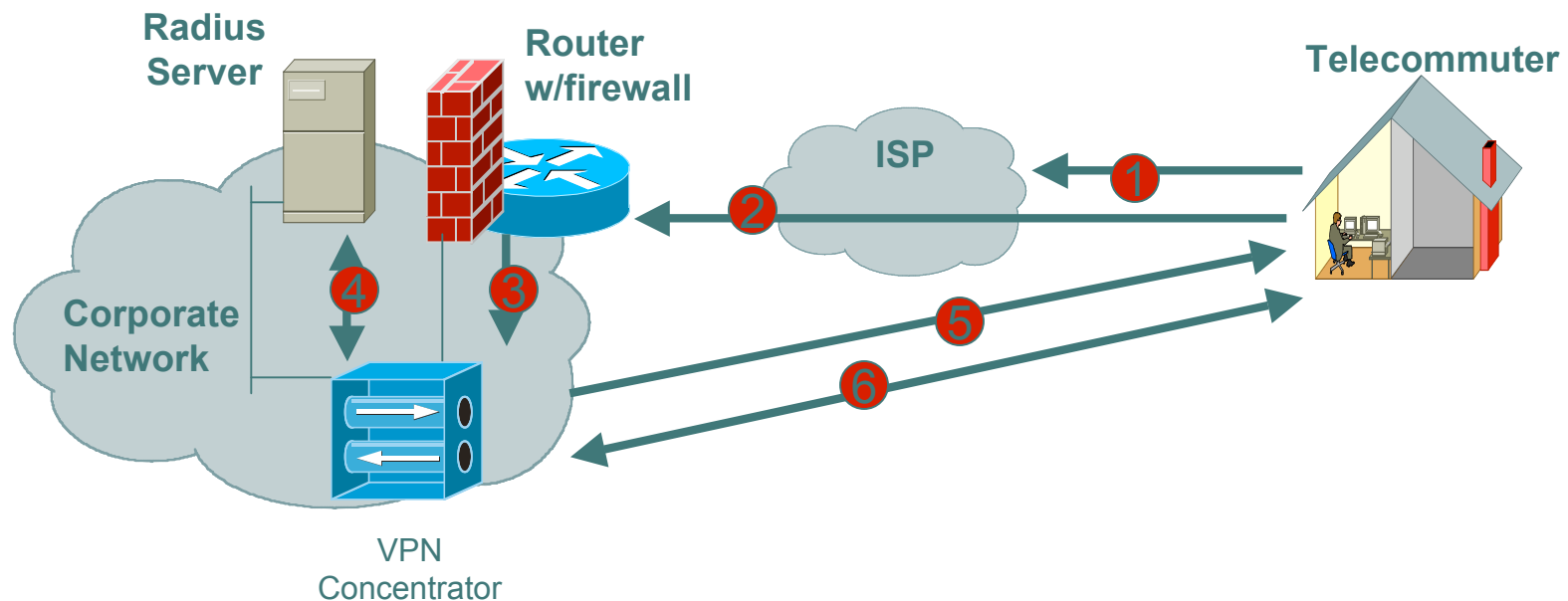




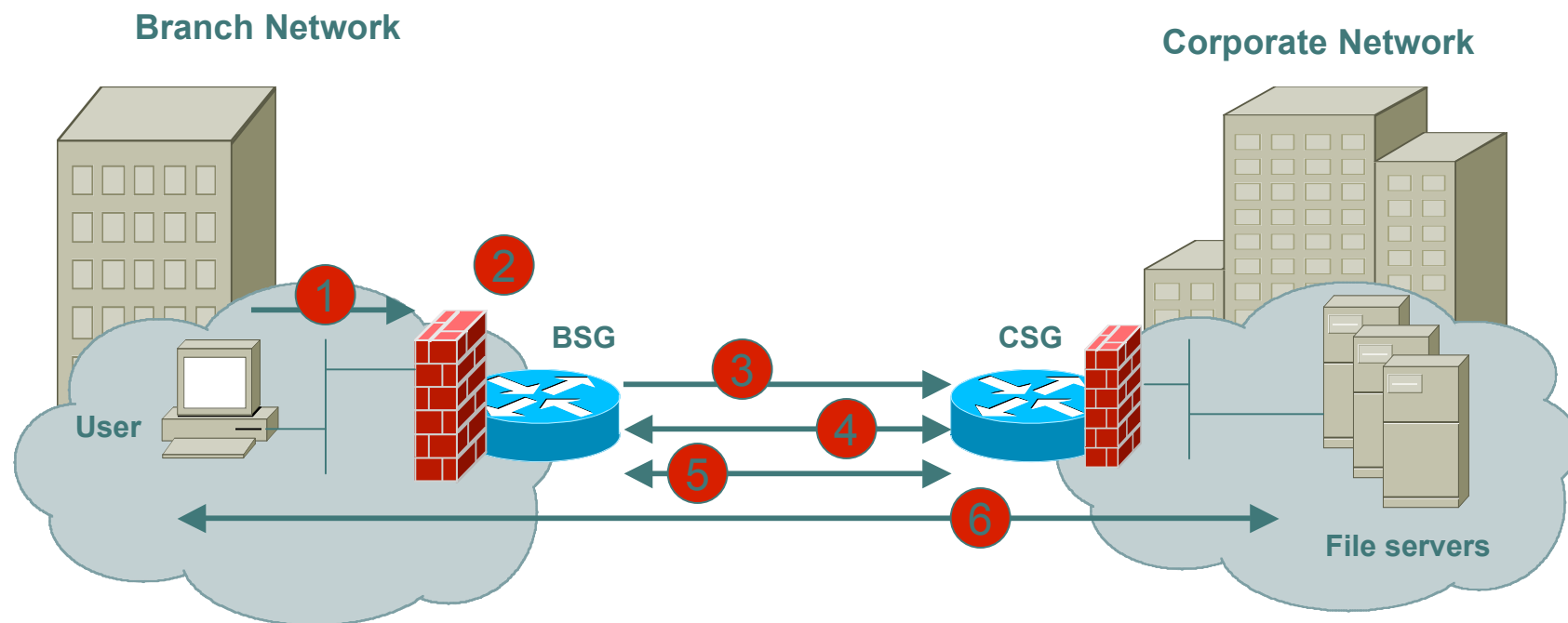
SSL/TLS and IPsec

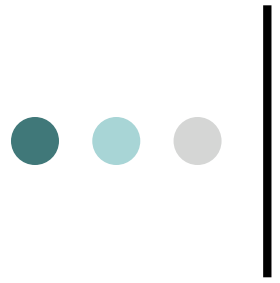
Any VPN is **not** automagically secure. You need to add security functionality to create secure VPNs. That means using firewalls for access control and using SSL/TLS & IPsec for confidentiality and data origin authentication.

Access VPN



Intranet VPN





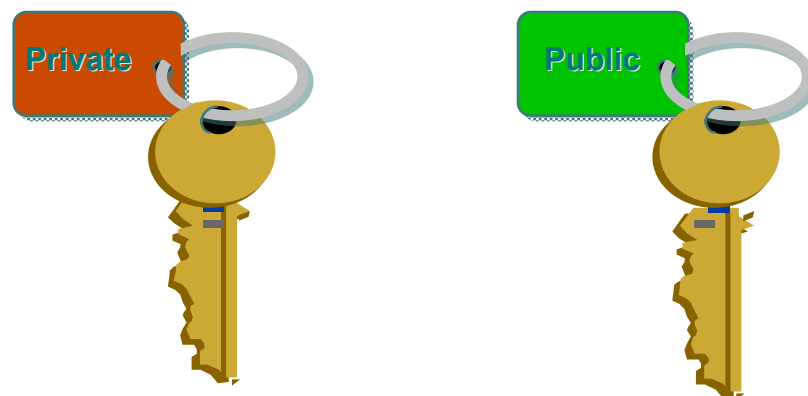
Crypto 101

- Cryptography Is Used For ?
 - Authentication Protocols
 - Data Origin Authentication
 - Data Integrity
 - Data Confidentiality
- Crypto Algorithms
 - Asymmetric (Public Key) Encryption
 - Symmetric (Secret Key) Encryption
 - Diffie-Hellman
 - Hash Functions

Public Key Encryption

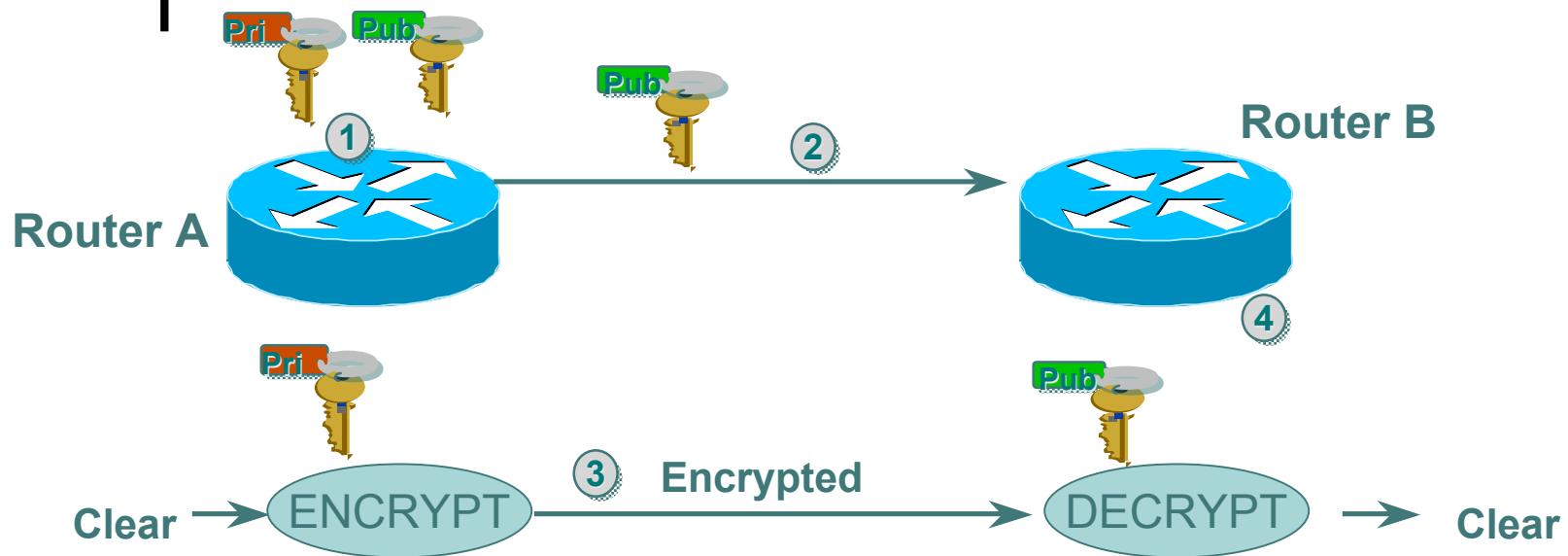
Uses public/private keys

- Keep private key private
- Anyone can see public key



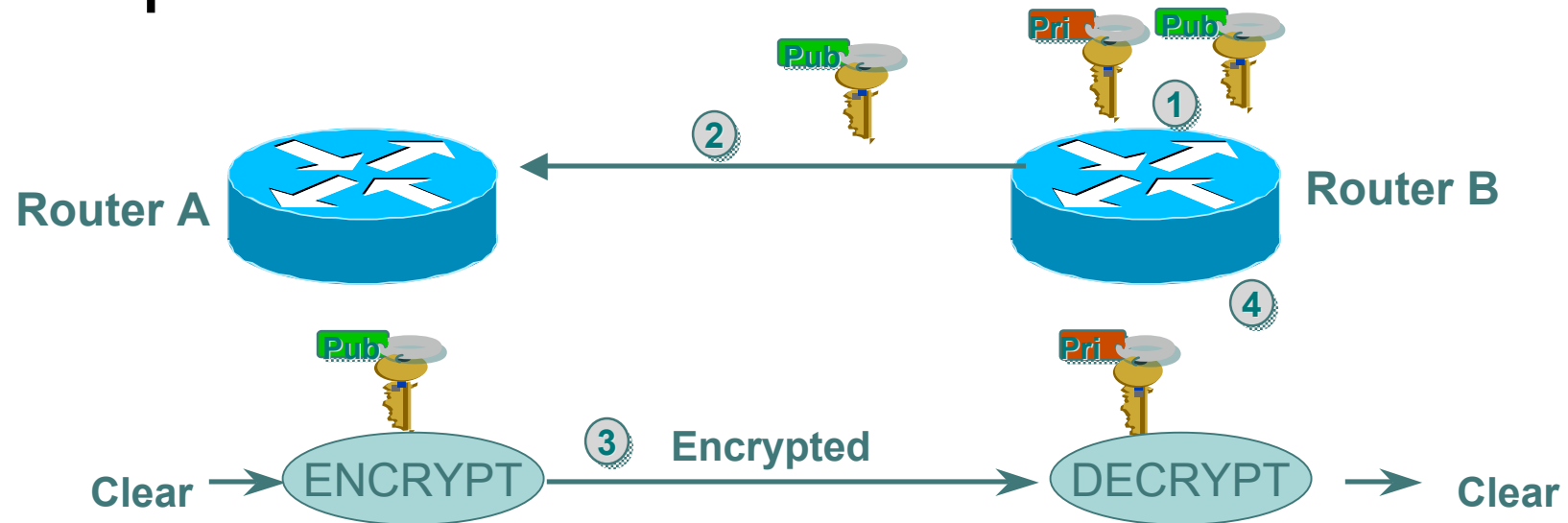
Computing Key pair is computationally expensive!!
Common Algorithms: RSA, El Gamal

Data Origin Authentication



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 Integrity and Confidentiality



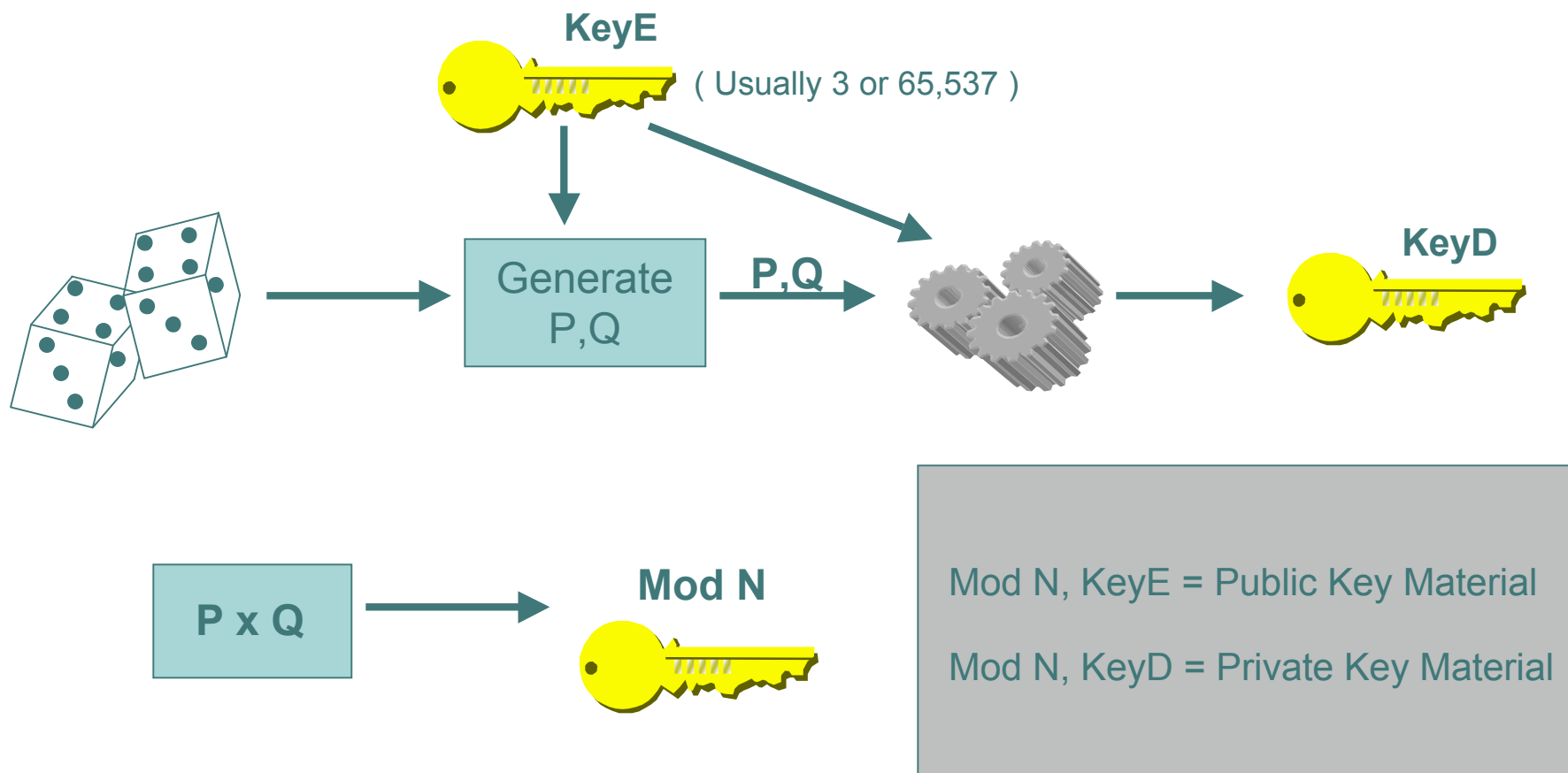
1. Router B generates public/private key pair
2. Router B sends its public key to Router A
3. Router A encrypts packet with router B's public key and sends encrypted packet to Router B
4. Router B receives encrypted packet and decrypts with its' private key



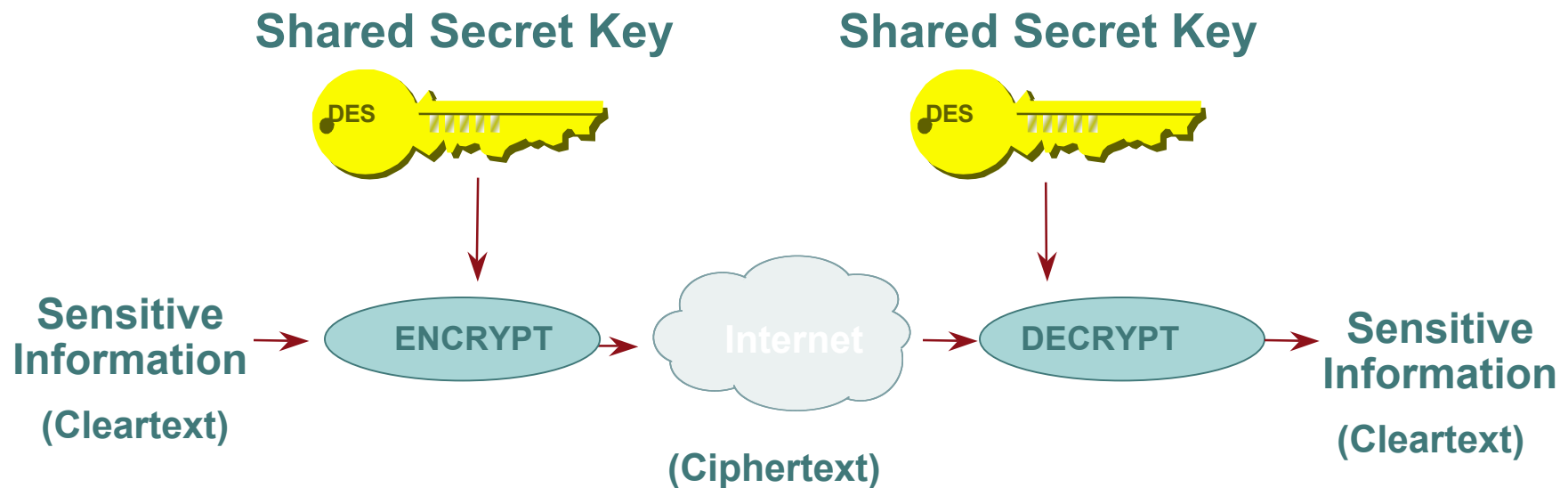
RSA Public Key Cryptography

- Based on relative ease of multiplying large primes together but almost impossible to factor the resulting product
- RSA keys: 3 special numeric values
- Algorithm produces public keys that are tied to specific private keys
- Provides both digital signatures and public-key encryption

Generating RSA Keys

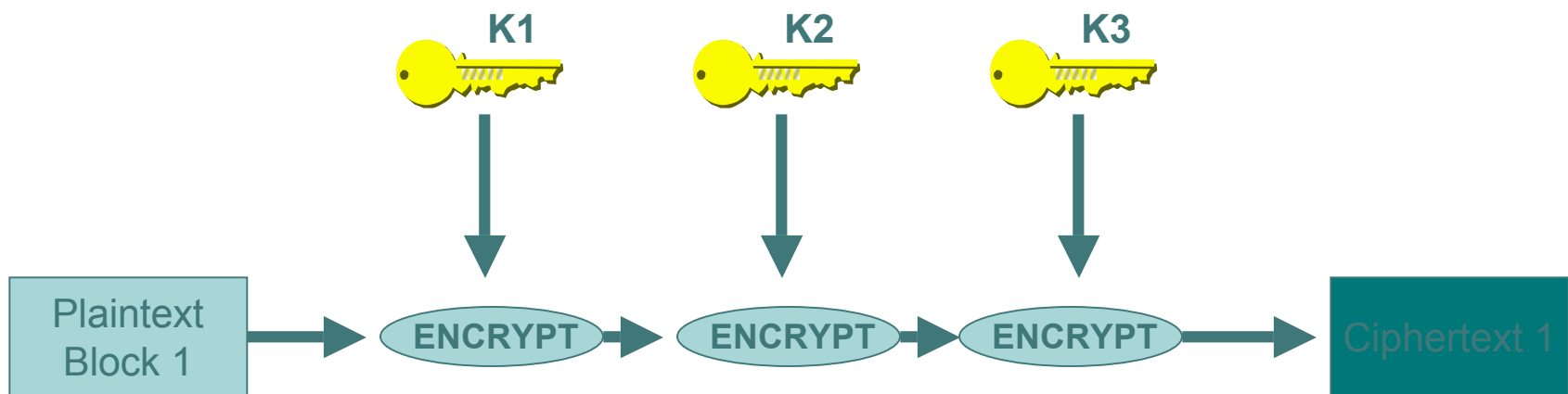


Secret Key Encryption



Common Algorithms: DES, 3DES, AES, IDEA

• • • | Triple DES (3DES)



- Many applications use $K3=K1$, yielding a key length of 112 bits
- Interoperable with conventional DES if $K1=K2=K3$



AES

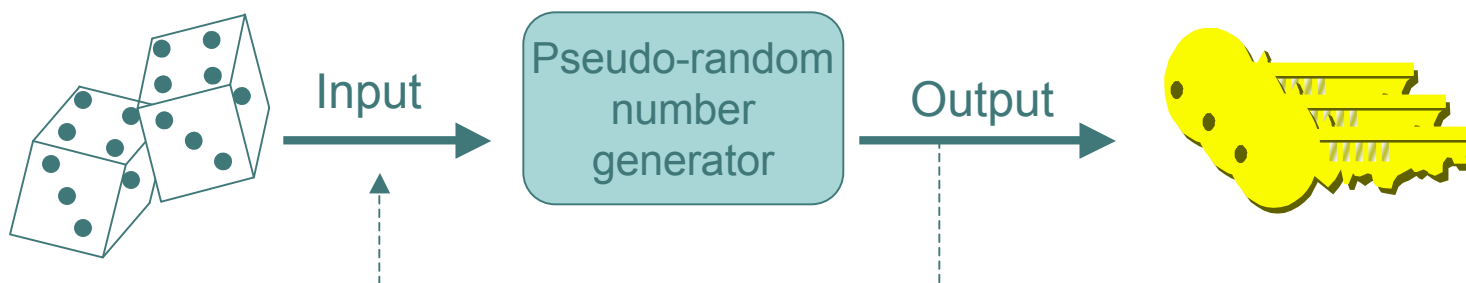
- Published in November 2001
- Rijndael algorithm developed by Dr. Joan Daemen and Dr. Vincent Rijmen
- Symmetric Block Cipher
 - 128 bit blocks
 - 3 key lengths: 128, 192, and 256 bits
 - symmetric and parallel
 - low memory requirement



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	$2^{256} = 1.1 \times 10^{77}$

Producing Effective Keys



- ❑ Producing random seed value can be slow and inefficient
- ❑ PRNG used when generating many separate keys
- ❑ Properties of sequence #'s produced by a good PRNG
 - ❑ Equal chance that a given number falls anywhere within the range of numbers being generated
 - ❑ The sequence should not repeat itself

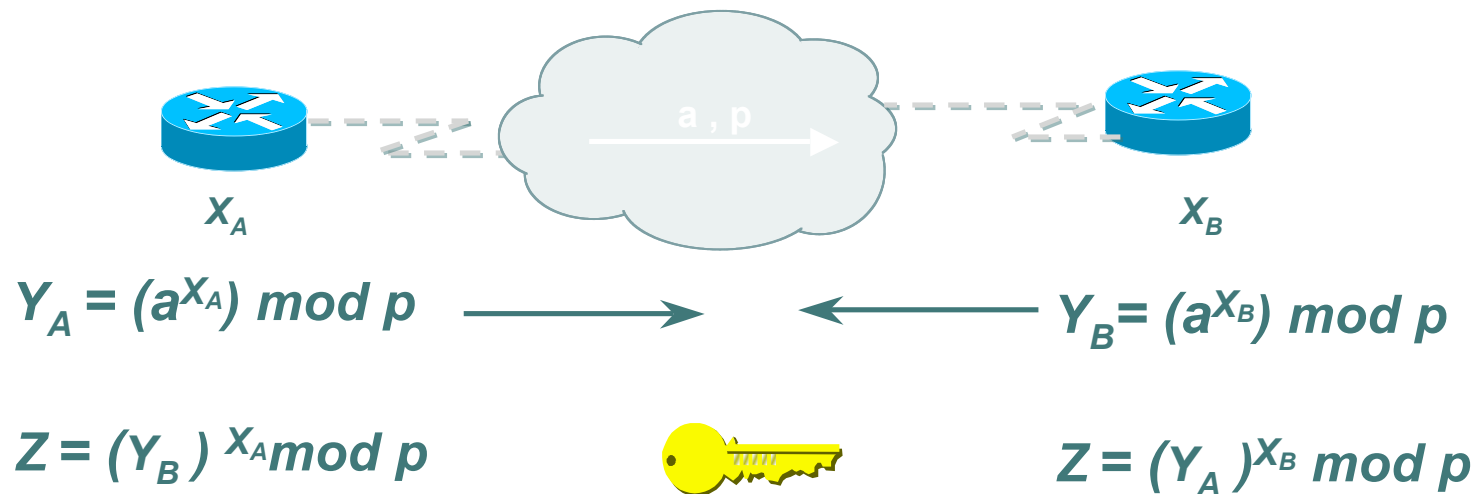


Scalability with Secret Key Cryptography

Configuring shared secret keys easily
becomes administrative nightmare

Automated mechanism to securely
derive secret keys => Diffie-Hellman

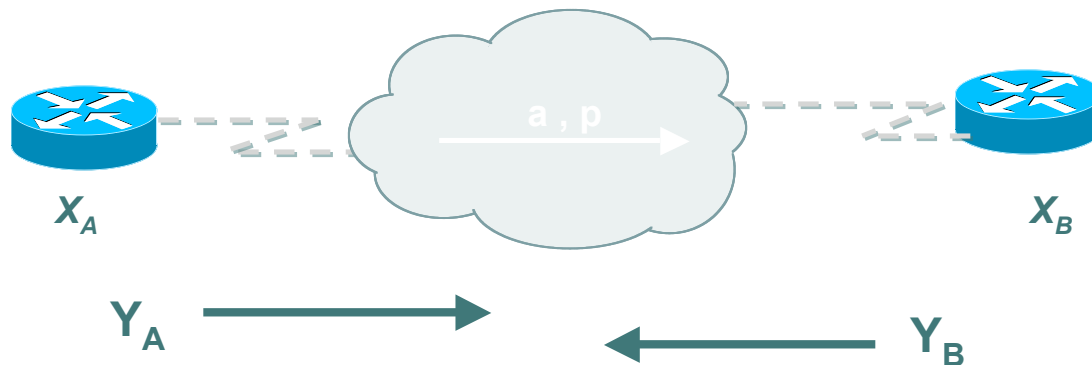
Deriving Secret Keys Using Public Key Technology (e.g., Diffie-Hellman)



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!!

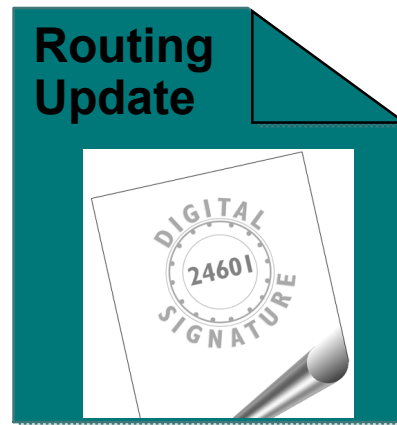


Hash Functions

A *hash function* takes an input message of arbitrary length and outputs fixed-length code. The fixed-length output is called the *hash*, or the *message digest*, of the original input message.

Common Algorithms: MD-5 (128), SHA-1 (160)

● ● ● | Digital Signatures



- A digital signature is a message appended to a packet
- Used to prove the identity of the sender and the integrity of the packet



Digital Signatures

- 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



Crypto 101 Summary

- Public Key Encryption
 - Typically used for data origin authentication
 - Often combined with hash function
- Secret Key Encryption
 - Typically used for data confidentiality
- Diffie-Hellman Algorithm
 - Uses public-key cryptography to derive secret key
 - Exchanges need to be authenticated
- Hash Functions
 - Easy to compute
 - Typically used for data origin authentication and data integrity
- Digital Signatures
 - Combines hash functions with public key cryptography

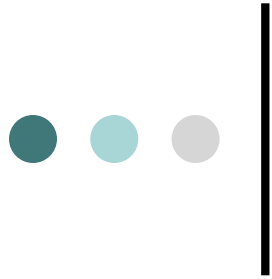


SSL/TLS Security Features

- Data encryption
- Server authentication
- Message integrity
- Client authentication (optional)

Note:

Separate keys are used for integrity and encryption



SSL/TLS Properties

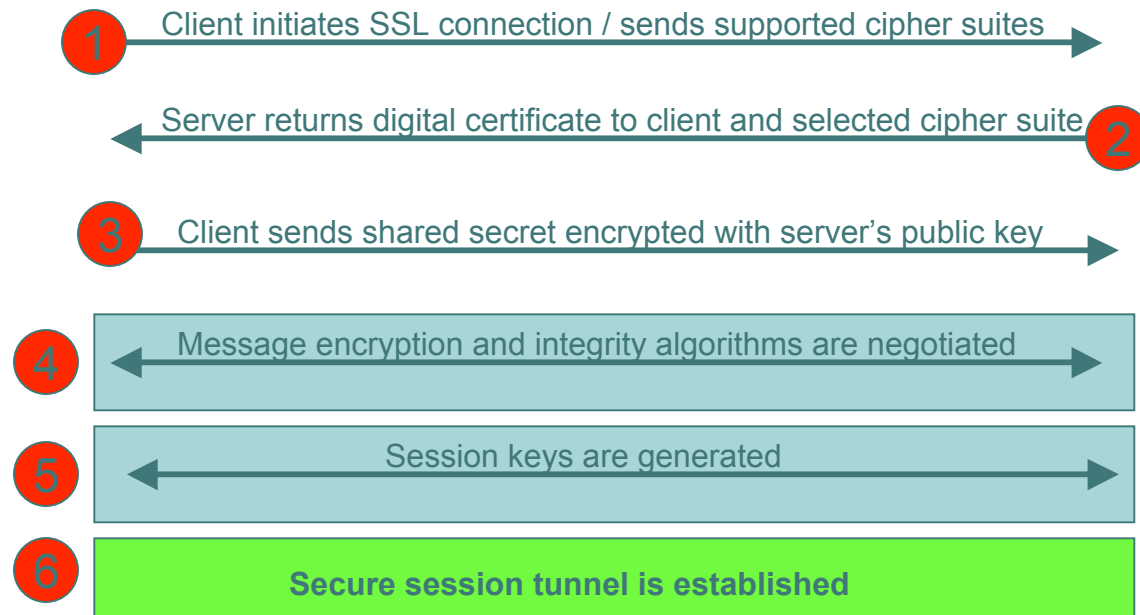
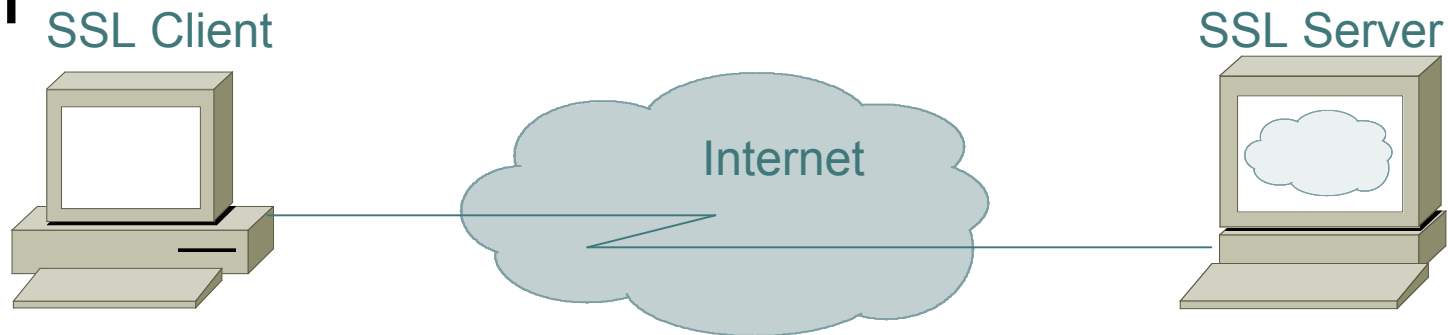
- Connection is private
 - Encryption is used after an initial handshake to define a secret key.
 - Symmetric cryptography used for data encryption (DES or RC4).
- Peer's identity can be authenticated
 - Asymmetric cryptography is used (RSA or DSS).
- Connection is reliable
 - Message transport includes a message integrity check using a keyed MAC.
 - Secure hash functions (such as SHA and MD5) are used for MAC computations.



SSL Protocol Elements

- Record Protocol
 - Functions as layer beneath all SSL messages
 - Indicates which integrity and encryption protection is applied to data
- Handshake Protocol
 - Negotiates crypto algorithms and keys
- Alert Protocol
 - Indicates errors or end of a session

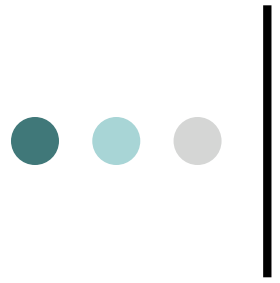
SSL Handshake Process





The SSL Record Protocol

- Each record individually encrypted and hashed
- Connections closed with a 'Close Notify'
- Previously established session can be resumed by providing session ID in 'Client Hello'
 - Abbreviated version of handshake protocol
 - Reuses previously established crypto parameters



SSL Client Authentication

- Client authentication (certificate based) is optional and not often used
- Many application protocols incorporate their own client authentication mechanism such as username/password or S/Key
- These authentication mechanisms are more secure when run over SSL



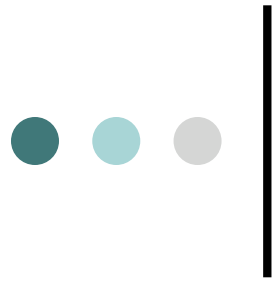
SSL/TLS Port Numbers

Protocol	Defined Port Number	SSL/TLS Port Number
HTTP	80	443
NNTP	119	563
SMTP	110	995
FTP-Data	20	989
FTP-Control	21	990
Telnet	23	992



IPsec

- Suite of protocols to secure IP traffic
 - Defined in RFC 2401-2409, RFC 2451
 - ietf.org/html.charters/ipsec-charter.html
- Components
 - AH (Authentication Header)
 - RFC requires HMAC-MD5-96 and HMAC-SHA1-96....older implementations also support keyed MD5
 - ESP (Encapsulating Security Payload)
 - RFC requires DES 56-bit CBC and Triple DES. Can also use RC5, IDEA, Blowfish, CAST, RC4, NULL
 - IKE (The Internet Key Exchange)



What Does IPsec Provide?

- Data integrity and data origin 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 data origin authentication
- Confidentiality



What Does IPsec Provide?

- 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



What is an SA?

- 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)



A Security Association Maps:

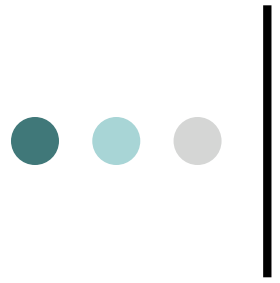
- From a host or gateway
 - To a particular IP destination address
 - With a particular security protocol (AH/ESP)
 - Using SPI selected by remote host or gateway
- To a host or gateway
 - To (one of) our IP address(es)
 - With a particular security protocol (ESP/AH)
 - Using SPI selected by us



A SPI Represents an SA

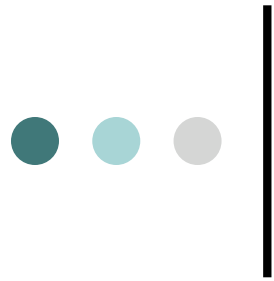
- The SPI is a 32-bit number
- The SPI is combined with the protocol (AH/ESP) and destination IP address to uniquely identify an SA
- An SA is unidirectional

When an ESP/AH packet is received, the SPI is used to look up all of the crypto parameters



IPsec Traffic Selectors

- Selectors for traffic matches....what kind of traffic will be acted on how
- Selectors include:
 - IP address or range
 - Optional IP protocol (UDP, TCP, etc)
 - Optional layer 4 (UDP, TCP) port
- Selected traffic is either protected with IPsec or dropped



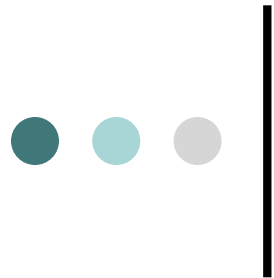
IPsec Components

- AH
 - RFC requires HMAC-MD5-96 and HMAC-SHA1-96....older implementations also support keyed MD5
- ESP
 - RFC requires DES 56-bit CBC and Triple DES. Can also use RC5, IDEA, Blowfish, CAST, RC4, NULL
- IKE



Authentication Header (AH)

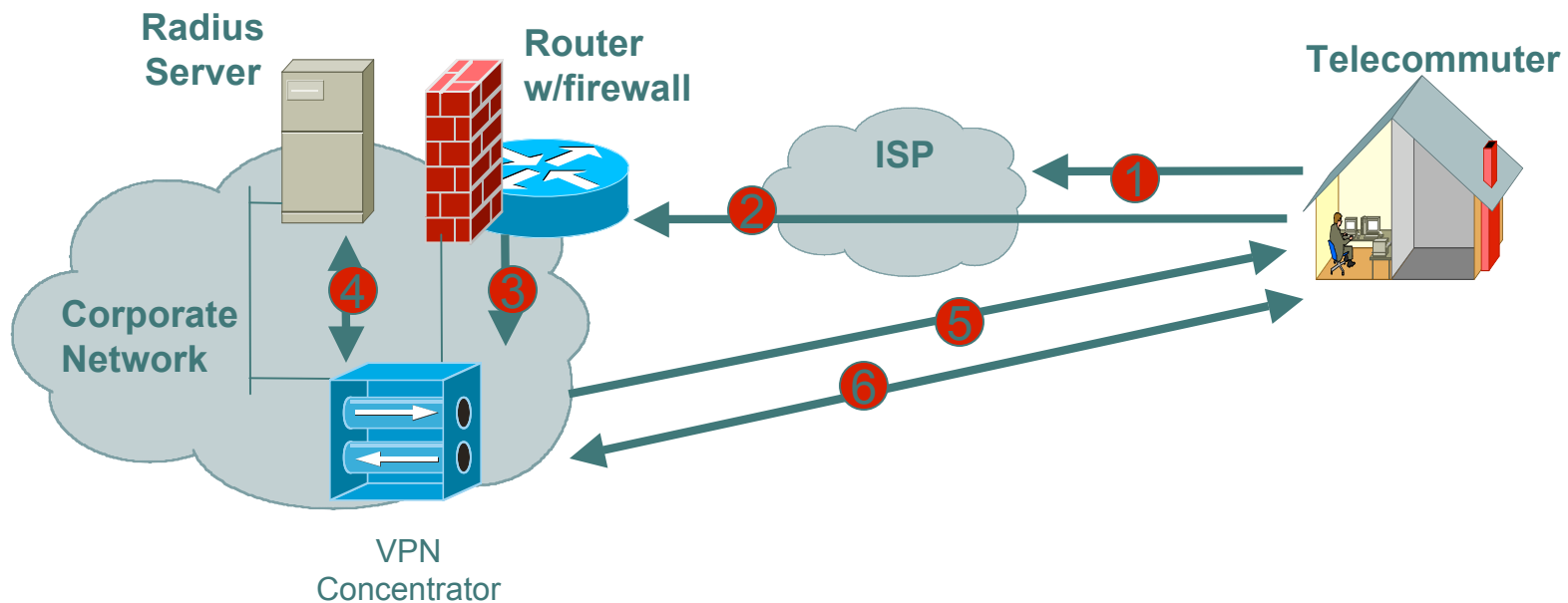
- Authentication is applied to the entire packet, with the mutable fields in the IP header zeroed out
- If both ESP and AH are applied to a packet, AH follows ESP



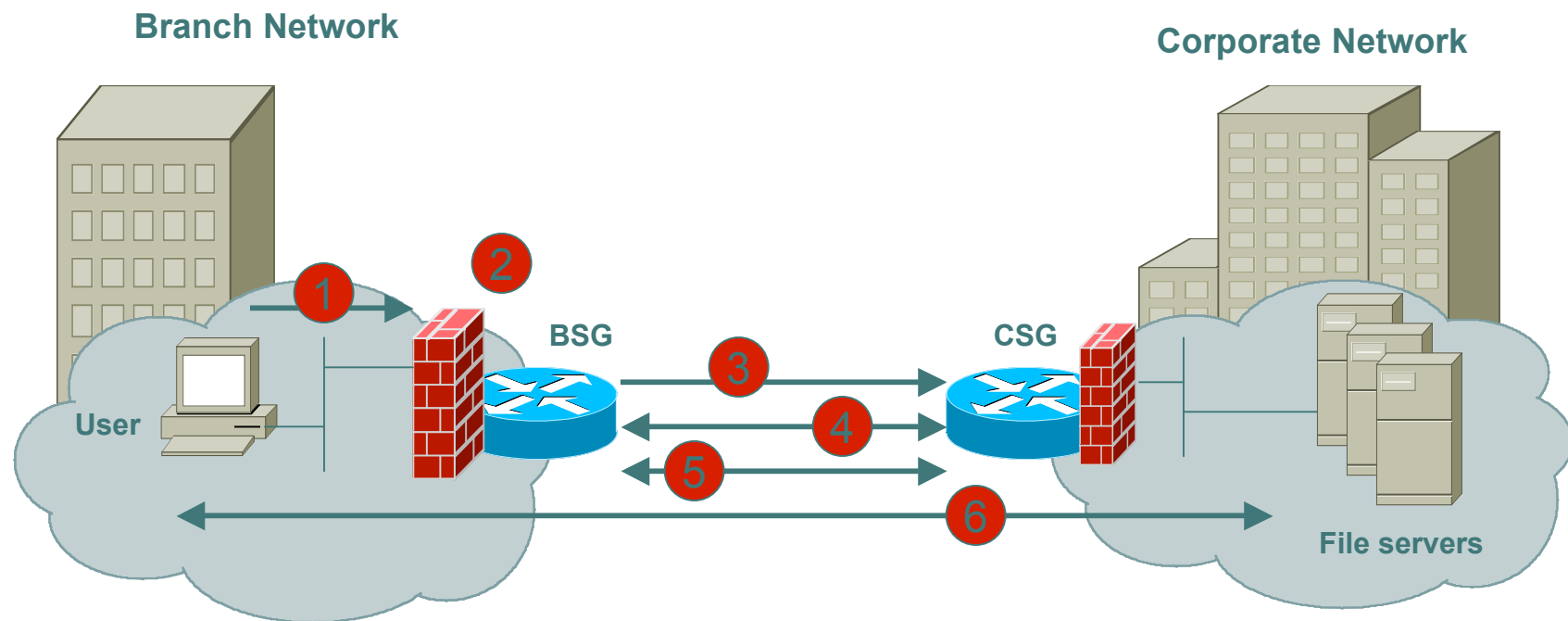
Encapsulating Security Payload (ESP)

- Must encrypt and/or authenticate in each packet (null encryption)
- Encryption occurs before authentication
- Authentication is applied to data in the IPsec header as well as the data contained as payload

AH/ESP Transport Mode

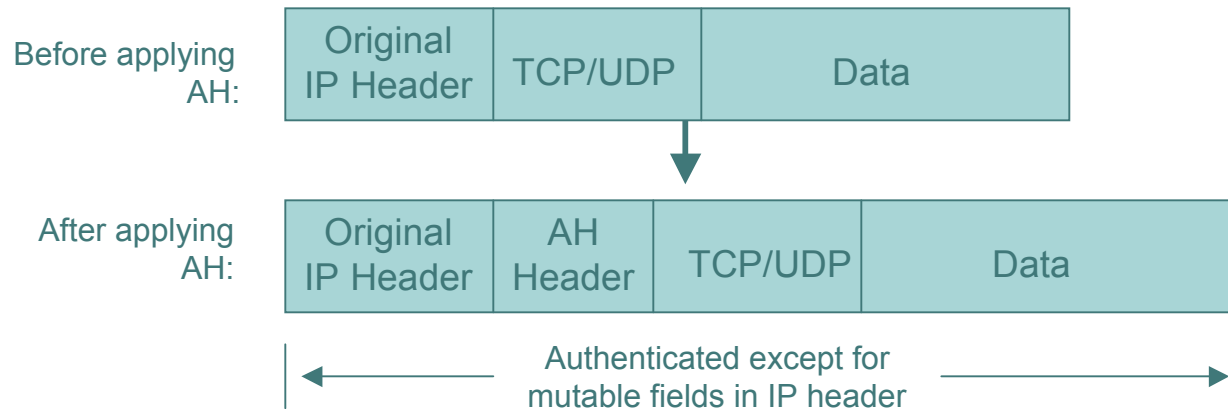


AH/ESP Tunnel Mode



Packet Format Alteration for AH Transport Mode

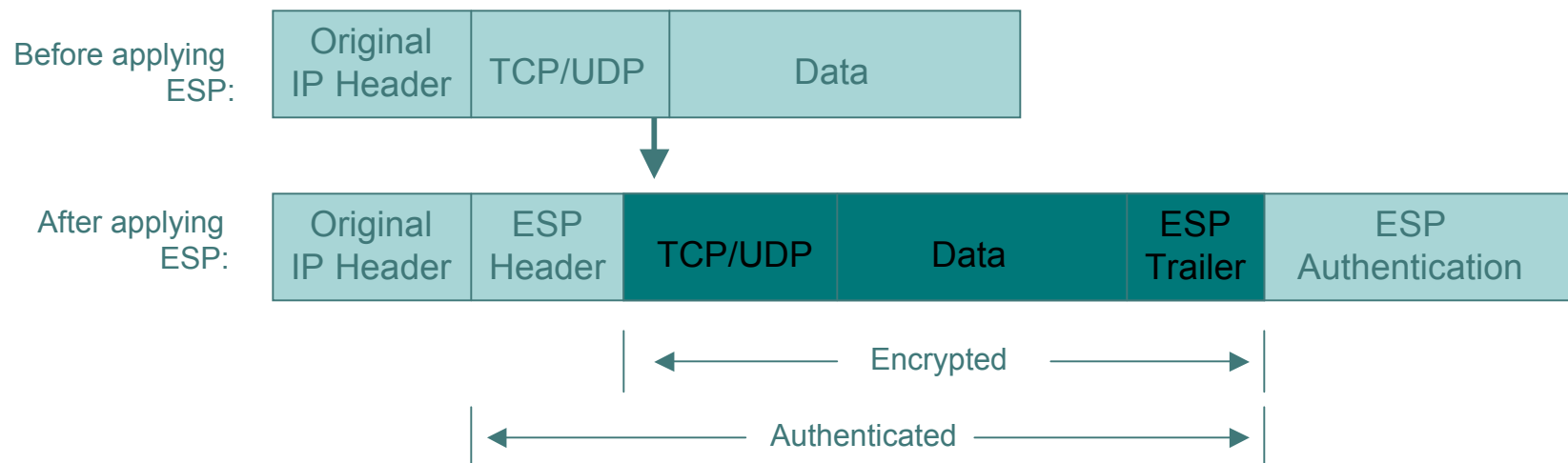
Authentication Header



- **ToS**
- **TTL**
- **Header Checksum**
- **Offset**
- **Flags**

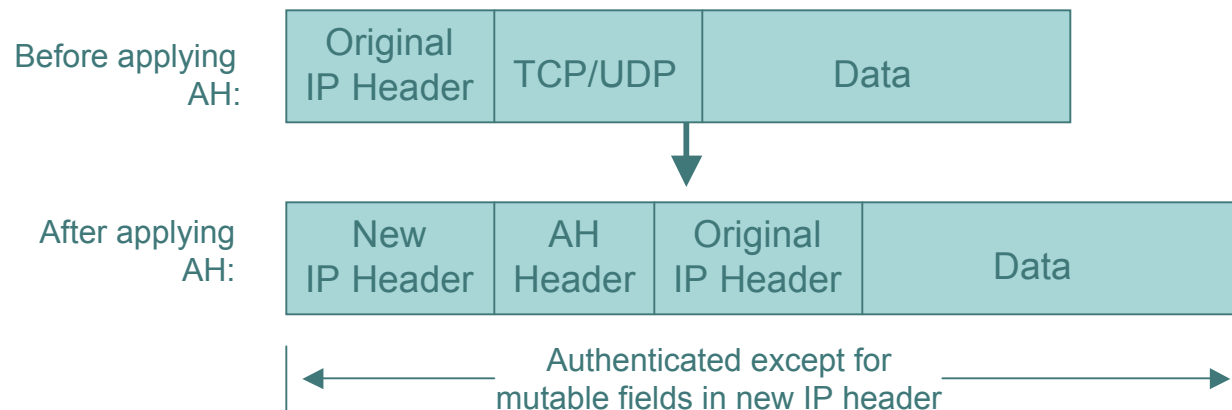
Packet Format Alteration for ESP Transport Mode

Encapsulating Security Payload



Packet Format Alteration for AH Tunnel Mode

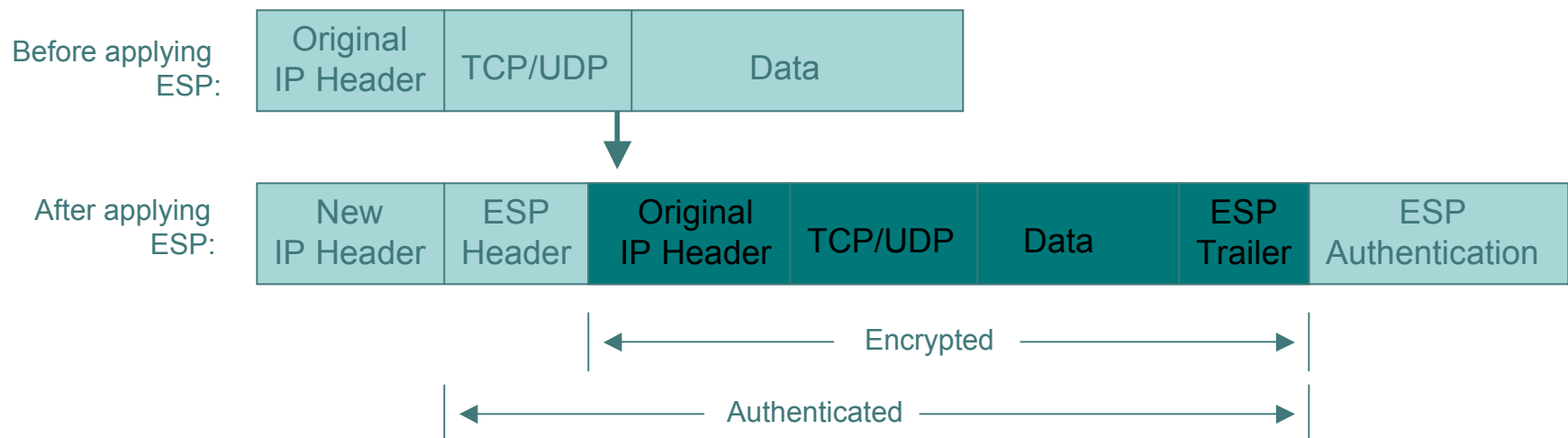
Authentication Header

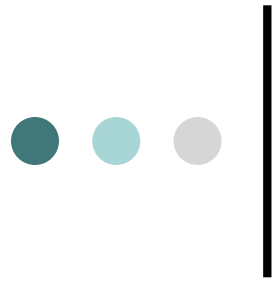


- ToS
- TTL
- Header Checksum
- Offset
- Flags

Packet Format Alteration for ESP Tunnel Mode

Encapsulating Security Payload





Internet Key Exchange (IKE)

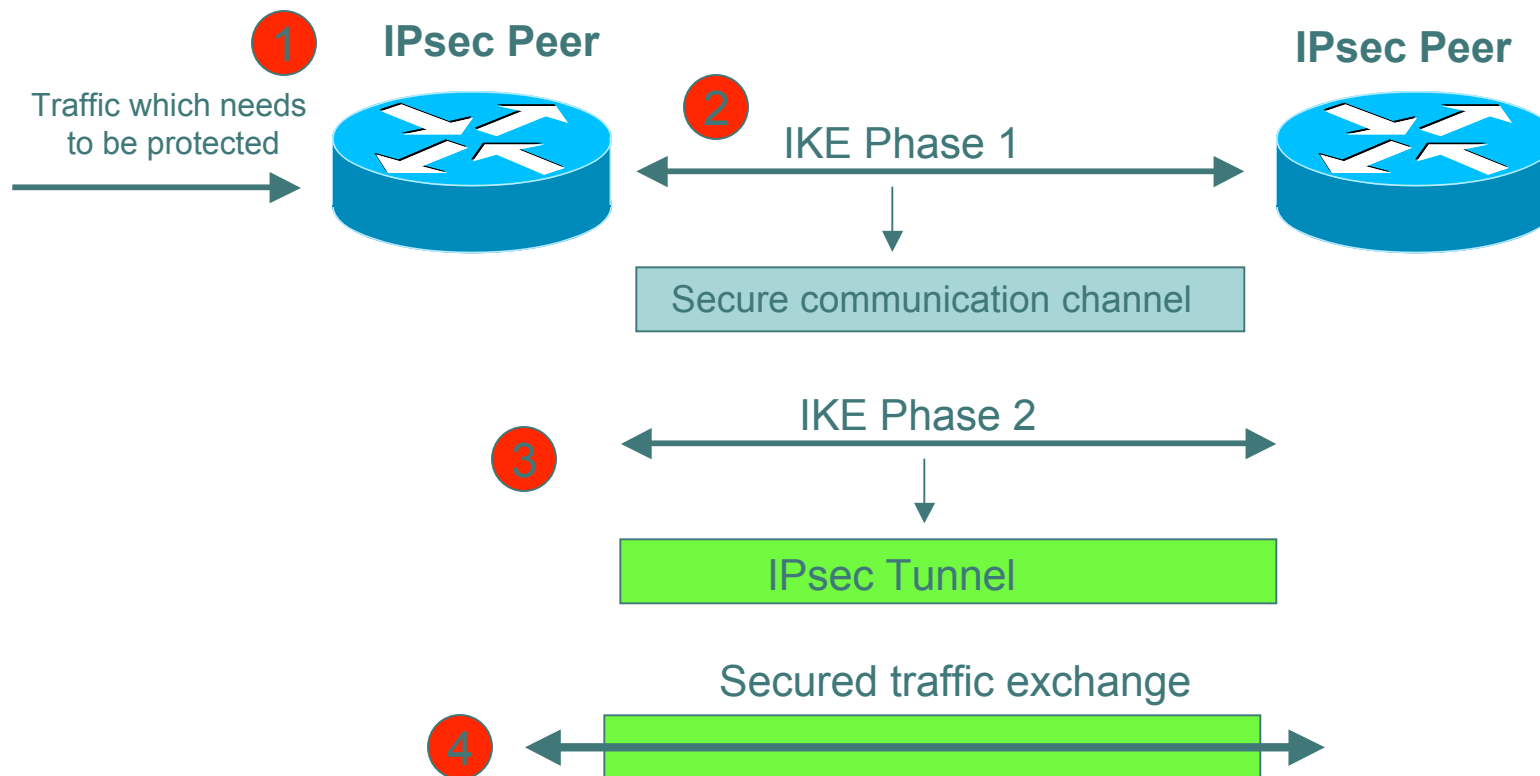
- Phase I

- Establish a secure channel (ISAKMP/IKE SA)
- Using either main mode or aggressive mode

- Phase II

- Establishes a secure channel between computers intended for the transmission of data (IPsec SA)
- Using quick mode

Overview of IKE

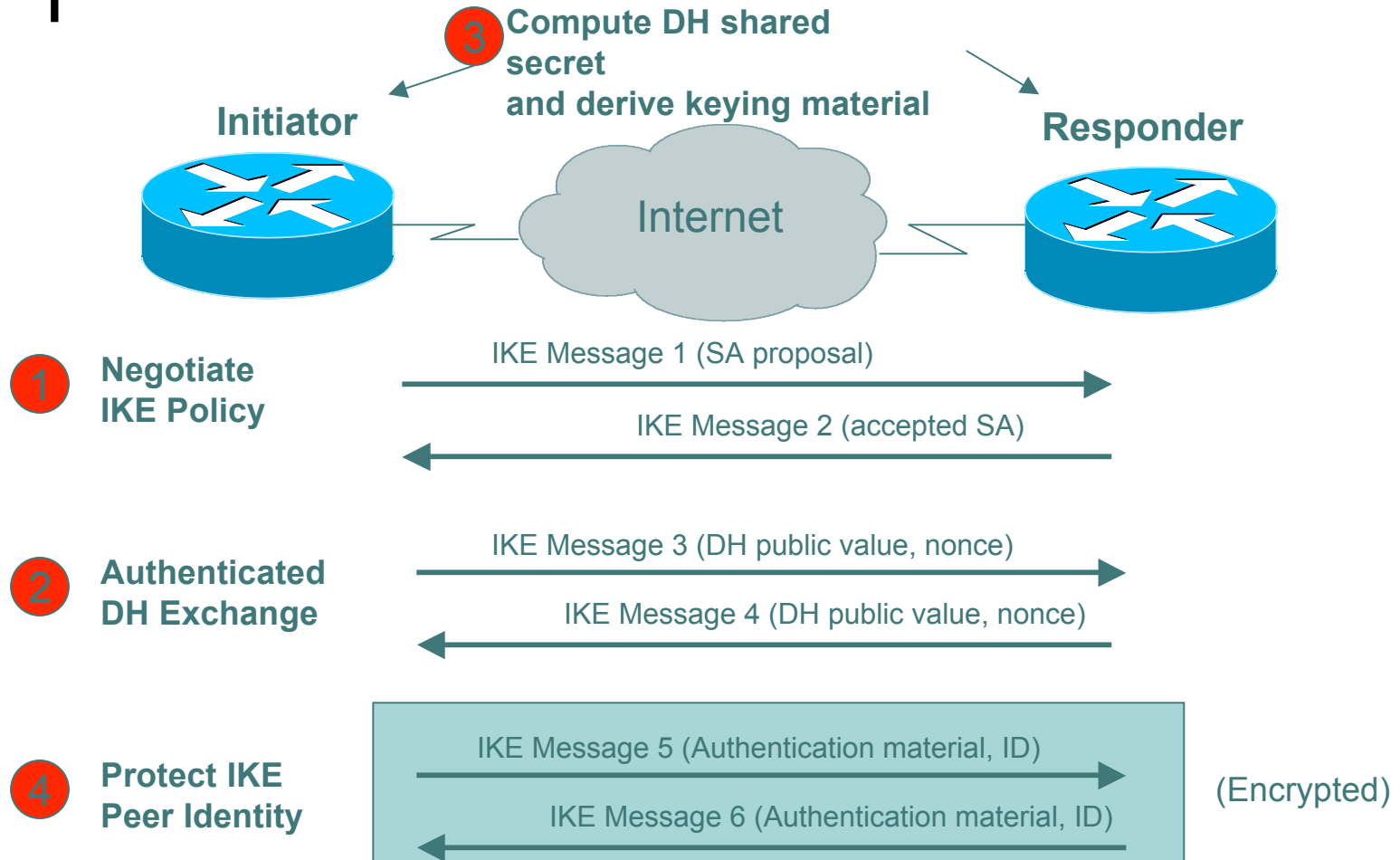




IKE Phase 1 Main Mode

- Main mode negotiates an ISAKMP SA which will be used to create IPsec SAs
- Three steps
 - SA negotiation (encryption algorithm, hash algorithm, authentication method, which DF group to use)
 - Do a Diffie-Hellman exchange
 - Provide authentication information
 - Authenticate the peer

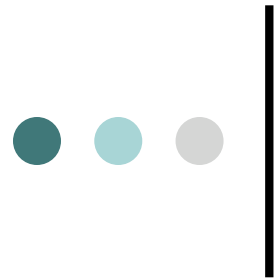
IKE Phase 1 Main Mode





What Is Diffie-Hellman?

- 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 3 different ways
 - Pre-shared secret
 - Nonce (RSA signature)
 - Digital signature



IKE Phase 1 Aggressive Mode

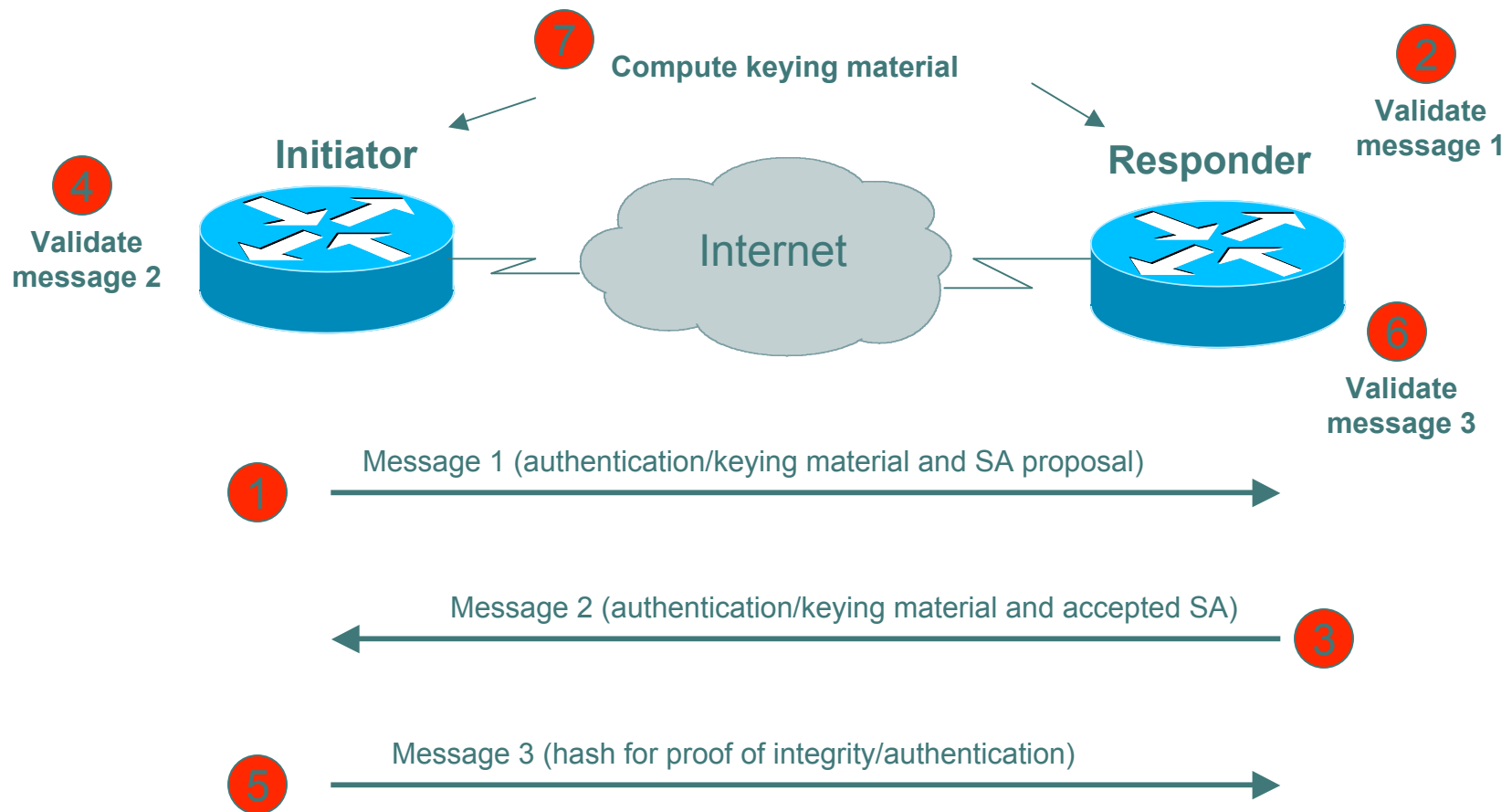
- Uses 3 (vs 6) messages to establish IKE SA
- No denial of service protection
- Does not have identity protection
- Optional exchange and not widely implemented



IKE Phase 2 Quick Mode

- All traffic is encrypted using the ISAKMP/IKE Security Association
- Each quick mode negotiation results in two IPsec Security Associations (one inbound, one outbound)
- Creates/refreshes keys

IKE Phase 2 Quick Mode





IKE Summary

- Negotiates parameters to establish and secure a channel between two peers
- Provides mutual authentication
- Establishes authenticated keys between peers
- Manages IPsec SAs
- Provides options for negotiation and SA establishment
- IKEv2
 - User authentication
 - Dynamic addressing
 - NAT traversal



Pretty Good IPsec Policy

- IKE Phase 1 (aka ISAKMP)
 - Main Mode
 - 3DES
 - SHA-1
 - DH Group 2 (MODP)
 - SA Lifetime (28880 seconds = 8 hours)
 - Pre-shared secret
- IKE Phase 2 (aka IPsec)
 - ESP Transport/Tunnel Mode
 - 3DES
 - SHA-1
 - PFS
 - DH Group 2 (MODP)
 - SA Lifetime (3600 seconds = 1 hour)



PFS- what is it?

- Perfect Forward Secrecy
- Doing new DH exchange to derive keying material

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



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



Configuring IPsec

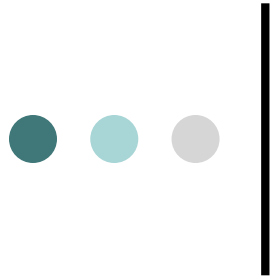
STEP 2 *Set the ISAKMP Identity*

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.



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

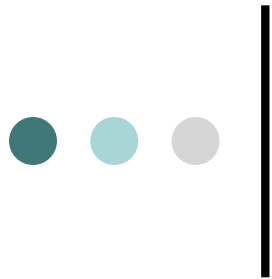


Configuring IPsec

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]
```

Configuring IPsec

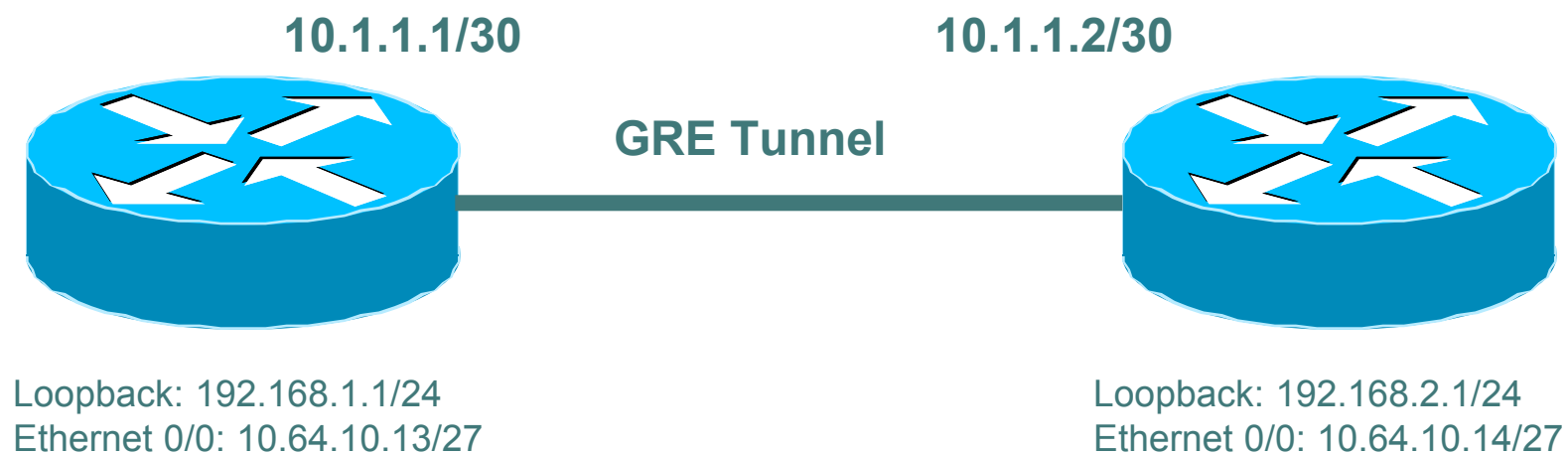
STEP 6 *Apply the IPsec Policy to an Interface*

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.

IPsec Example (EIGRP)





Sample Configuration (EIGRP)

!--- IKE policies

crypto isakmp policy 25

hash md5

authentication pre-share

crypto isakmp key cisco123 address 192.168.2.1

!--- IPSec policies

crypto ipsec transform-set eigrp-sec esp-des esp-md5-hmac

mode transport

crypto map GRE local-address Loopback0

crypto map GRE 50 ipsec-isakmp

set peer 192.168.2.1

set transform-set eigrp-sec

match address 101



Sample Configuration (EIGRP) cont.

```
interface Loopback0
ip address 192.168.1.1 255.255.255.0
!
interface Tunnel0
ip address 10.1.1.1 255.255.255.252
tunnel source Loopback0
tunnel destination 192.168.2.1
crypto map GRE
!
interface FastEthernet0/0
ip address 10.64.10.13 255.255.255.224
Crypto map GRE
!
router eigrp 10
network 10.1.1.0 0.0.0.3
network 172.16.1.0 0.0.0.255
network 192.168.1.0
!
access-list 101 permit gre host 192.168.1.1 host 192.168.2.1
```



Juniper BGP IPsec Example

```
[edit security ipsec]
+ proposal test-proposal {
+   protocol esp;
+   authentication-algorithm hmac-sha1-96;
+   encryption-algorithm 3des-cbc;
+   lifetime-seconds 3600;
+ }
+ policy test-ipsecwike {
+   perfect-forward-secrecy {
+     keys group2;
+   }
+   proposals test-proposal;
+ }
```

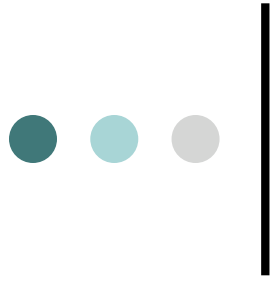
```
[edit security ipsec]
  security-association bgp-gw8-sa { ... }
+ security-association test-sa {
+   mode transport;
+   dynamic {
+     ipsec-policy test-ipsecwike }
+ }
```

```
[edit security]
+ ike {
+   proposal test-ike {
+     authentication-method pre-shared-keys;
+     dh-group group2;
+     authentication-algorithm sha1;
+     encryption-algorithm 3des-cbc;
+     lifetime-seconds 28880;
+   }
+   policy 198.6.255.32 {
+     mode main;
+     proposals test-ike;
+     pre-shared-key hexadecimal
"$9$QB21F9AuO1hyl0ONdwYoa9AtpRhWLx7dbA
pORSyW8NdbS2aiHm";
+   }
+ }
```



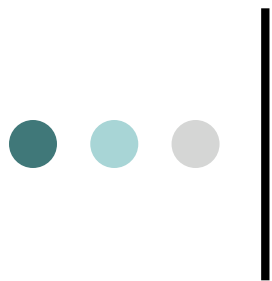
Logging

- Logging servers should be physically and logically secure
- Accept messages only from trusted hosts
- Encrypt log messages



Syslog

- Event logs created by syslog daemon
- Configured in */etc/syslog.conf*
- Usually logs stored in */var/log*
 - */var/log/secure*: successful and failed logins
 - */var/log/messages*: general messages
- Other information on logged in users can be found in */var/adm/*

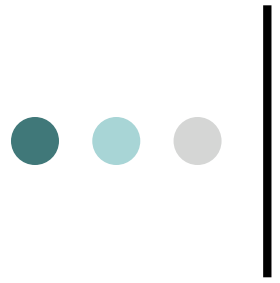


Checking UNIX Logs

```
cat <<! >checklist
/unix
/bin/*
/usr/bin/*
/usr/ucb/*
/etc/inetd.conf
/etc/passwd
!
eval ls -d 'cat checklist' >filelist.new
echo
echo "*** changes to the list of files checked:"
diff filelist filelist.new
echo
echo "*** changes in files:"
>>sum.new
for I in 'cat filelist'
do
echo "$I 'hash2.0 4 256 <$I'" >>sum.new
done
```

```
diff sum sum.new
```

Hash2.0 uses the 4-pass 256-bit output version of Merkle's snefru algorithm to compute checksum. Use hash2.0 since there exist tools to manipulate the output of the *sum* command.



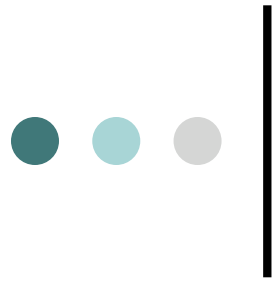
Syslog Alternatives

- Syslog-NG
 - <http://www.balabit.hu/products/syslog-ng/>
 - more extensive log message filtering
- Nsyslogd
 - <http://coombs.anu.edu.au/~avalon/nsyslog.html>
 - Supports SSL



Automated Log Analysis Tools

- SWATCH (The Simple Watcher)
 - <http://www.oit.ucsb.edu/~eta/swatch/>
 - need to write tools
- LogWatch
 - <http://www.logwatch.org/>
 - works right out of box but configuration changes require knowledge of PERL
- Checksyslog
 - <http://www.jammed.com/~jwa/hacks/security/checksyslog/checksyslog-doc.html>
 - very simplistic tool



Intrusion Detection Systems

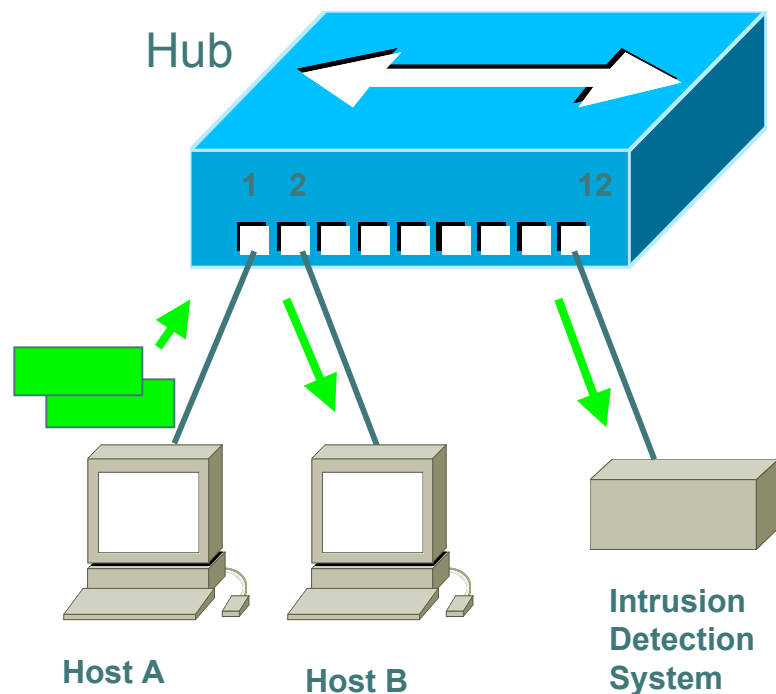
- Two methods of intrusion detection
 - Signature detection (pattern matching)
 - Low false positive / Detects only known attacks
 - Statistical anomaly detection
 - High false positive / Detects wider range of attacks



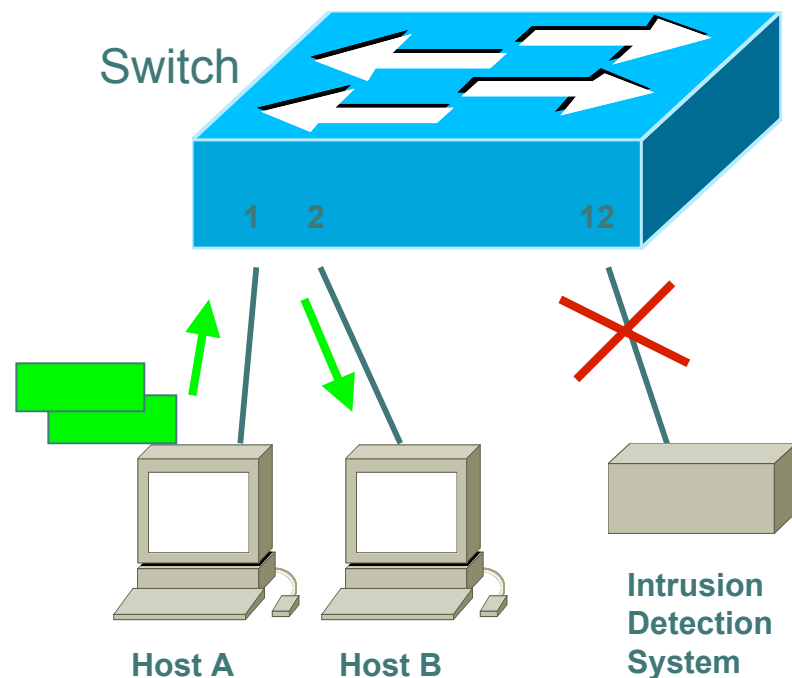
Signature vs Anomaly Detection

- Modeling signature detection is easy
 - If a known attack occurred in an observable area, then $p(\text{detection}) = 1$, else $p(\text{detection}) = 0$
- Modeling anomaly detection is more difficult
 - Noisy and/or unusual attacks are more likely seen
 - Denial of Service, port scans, unused services, etc.
 - Other types of attacks may be missed
 - Malformed web requests, some buffer overflows, etc.

Hub vs Switch with IDS

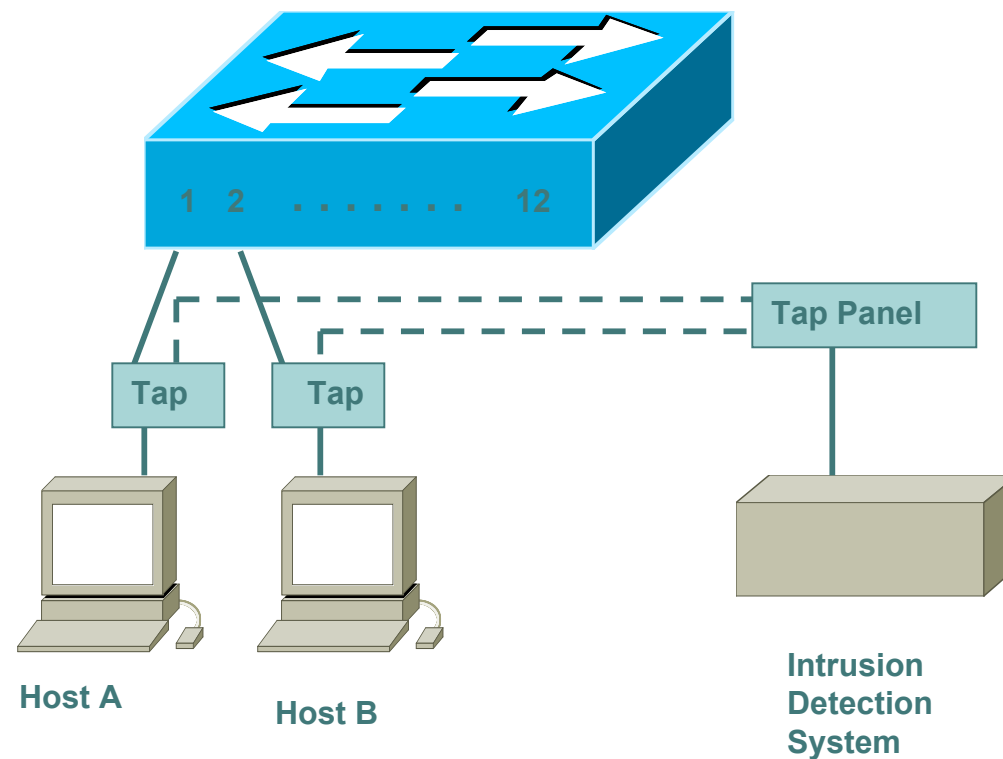


Traffic from host A to host B gets sent to **all** hub ports so the IDS can effectively monitor the traffic.



Traffic from host A to host B gets sent only to the port which connects host B and the IDS does not see any traffic.

Using NIDS with Cable Taps



● ● ● | Bypassing IDS Systems

- How varying TCP/IP stacks behave to slightly invalid input.
 - send TCP options, cause timeouts to occur for IP fragments or TCP segments
 - overlap fragments/segments
 - send slight wrong values in TCP flags or sequence numbers.

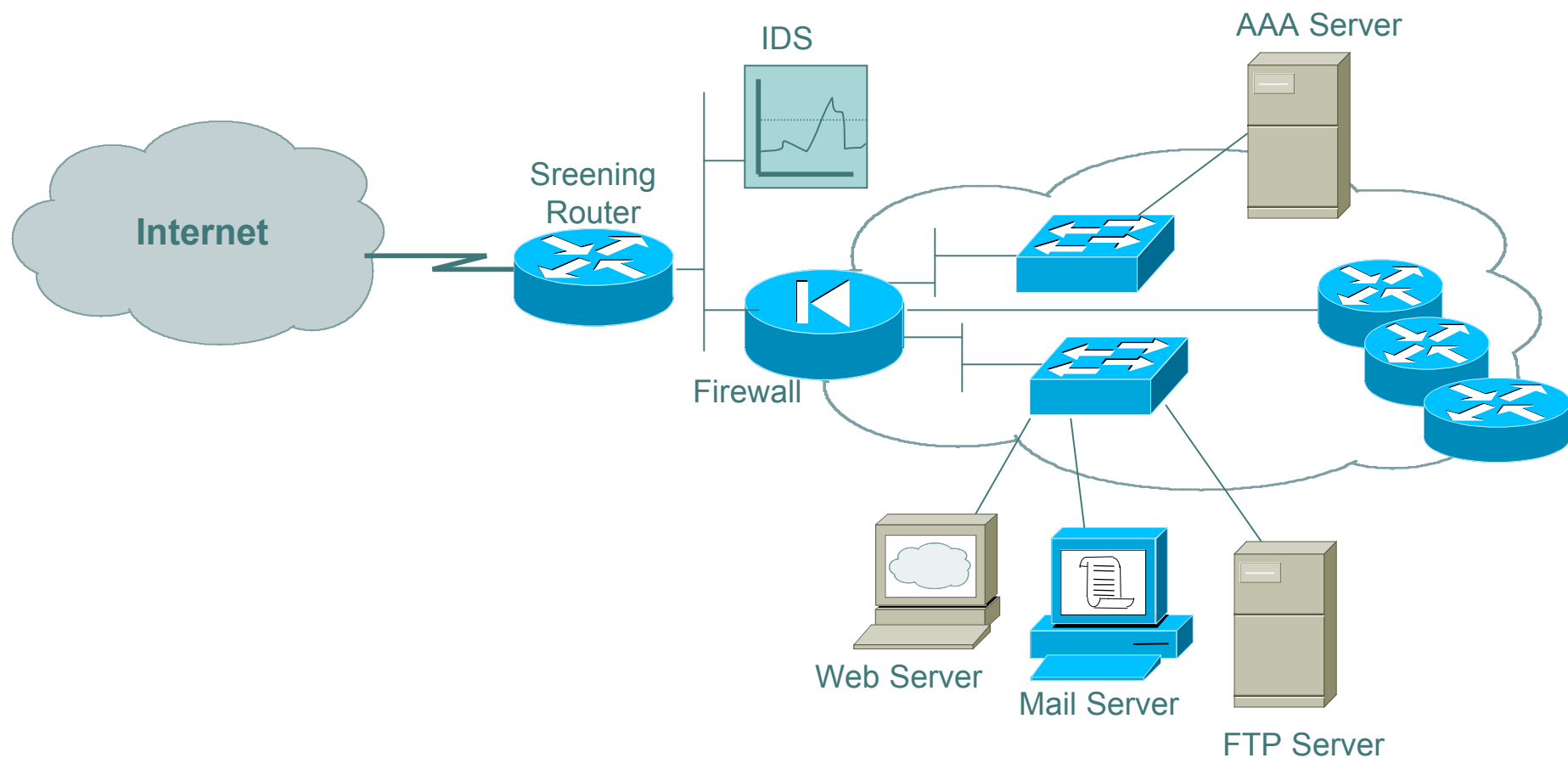
[If overlapping fragments are sent with different data, some systems prefer the data from the first fragment (WinNT, Solaris), whereas others keep the data from the last fragment (Linux, BSD). The NIDS has no way of knowing which the end-node will accept, and may guess wrong.]



IDS Limitations

- Vern Paxson's USENIX presentation in 1998 on 'Bro - A system for Detecting Network Intruders in real Time'
 - <ftp://ftp.ee.lbl.gov/papers/bro-usenix98-revised.ps.Z>
- Thomas H. Ptacek and Timothy N. Newsham., "Insertion, Evasion, And Denial Of Service: Eluding Network Intrusion Detection," Technical Report, Secure Networks, Inc., January 1998.
 - <http://citeseer.nj.nec.com/ptacek98insertion.html>

Using Network vs Host IDS





LAB Day 2

- Ingress / Egress Filtering
- IPsec configurations