# IPSec and SSL Virtual Private Networks

ITU/APNIC/MICT IPv6 Security Workshop

23rd – 27th May 2016

Bangkok

Last updated 29 June 2014

# Acknowledgment

- Content sourced from
  - Merike Kaeo of Double Shot Security
  - Contact: merike@doubleshotsecurity.com

# Virtual Private Networks

- Creates a secure tunnel over a public network
- Any VPN is not automagically secure
  - You need to add security functionality to create secure VPNs
  - That means using firewalls for access control
  - And probably IPsec or SSL/TLS for confidentiality and data origin authentication

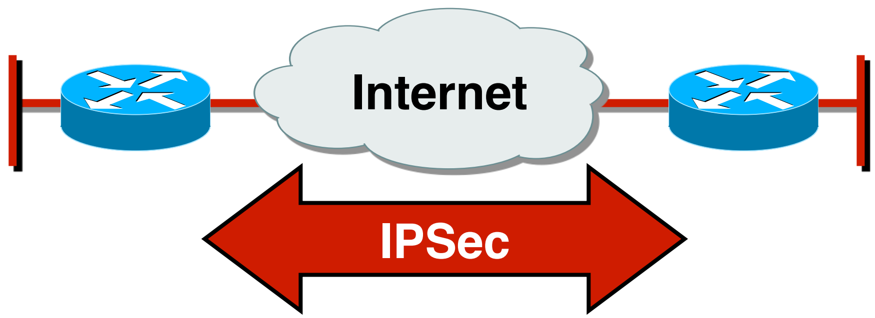
# VPN Protocols

- IPsec (Internet Protocol Security)
  - Open standard for VPN implementation
  - Operates on the network layer

# Other VPN Implementations

- MPLS VPN
  - Used for large and small enterprises
  - Pseudowire, VPLS, VPRN
- GRE Tunnel
  - Packet encapsulation protocol developed by Cisco
  - Not encrypted
  - Implemented with IPsec
- L2TP IPsec
  - Uses L2TP protocol
  - Usually implemented along with IPsec
  - IPsec provides the secure channel, while L2TP provides the tunnel

# What is IPSec?

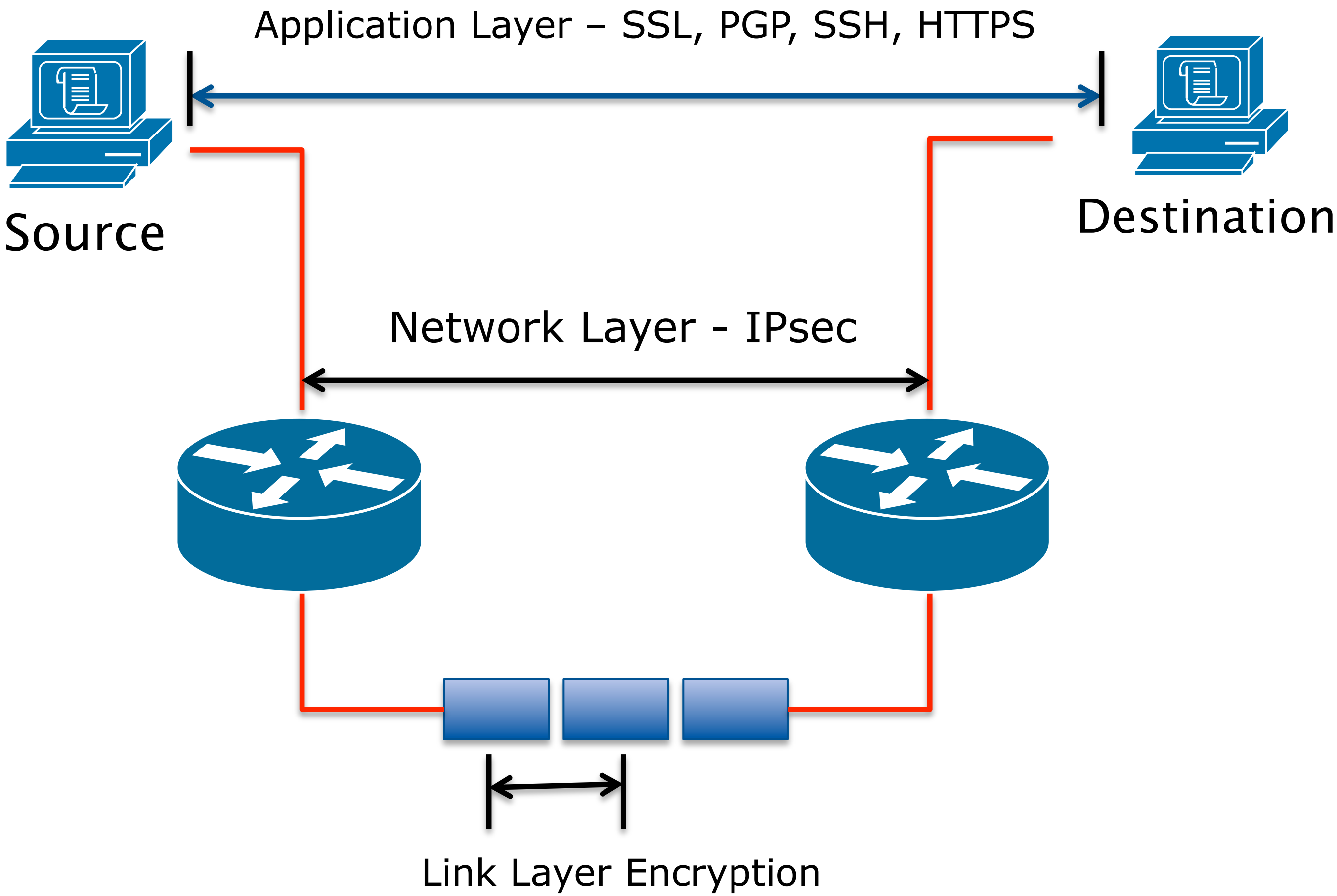



- IETF standard that enables encrypted communication between peers:
  - Consists of open standards for securing private communications
  - Network layer encryption ensuring data confidentiality, integrity, and authentication
  - Scales from small to very large networks

# What Does IPsec Provide ?

- Confidentiality….many algorithms to choose from
- 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
- 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

# Different Layers of Encryption



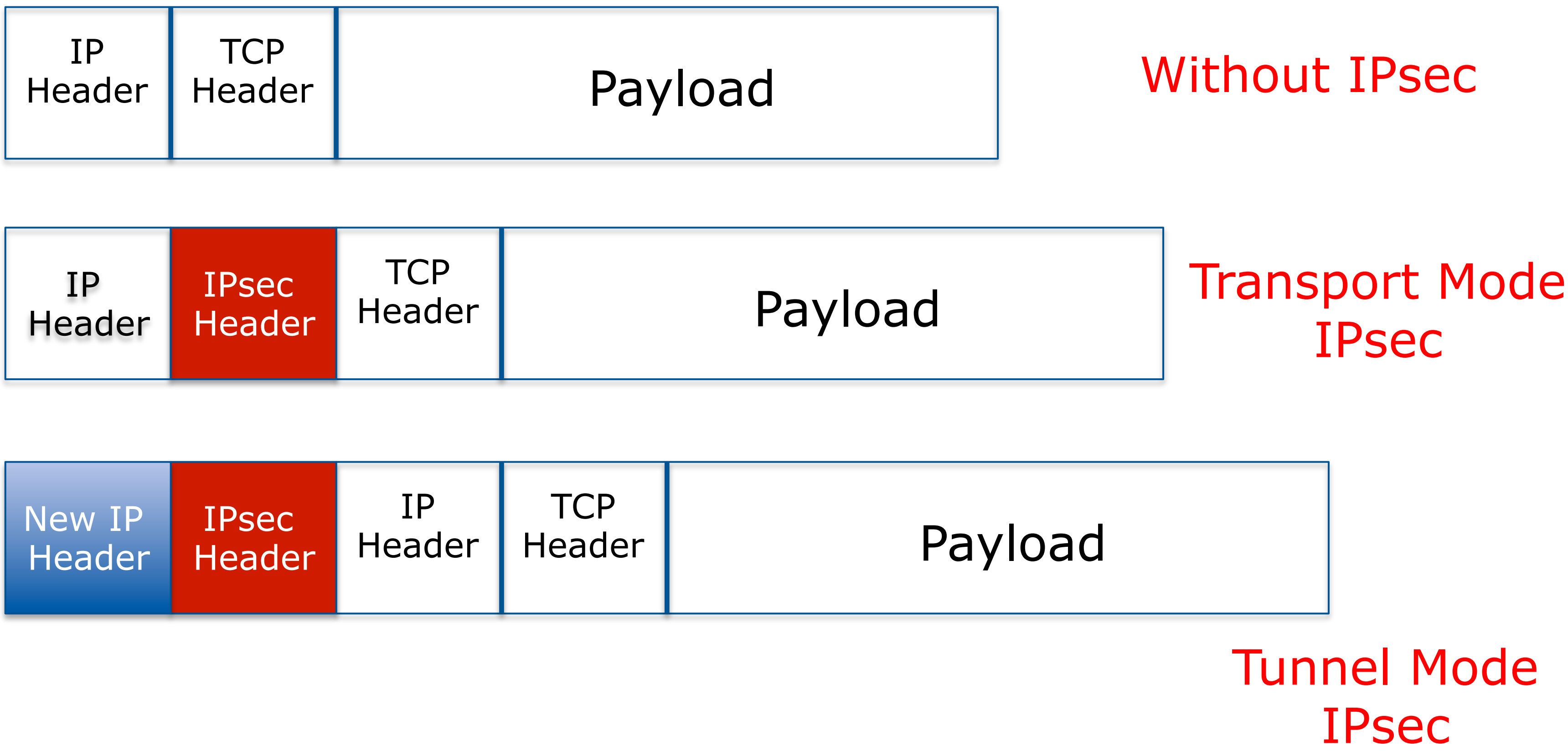
# Relevant Standard(s)

- IETF specific
  - rfc2409: IKEv1
  - rfc4301: IPsec Architecture (updated)
  - rfc4303: IPsec ESP (updated)
  - rfc4306: IKEv2
  - rfc4718: IKEv2 Clarifications
  - rfc4945: IPsec PKI Profile
- IPv6 and IPsec
  - rfc4294: IPv6 Node Requirements
  - rfc4552: Authentication/Confidentiality for OSPFv3
  - rfc4877: Mobile IPv6 Using IPsec (updated)
  - rfc4891: Using IPsec to secure IPv6-in-IPv4 Tunnels

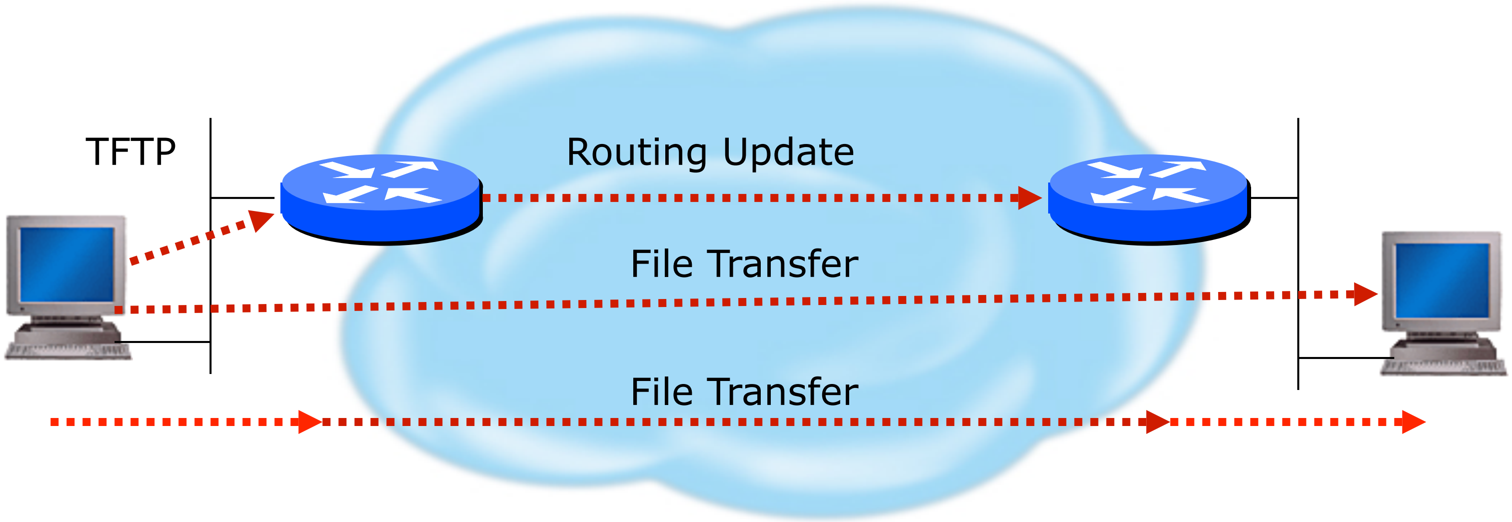
# IPsec Modes

- Tunnel Mode
  - Entire IP packet is encrypted and becomes the data component of a new (and larger) IP packet.
  - Frequently used in an IPsec site-to-site VPN
- Transport Mode
  - IPsec header is inserted into the IP packet
  - No new packet is created
  - Works well in networks where increasing a packet’s size could cause an issue
  - Frequently used for remote-access VPNs

# Tunnel vs. Transport Mode IPsec



# 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

# IPsec Components

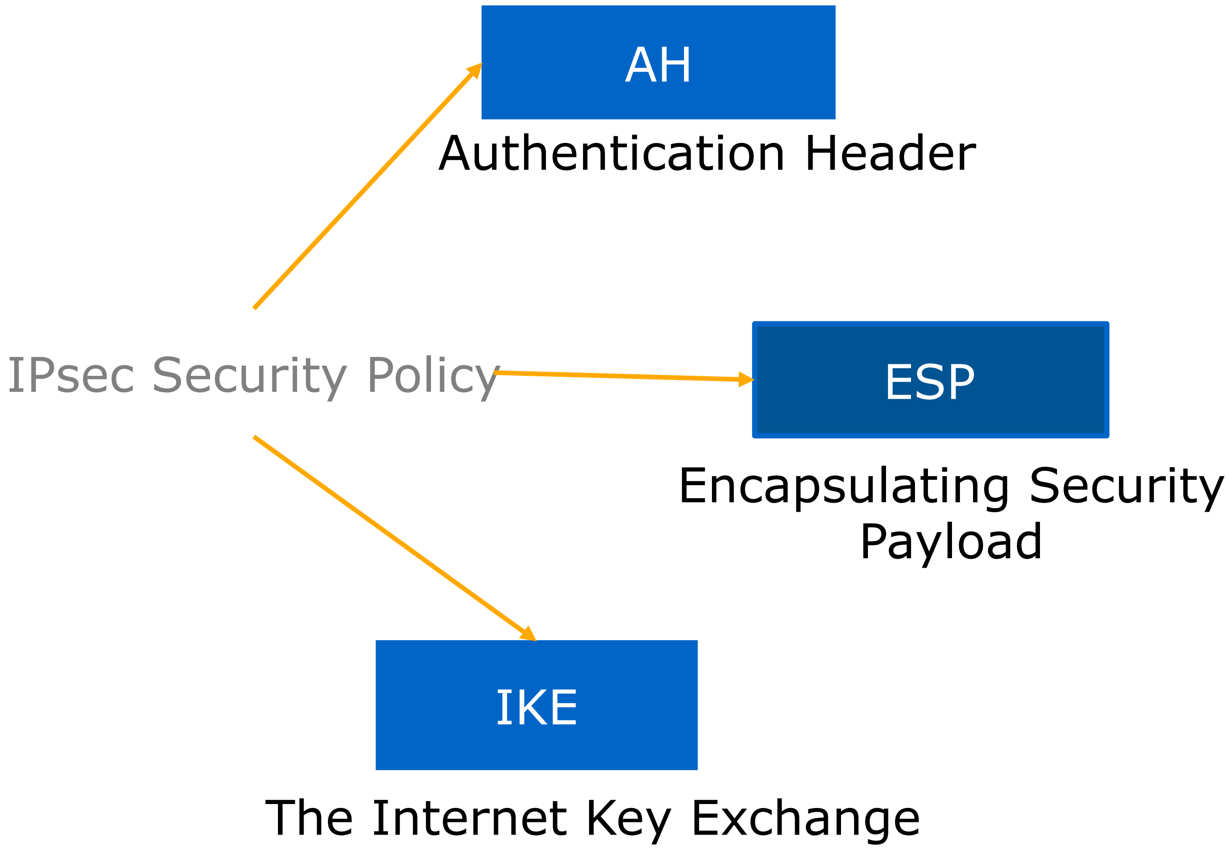
- AH (Authentication Header)
  - Authentication is applied to the entire packet, with the mutable fields in the IP header zeroed out
  - If both ESP and AH are applied to a packet, AH follows ESP
  - Standard requires HMAC-MD5-96 and HMAC-SHA1-96….older implementations also support keyed MD5

# IPsec Components

- 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

# IPsec Architecture



# Security Associations (SA)

- A collection of parameters required to establish a secure session
- Uniquely identified by three parameters consisting of
  - Security Parameter Index (SPI)
  - IP destination address
  - Security protocol (AH or ESP) identifier
- An SA is unidirectional
  - Two SAs required for a bidirectional communication
- A single SA can be used for AH or ESP, but not both
  - must create two (or more) SAs for each direction if using both AH and ESP

# Authentication Header (AH)

- Provides source authentication and data integrity
  - Protection against source spoofing and replay attacks
- Authentication is applied to the entire packet, with the mutable fields in the IP header zeroed out
- If both AH and ESP are applied to a packet, AH follows ESP
- Operates on top of IP using protocol 51
- In IPv4, AH protects the payload and all header fields except mutable fields and IP options (such as IPsec option)

# Encapsulating Security Payload (ESP)

- Uses IP protocol 50
- Provides all that is offered by AH, plus data confidentiality
  - It uses symmetric key encryption
- 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

# Internet Key Exchange (IKE)

- "An IPsec component used for performing mutual authentication and establishing and maintaining Security Associations." (RFC 5996)
- Typically used for establishing IPsec sessions
- A key exchange mechanism
- Five variations of an IKE negotiation:
  - Two modes (aggressive and main modes)
  - Three authentication methods (pre-shared, public key encryption, and public key signature)
- Uses UDP port 500

# IKE Modes

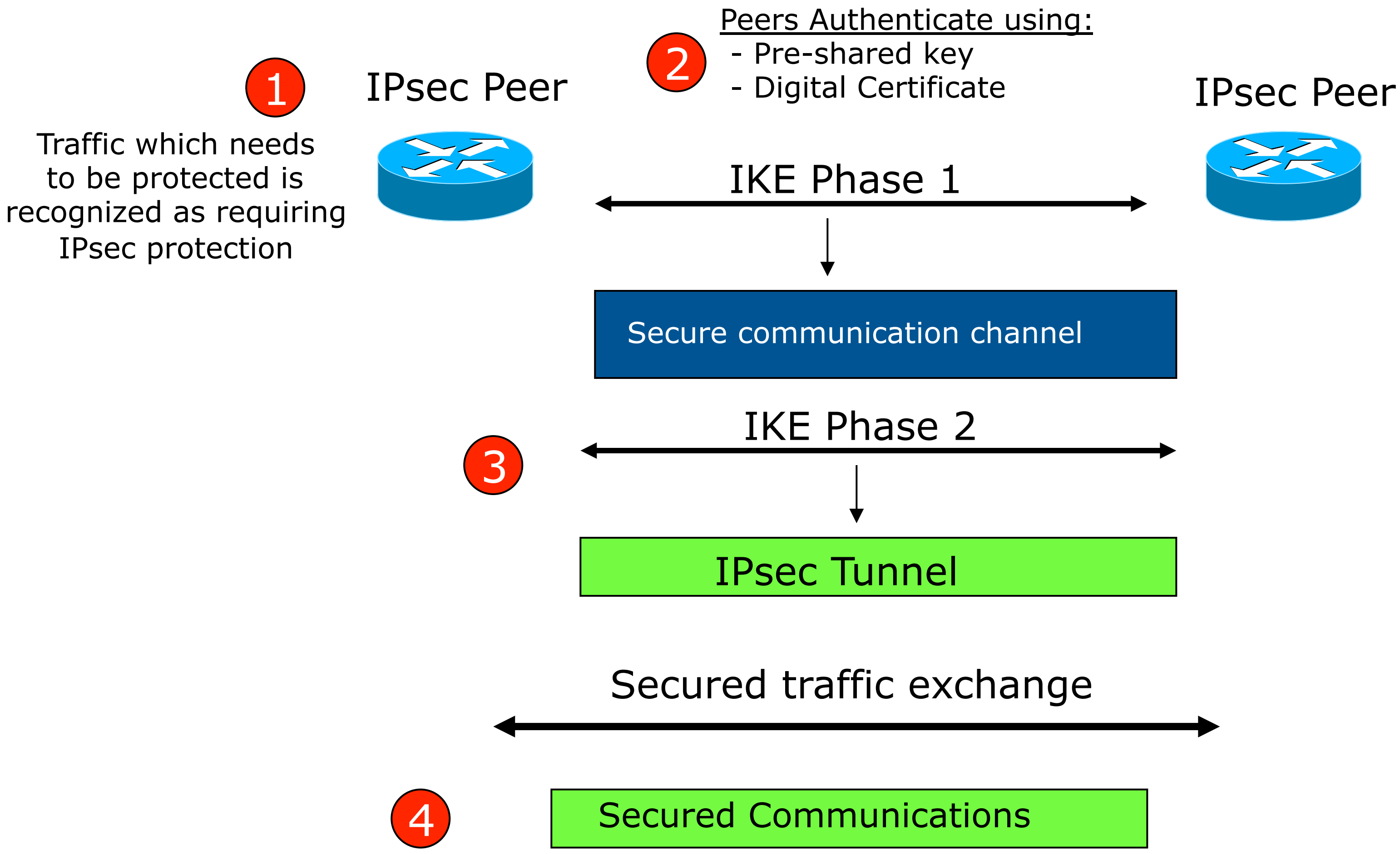
| Mode | Description |
|---|---|
| Main Mode | 1. Three exchanges of information between IPsec peers.<br>2. Initiator sends one or more proposals to the other peer (responder), responder selects a proposal<br>3. Diffie-Hellman (DH) key exchange<br>4. Establish ISAKMP session |
| Aggressive Mode | 1. Achieves same result as main mode using only 3 packets<br>2. First packet sent by initiator containing all info to establish SA<br>3. Second packet by responder with all security parameters selected<br>4. Third packet finalizes authentication of the ISAKMP session |
| Quick Mode | 1. Negotiates the parameters for the IPsec session.<br>2. Entire negotiation occurs within the protection of ISAKMP session |

# Internet Key Exchange (IKE)

- Phase I
  - Establish a secure channel (ISAKMP SA)
  - Using either main mode or aggressive mode
  - Authenticate computer identity using certificates or pre-shared secret

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

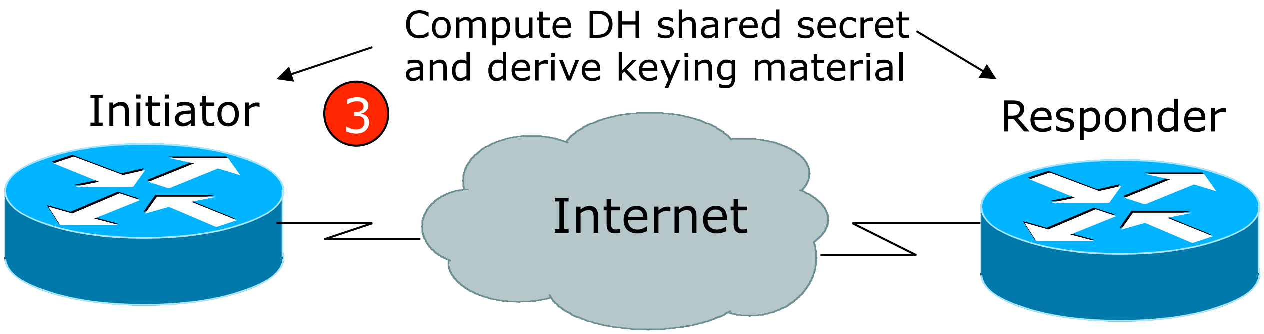
# IPsec with IKE



# 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




1 **Negotiate IKE Policy**

- IKE Message 1 (SA proposal) →
- ← IKE Message 2 (accepted SA)

2 **Authenticated DH Exchange**

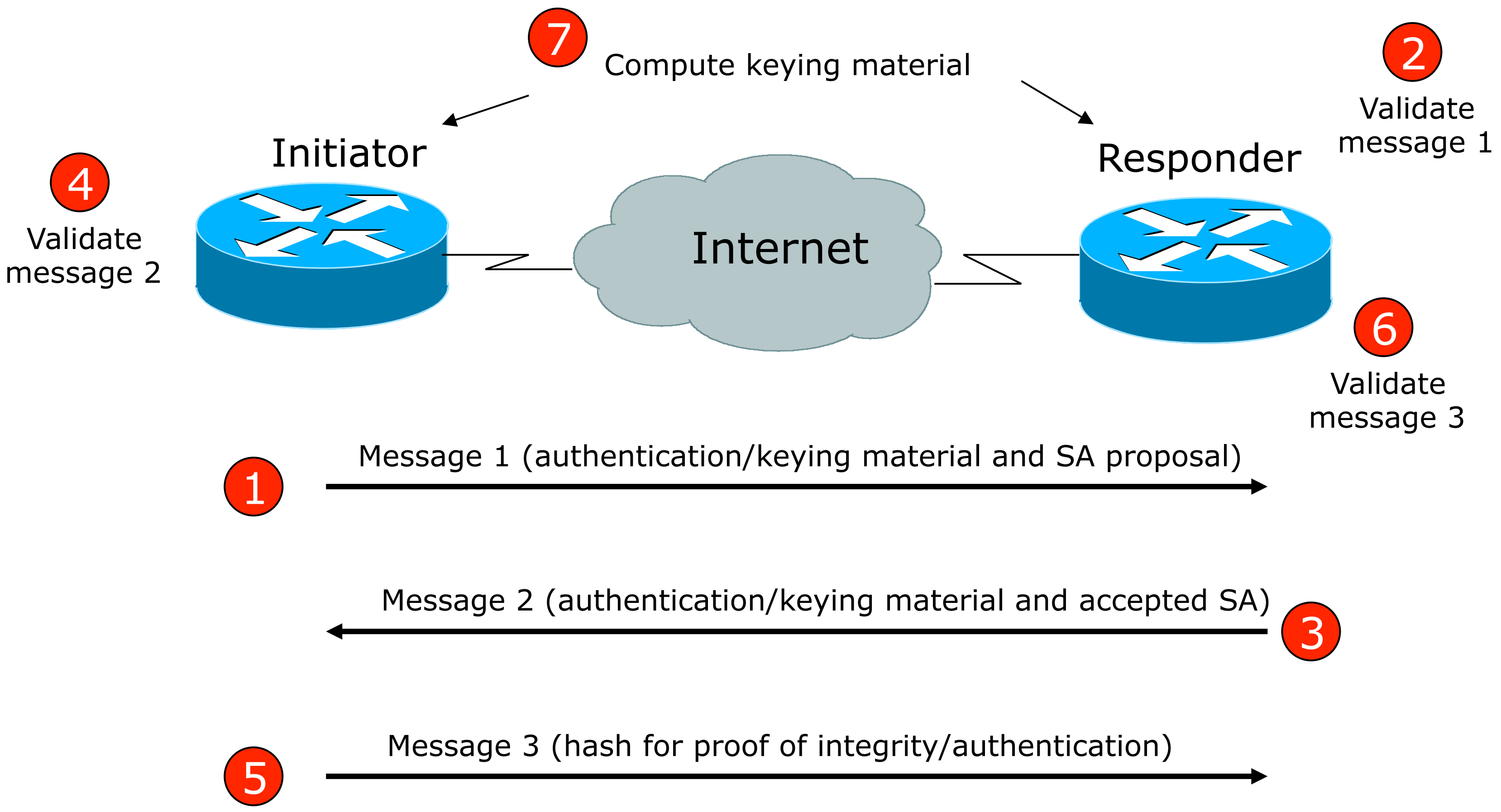
- IKE Message 3 (DH public value, nonce) →
- ← IKE Message 4 (DH public value, nonce)

4 **Protect IKE Peer Identity**

- IKE Message 5 (Authentication material, ID) →
- ← IKE Message 6 (Authentication material, ID)

(Encrypted)

# IKE Phase 2 Quick Mode



# IKE v2: Replacement for Current IKE Specification

- Feature Preservation
  - Most features and characteristics of baseline IKE v1 protocol are being preserved in v2
- Compilation of Features and Extensions
  - Quite a few features that were added on top of the baseline IKE protocol functionality in v1 are being reconciled into the mainline v2 framework
- Some New Features

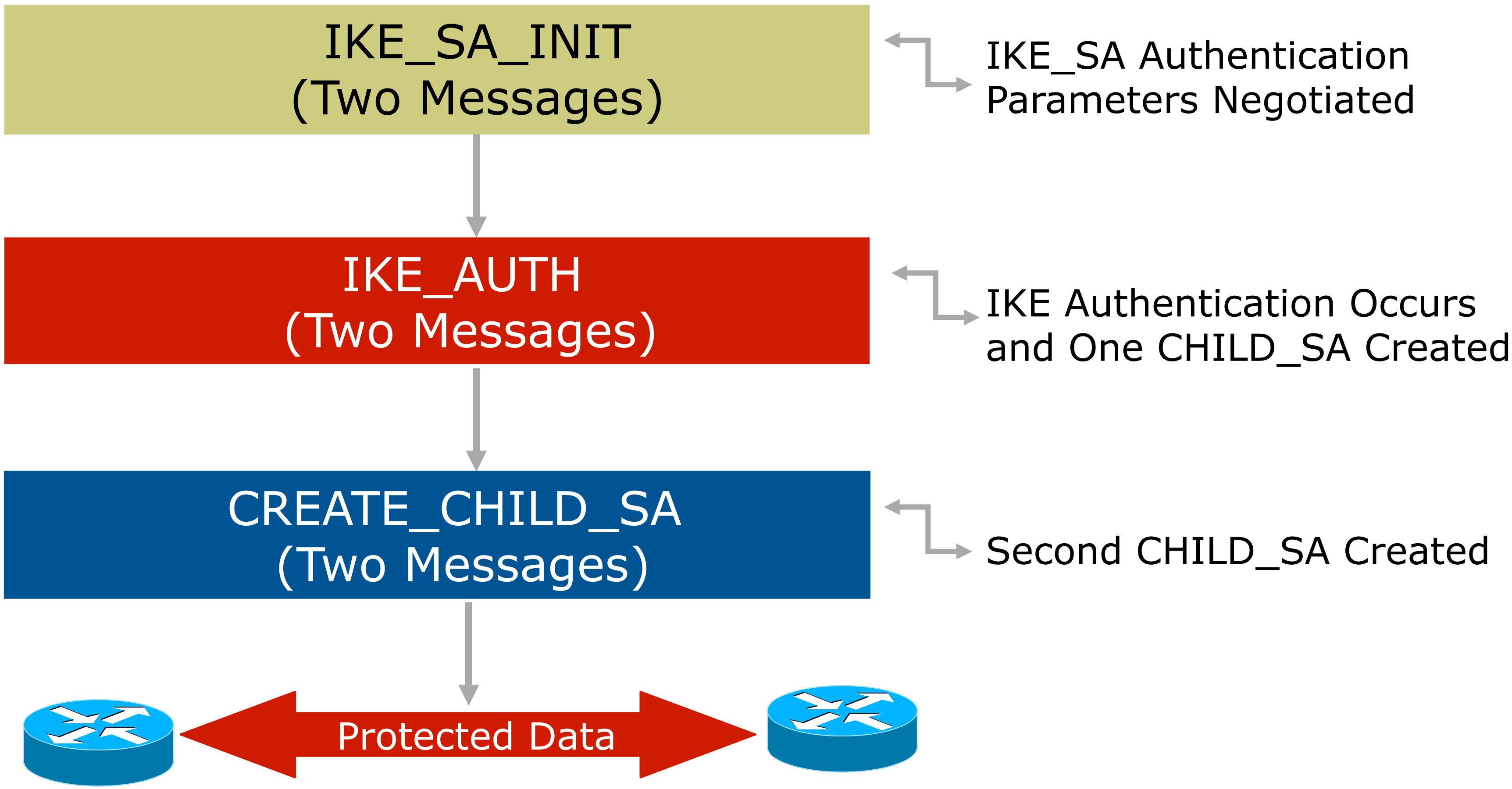
# IKE v2: What Is Not Changing

- Features in v1 that have been debated but are ultimately being preserved in v2
  - Most payloads reused
  - Use of nonces to ensure uniqueness of keys
- v1 extensions and enhancements being merged into mainline v2 specification
  - Use of a 'configuration payload' similar to MODECFG for address assignment
  - 'X-auth' type functionality retained through EAP
  - Use of NAT Discovery and NAT Traversal techniques

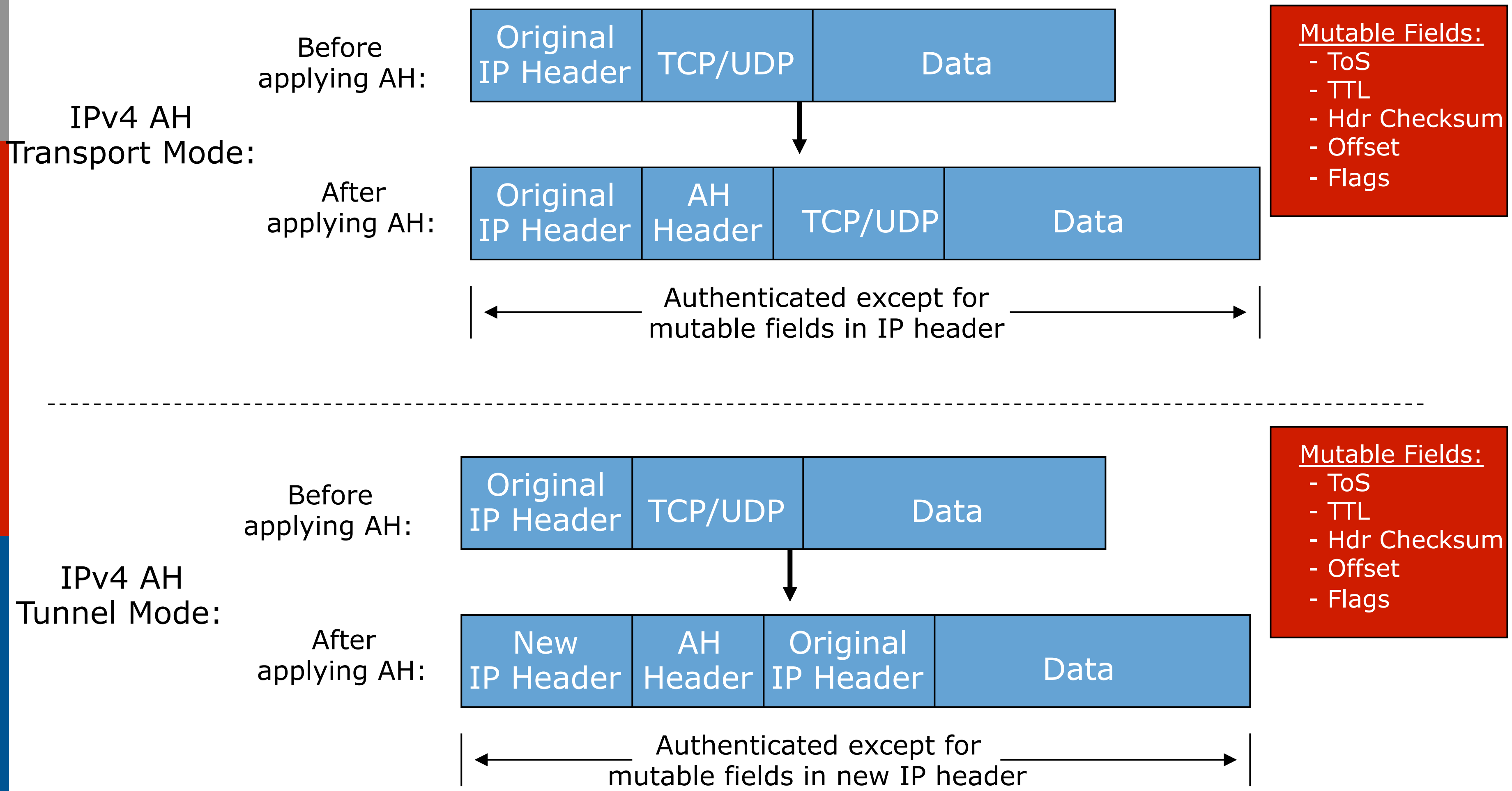
# IKE v2: What Is Changing

- Significant Changes Being to the Baseline Functionality of IKE
  - EAP adopted as the method to provide legacy authentication integration with IKE
  - Public signature keys and pre-shared keys, the only methods of IKE authentication
  - Use of ‘stateless cookie’ to avoid certain types of DOS attacks on IKE
  - Continuous phase of negotiation

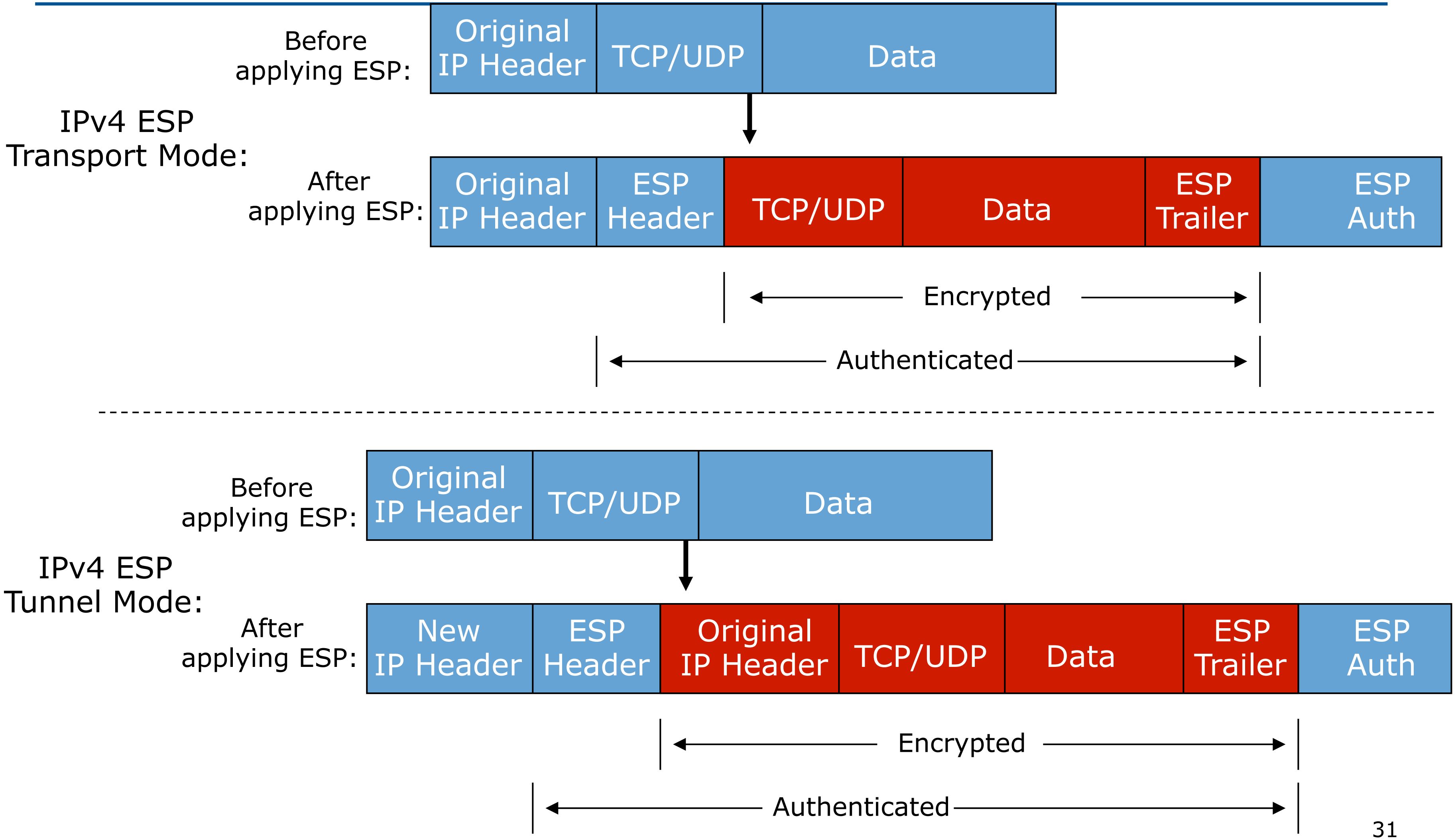
# How Does IKE v2 Work?



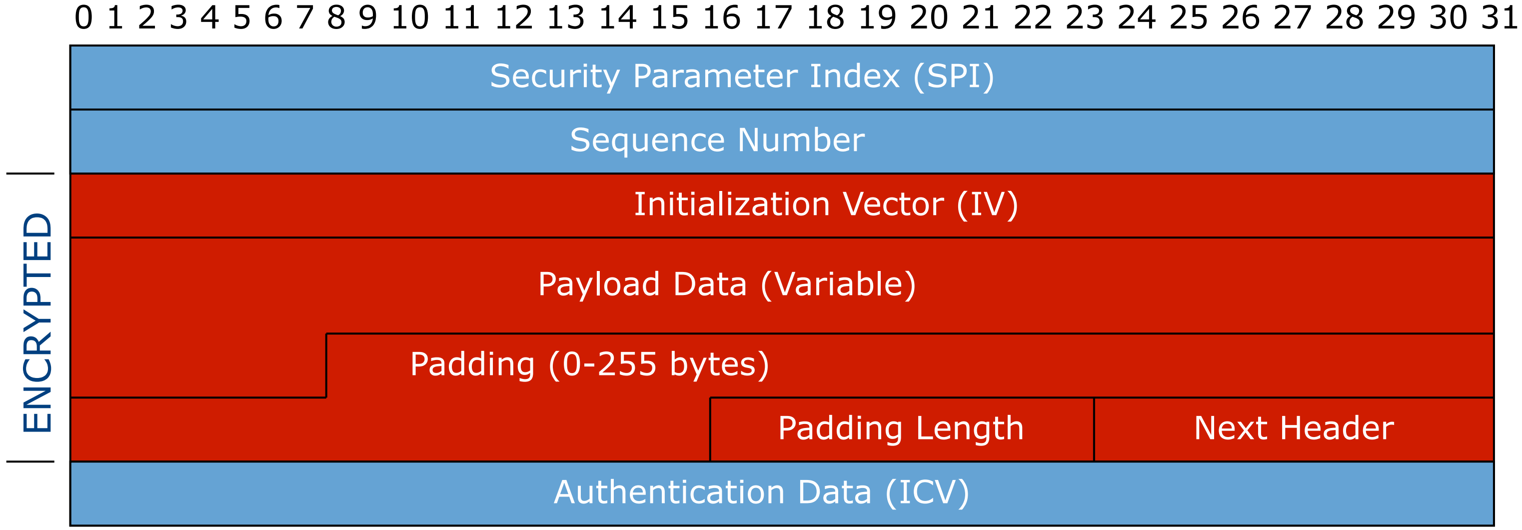
# IPv4 IPsec AH



# IPv4 IPsec ESP



# 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) |

# Considerations For Using IPsec

- Security Services
  - Data origin authentication
  - Data integrity
  - Replay protection
  - Confidentiality
- Size of network
- How trusted are end hosts – can a-priori communication policies be created?
- Vendor support
- What other mechanisms can accomplish similar attack risk mitigation

# Non-Vendor Specific Deployment Issues

- Historical Perception
  - Configuration nightmare
  - Not interoperable
- Performance Perception
  - Need empirical data
  - Where is the real performance hit?
- Standards Need Cohesion

# 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

# IPsec Concerns

- Are enough people aware that IKEv2 is not backwards compatible with IKEv1?
  - IKEv1 is used in most IPsec implementations
  - Will IKEv2 implementations first try IKEv2 and then revert to IKEv1?
- Is IPsec implemented for IPv6?
  - Some implementations ship IPv6 capable devices without IPsec capability and host requirements is changed from MUST to SHOULD implement
- OSPFv3
  - All vendors 'IF' they implement IPsec used AH
  - Latest standard to describe how to use IPsec says MUST use ESP w/Null encryption and MAY use AH

# IPsec Concerns (cont)

- What is transport mode interoperability status?
  - Will end user authentication be interoperable?
- PKI Issues
  - Which certificates do you trust?
  - How does IKEv1 and/or IKEv2 handle proposals with certificates?
  - Should common trusted roots be shipped by default?
  - Who is following and implementing pki4ipsec-ikecert-profile (rfc4945)
- Have mobility scenarios been tested?
  - Mobility standards rely heavily on IKEv2
- ESP – how to determine if ESP-Null vs Encrypted

# Default Issues

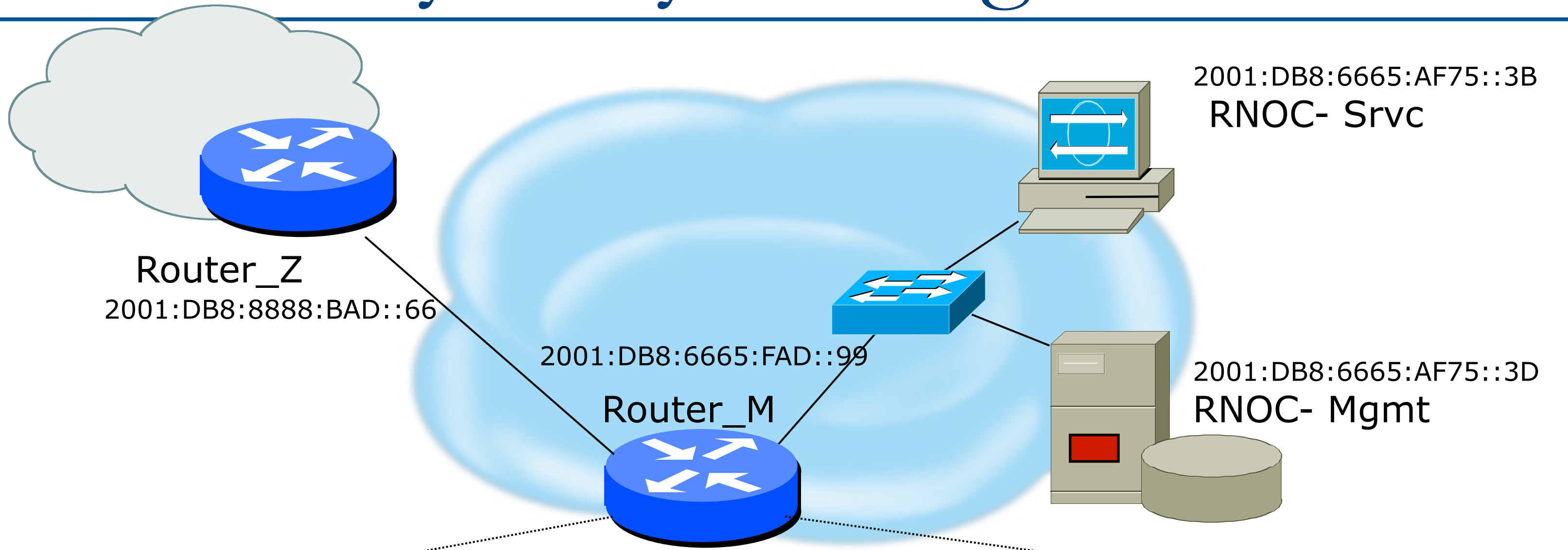
| Vendor A | Vendor B | Vendor C |
|---|---|---|
| IKE Phase 1 | IKE Phase 1 | IKE Phase 1 |
| SHA1 | MD5 | SHA1 |
| RSA-SIG | Pre-Share Key | Pre-Share Key |
| Group 1 | Group 5 | Group 2 |
| Lifetime 86400 Sec | Lifetime 86400 Sec | Lifetime 86400 Sec |
| Main Mode | Main Mode | Aggressive Mode |
| IKE Phase 2 | IKE Phase 2 | IKE Phase 2 |
| PFS | PFS | PFS |
| Group 1 | Group 5 | 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'

# Interoperable Defaults For SAs

- Security Association groups elements of a conversation together
  - 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  (8 hours = 480 min = 28800 sec)
  - SHA-2 (256 bit keys)
  - DH Group 14 (aka MODP# 14)

- IKE Phase 2 (aka IPsec SA or Quick Mode)
  - 3DES (AES-192 if both ends support it)
  - Lifetime (1 hour = 60 min = 3600 sec)
  - SHA-2 (256 bit keys)
  - PFS 2
  - DH Group 14 (aka MODP# 14)

# Sample Router Configuration

```
crypto isakmp policy 1
  authentication pre-share
  encryption aes
  hash sha
  group 5
crypto isakmp key Training123 address 172.16.11.66
!
crypto ipsec transform-set ESP-AES-SHA esp-aes esp-sha-hmac
!
crypto map LAB-VPN 10 ipsec-isakmp
  match address 101
  set transform-set ESP-AES-SHA
  set peer 172.16.11.66
```

**Phase 1 SA**

**Encryption and Authentication**

**Phase 2 SA**

# Sample Router Configuration

```
interface FastEthernet 0/1
 crypto map LAB-VPN
exit
!
access-list 101 permit ip 172.16.16.0 0.0.0.255 172.16.20.0 0.0.0.255
```

**Apply on outbound interface**

**Define interesting VPN traffic**

# Help With Configuring IPsec

- Documentation Profiles for IPsec Interoperability
  - http://www.vpnc.org/InteropProfiles/
- Documents for Cisco IPsec configuration:
  - http://www.cisco.com/en/US/tech/tk583/tk372/technologies_configuration_example09186a0080093f73.shtml
  - http://www.cisco.com/en/US/tech/tk583/tk372/technologies_configuration_example09186a0080093f86.shtml
- Document for Juniper IPsec configuration:
  - http://kb.juniper.net/InfoCenter/index?page=content&id=KB10128

# Capture: Telnet

| | | | | | |
|---|---|---|---|---|---|
| 8 | 3.113043 | Cisco_de:76:91 | Spanning-tree-(for-bridges) | STP | 60 Conf. Root = 32768/1/00:13:80:de:76:80 Cost = 0 Port = ( |
| 9 | 3.125855 | 192.168.1.1 | 172.16.2.1 | TELNET | 60 Telnet Data ... |
| 10 | 3.127649 | 172.16.2.1 | 192.168.1.1 | TELNET | 60 Telnet Data ... |
| 11 | 3.127651 | 172.16.2.1 | 192.168.1.1 | TCP | 60 [TCP Keep-Alive] telnet > 56784 [PSH, ACK] Seq=1 Ack=2 Wi |
| 12 | 3.279317 | 2001:df0:aa::5 | ff02::1:ff00:1 | ICMPv6 | 86 Neighbor Solicitation for 2001:df0:aa::1 from 00:0d:28:49 |
| 13 | 3.328358 | 192.168.1.1 | 172.16.2.1 | TCP | 60 56784 > telnet [ACK] Seq=2 Ack=2 Win=3987 Len=0 |
| 14 | 3.470005 | 192.168.1.1 | 172.16.2.1 | TELNET | 60 Telnet Data ... |
| 15 | 3.471802 | 172.16.2.1 | 192.168.1.1 | TELNET | 60 Telnet Data ... |
| 16 | 3.471804 | 172.16.2.1 | 192.168.1.1 | TCP | 60 [TCP Keep-Alive] telnet > 56784 [PSH, ACK] Seq=2 Ack=3 Wi |
| 17 | 3.672949 | 192.168.1.1 | 172.16.2.1 | TCP | 60 56784 > telnet [ACK] Seq=3 Ack=3 Win=3986 Len=0 |
| 18 | 3.854380 | 192.168.1.1 | 172.16.2.1 | TELNET | 60 Telnet Data ... |
| 19 | 3.856188 | 172.16.2.1 | 192.168.1.1 | TELNET | 60 Telnet Data ... |
| 20 | 3.856190 | 172.16.2.1 | 192.168.1.1 | TELNET | 60 [TCP Retransmission] Telnet Data ... |
| 21 | 3.978556 | 192.168.1.1 | 172.16.2.1 | TELNET | 60 Telnet Data ... |
| 22 | 3.980454 | 172.16.2.1 | 192.168.1.1 | TELNET | 60 Telnet Data ... |
| 23 | 3.980456 | 172.16.2.1 | 192.168.1.1 | TCP | 60 [TCP Keep-Alive] telnet > 56784 [PSH, ACK] Seq=6 Ack=5 Wi |
| 24 | 4.099046 | 192.168.1.1 | 172.16.2.1 | TELNET | 60 Telnet Data ... |
| 25 | 4.100949 | 172.16.2.1 | 192.168.1.1 | TELNET | 60 Telnet Data ... |
| 26 | 4.100950 | 172.16.2.1 | 192.168.1.1 | TCP | 60 [TCP Keep-Alive] telnet > 56784 [PSH, ACK] Seq=7 Ack=6 Wi |
| 27 | 4.243593 | 192.168.1.1 | 172.16.2.1 | TELNET | 60 Telnet Data ... |
| 28 | 4.245501 | 172.16.2.1 | 192.168.1.1 | TELNET | 60 Telnet Data ... |
| 29 | 4.245503 | 172.16.2.1 | 192.168.1.1 | TCP | 60 [TCP Keep-Alive] telnet > 56784 [PSH, ACK] Seq=8 Ack=7 Wi |

Follow TCP Stream

Stream Content

```
..... .....!...........................P....

User Access Verification

Password: .. ..!..!........ apn.......apnic2

router2>

router2>

router2>

router2>eenn

% No password set
router2>

router2>

router2>

router2>

router2>

router2>

router2>

router2>

router2>

router2>sshh  iipp  ??
```

```
router2>sshh  iipp  ??
  accounting              The active IP accounting database
  admission               Network Admission Control information
  aliases                 IP alias table
  arp                     IP ARP table
  as-path-access-list     List AS path access lists
  auth-proxy              Authentication Proxy information
  bgp                     BGP information
  cache                   IP fast-switching route cache
  casa                    display casa information
  cef                     Cisco Express Forwarding
  ddns                    Dynamic DNS
  dfp                     DFP information
  dhcp                    Show items in the DHCP database
  dvmrp                   DVMRP information
  eigrp                   IP-EIGRP show commands
  extcommunity-list       List extended-community list
  flow                    NetFlow switching
  helper-address          helper-address table
  host-list               Host list
  http                    HTTP information
  igmp                    IGMP information
  inspect                 CBAC (Context Based Access Control) information
 --More-- ..........        .........
router2>sh ip .. ... ... .iipp  iinntt.
router2>sh ip interface ??
  Async                 Async interface
  BVI                   Bridge-Group Virtual Interface
  CDMA-Ix               CDMA Ix interface
  CTunnel               CTunnel interface
  Dialer                Dialer interface
```

# Capture: Telnet + IPsec

| | | | | | | |
|---|---|---|---|---|---|---|
| 178 | 67.482683 | 2001:df0:aa::8 | ff02::1:ff00:2 | ICMPv6 | 86 | Neighbor Solicitation for 2001: |
| 179 | 67.594031 | 192.168.1.1 | 192.168.1.2 | ESP | 134 | ESP (SPI=0x7ea7f8ed) |
| 180 | 67.601908 | 192.168.1.2 | 192.168.1.1 | ESP | 118 | ESP (SPI=0x742f79b4) |
| 181 | 67.601910 | 192.168.1.2 | 192.168.1.1 | ESP | 118 | ESP (SPI=0x742f79b4) |
| 182 | 67.605809 | 192.168.1.1 | 192.168.1.2 | ESP | 118 | ESP (SPI=0x7ea7f8ed) |
| 183 | 67.626089 | 192.168.1.2 | 192.168.1.1 | ESP | 134 | ESP (SPI=0x742f79b4) |
| 184 | 67.626091 | 192.168.1.2 | 192.168.1.1 | ESP | 134 | ESP (SPI=0x742f79b4) |
| 185 | 67.627695 | 192.168.1.2 | 192.168.1.1 | ESP | 166 | ESP (SPI=0x742f79b4) |
| 186 | 67.627697 | 192.168.1.2 | 192.168.1.1 | ESP | 166 | ESP (SPI=0x742f79b4) |
| 187 | 67.631728 | 192.168.1.1 | 192.168.1.2 | ESP | 118 | ESP (SPI=0x7ea7f8ed) |
| 188 | 67.632884 | 192.168.1.1 | 192.168.1.2 | ESP | 118 | ESP (SPI=0x7ea7f8ed) |
| 189 | 67.751716 | 192.168.1.1 | 192.168.1.2 | ESP | 150 | ESP (SPI=0x7ea7f8ed) |
| 190 | 67.765217 | 192.168.1.2 | 192.168.1.1 | ESP | 118 | ESP (SPI=0x742f79b4) |
| 191 | 67.765219 | 192.168.1.2 | 192.168.1.1 | ESP | 118 | ESP (SPI=0x742f79b4) |
| 192 | 67.766634 | 192.168.1.2 | 192.168.1.1 | ESP | 118 | ESP (SPI=0x742f79b4) |
| 193 | 67.766636 | 192.168.1.2 | 192.168.1.1 | ESP | 118 | ESP (SPI=0x742f79b4) |
| 194 | 67.768056 | 192.168.1.2 | 192.168.1.1 | ESP | 118 | ESP (SPI=0x742f79b4) |
| 195 | 67.768058 | 192.168.1.2 | 192.168.1.1 | ESP | 118 | ESP (SPI=0x742f79b4) |
| 196 | 67.769385 | 192.168.1.2 | 192.168.1.1 | ESP | 118 | ESP (SPI=0x742f79b4) |
| 197 | 67.769387 | 192.168.1.2 | 192.168.1.1 | ESP | 118 | ESP (SPI=0x742f79b4) |
| 198 | 67.770803 | 192.168.1.2 | 192.168.1.1 | ESP | 118 | ESP (SPI=0x742f79b4) |
| 199 | 67.770804 | 192.168.1.2 | 192.168.1.1 | ESP | 118 | ESP (SPI=0x742f79b4) |
| 200 | 67.770963 | 192.168.1.1 | 192.168.1.2 | ESP | 134 | ESP (SPI=0x7ea7f8ed) |

# SSL/TLS

- Most widely-used protocol for security
- Encrypts the segments of network connections above the Transport Layer
- SSL and TLS
  - SSL v3.0 specified in an I-D in 1996 (draft-freier-ssl-version3-02.txt)
  - TLS v1.0 specified in RFC 2246 in 1999
  - TLS v1.0 = SSL v3.1 ≈ SSL v3.0
  - TLS v1.1 in 2006
  - TLS v1.2 in 2008
- Goals of protocol
  - Secure communication between applications
  - Data encryption
  - Server authentication
  - Message integrity
  - Client authentication (optional)

# Some Applications Using TLS/SSL

- Securing WWW traffic (HTTPS)
- Browsers Apache
  - Apache_mod_ssl
- DNSSEC requires SSL
- Postfix, Sendmail, SMTP
- sTelnet
- OpenSSH
- SFTP
- SSL VPNs such as OpenVPN and OpenConnect
- VoIP and SIP signaling
- EAP-TLS for WiFi

http://www.openssl.org/related/apps.html

# Benefits of TLS

- Application-layer independent
  - can be implemented with any applications
  - a wide range of applications supporting it
- SSL makes use of both asymmetric and symmetric key cryptography.
  - performance reasons.
  - Only the initial "client key exchange message" is encrypted with asymmetric encryption.
  - Symmetric encryption is better in terms of performance/ speed

# Benefits of TLS (cont)

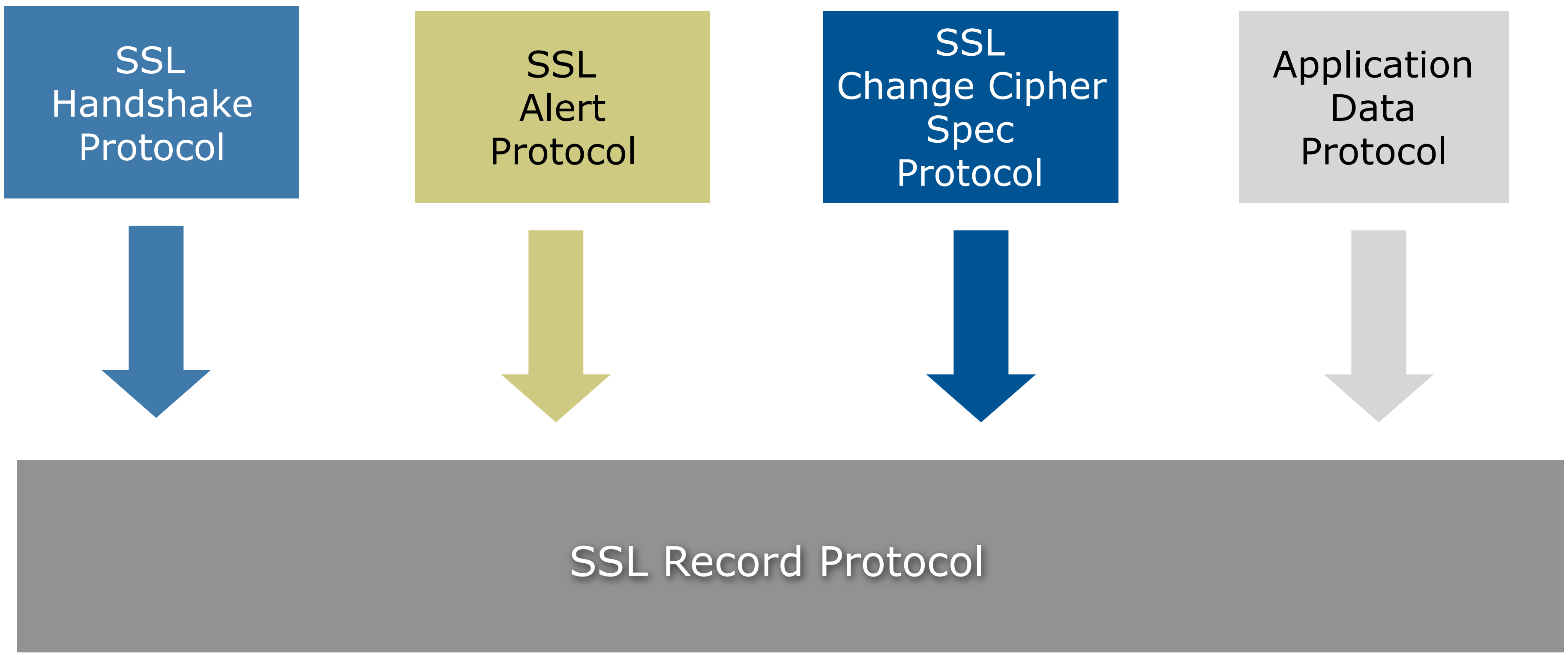
- Uses X.509 certificates
  - Certificates and Public Key Infrastructure
- SSL protocol layers comes on top of TCP (transport Layer), and is below application layer.
  - no network infrastructure changes are required to deploy SSL
- Each and every connection that's made, through SSL has got one session information.
  - Session can also be reused or resumed for other connections to the server

# SSL/TLS Properties

- Connection is private
  - Encryption is used after an initial handshake to define a secret key.
  - Symmetric cryptography used for data encryption
- 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 Building Blocks

SSL is a Combination of a Primary Record Protocol with Four 'Client' Protocols



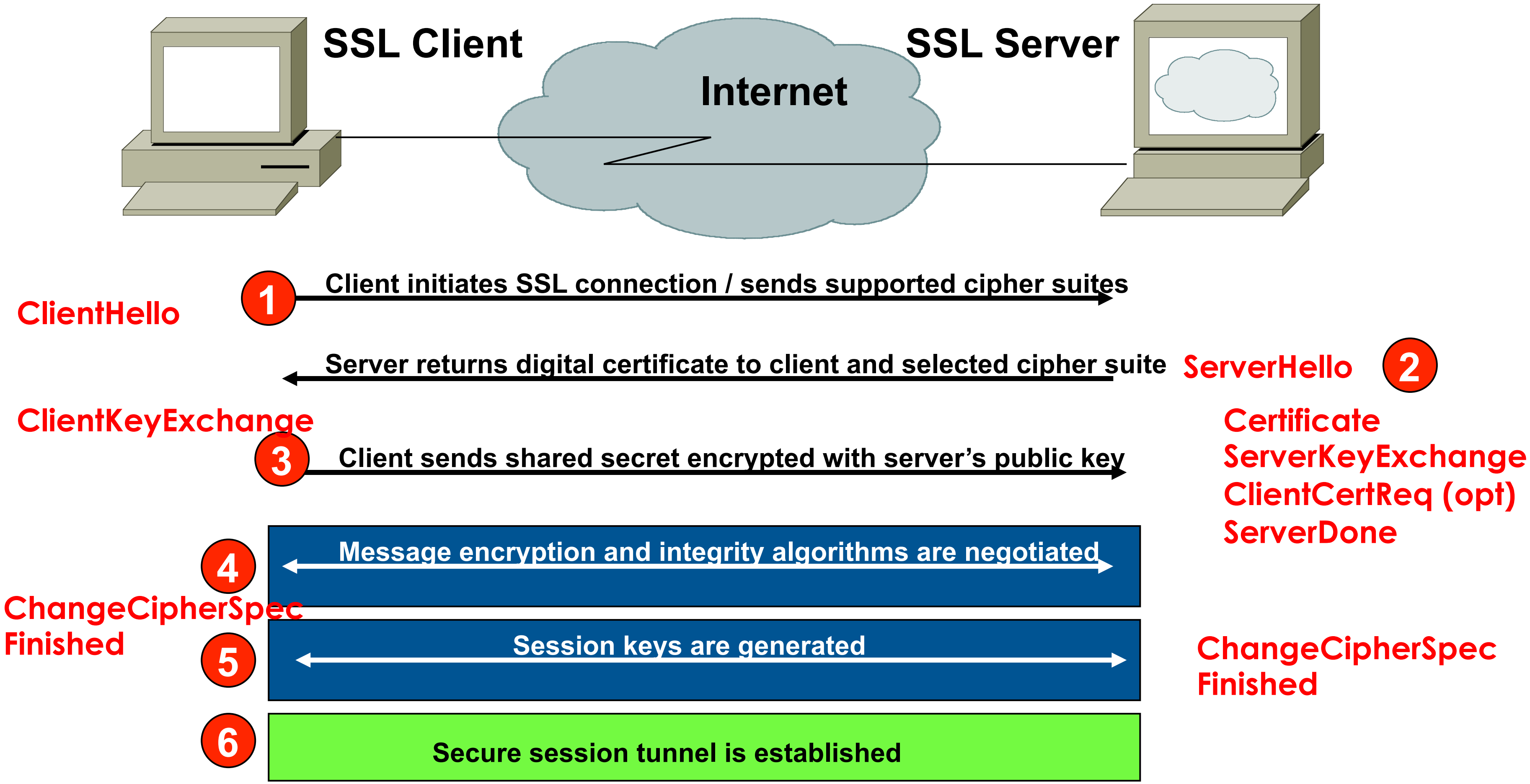
# SSL Protocol Building Block Functions

| | |
|---|---|
| SSL Handshake Protocol | Negotiates crypto algorithms and keys |
| SSL Alert Protocol | Indicates error or the end of a session |
| SSL Change Cipher Spec Protocol | Used to Signal Transition to New Cipher and Keys Generally Towards the End of a Handshake Negotiation |
| SSL Record Protocol | Indicates which encryption and integrity protection is applied to the data |

# SSL Record Layer

- Provides fragmentation, compression, integrity protection, and encryption for data objects exchanged between clients and servers
- Maintains a current and a pending connection state
- Upper Layer → TLSPlaintext → TLSCompressed → TLSCiphertext → (send to transport)

# The SSL Handshake Process




<- Application Data ->

SSL version, Random data (ClientHello.random), sessionID, cipher suits, compression algorithm

Client computes the premaster key

SSL version, Cipher suits, Random data ( ServerHello.random), sessionID

# 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 IANA Assigned Port #s

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

# Capture: SSL Decryption (easy)

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | 0.000037 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 38713 > https [ACK] Seq=1 Ack=1 Win=32767 Len=0 TSval=525562115 TSecr=525562115 |
| 4 | 0.000158 | 127.0.0.1 | 127.0.0.1 | SSLv2 | 171 | Client Hello |
| 5 | 0.000178 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | https > 38713 [ACK] Seq=1 Ack=106 Win=32767 Len=0 TSval=525562115 TSecr=525562115 |
| 6 | 0.002160 | 127.0.0.1 | 127.0.0.1 | SSLv3 | 995 | Server Hello, Certificate, Server Hello Done |
| 7 | 0.002609 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 38713 > https [ACK] Seq=106 Ack=930 Win=32767 Len=0 TSval=525562117 TSecr=525562117 |
| 8 | 2.808933 | 127.0.0.1 | 127.0.0.1 | SSLv3 | 278 | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message |
| 9 | 2.822770 | 127.0.0.1 | 127.0.0.1 | SSLv3 | 141 | Change Cipher Spec, Encrypted Handshake Message |
| 10 | 2.822809 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 38713 > https [ACK] Seq=318 Ack=1005 Win=32767 Len=0 TSval=525564938 TSecr=525564938 |
| 11 | 2.833071 | 127.0.0.1 | 127.0.0.1 | SSLv3 | 503 | Application Data |
| 12 | 2.873275 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | https > 38713 [ACK] Seq=1005 Ack=755 Win=32767 Len=0 TSval=525564989 TSecr=525564948 |
| 13 | 2.938485 | 127.0.0.1 | 127.0.0.1 | SSLv3 | 103 | Encrypted Handshake Message |
| 14 | 2.938750 | 127.0.0.1 | 127.0.0.1 | SSLv3 | 183 | Encrypted Handshake Message |
| 15 | 2.938761 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | https > 38713 [ACK] Seq=1042 Ack=872 Win=32767 Len=0 TSval=525565054 TSecr=525565054 |
| 16 | 2.938999 | 127.0.0.1 | 127.0.0.1 | SSLv3 | 1073 | Encrypted Handshake Message, Encrypted Handshake Message, Encrypted Handshake Message |
| 17 | 2.940026 | 127.0.0.1 | 127.0.0.1 | SSLv3 | 337 | Encrypted Handshake Message, Change Cipher Spec, Encrypted Handshake Message |
| 18 | 2.943406 | 127.0.0.1 | 127.0.0.1 | SSLv3 | 172 | Change Cipher Spec, Encrypted Handshake Message |
| 19 | 2.944825 | 127.0.0.1 | 127.0.0.1 | SSLv3 | 5756 | Application Data, Application Data |
| 20 | 2.944864 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 38713 > https [ACK] Seq=1143 Ack=7845 Win=32767 Len=0 TSval=525565060 TSecr=525565059 |
| 21 | 2.964424 | 127.0.0.1 | 127.0.0.1 | SSLv3 | 471 | Application Data |
| 33 | 3.004256 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | https > 38713 [ACK] Seq=7845 Ack=1548 Win=32767 Len=0 TSval=525565120 TSecr=525565080 |

Using stolen key file

| | | | | | | |
|---|---|---|---|---|---|---|
| 25 | 2.964810 | 127.0.0.1 | 127.0.0.1 | SSLv3 | 186 | Client Hello |
| 26 | 2.964819 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | https > 38714 [ACK] Seq=1 Ack=121 Win=32767 Len=0 TSval=525565080 TSecr=525565080 |
| 27 | 2.992274 | 127.0.0.1 | 127.0.0.1 | SSLv3 | 220 | Server Hello, Change Cipher Spec, Finished |
| 28 | 2.992312 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 38714 > https [ACK] Seq=121 Ack=155 Win=32767 Len=0 TSval=525565108 TSecr=525565108 |
| 29 | 2.992855 | 127.0.0.1 | 127.0.0.1 | HTTP | 562 | GET /icons/debian/openlogo-25.jpg HTTP/1.1 |
| 30 | 2.993501 | 127.0.0.1 | 127.0.0.1 | HTTP | 596 | HTTP/1.1 404 Not Found (text/html) |
| 31 | 2.993840 | 127.0.0.1 | 127.0.0.1 | HTTP | 471 | GET /icons/apache_pb.png HTTP/1.1 |
| 32 | 2.994179 | 127.0.0.1 | 127.0.0.1 | HTTP | 1828 | HTTP/1.1 200 OK (PNG) |
| 33 | 3.004256 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | https > 38713 [ACK] Seq=7845 Ack=1548 Win=32767 Len=0 TSval=525565120 TSecr=525565080 |
| 34 | 3.033250 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 38714 > https [ACK] Seq=1022 Ack=2447 Win=32767 Len=0 TSval=525565149 TSecr=525565109 |
| 35 | 3.501643 | 127.0.0.1 | 127.0.0.1 | HTTP | 588 | HTTP/1.1 404 Not Found (text/html) |
| 36 | 3.507001 | 127.0.0.1 | 127.0.0.1 | HTTP | 439 | GET /favicon.ico HTTP/1.1 |
| 37 | 3.507541 | 127.0.0.1 | 127.0.0.1 | HTTP | 580 | HTTP/1.1 404 Not Found (text/html) |
| 38 | 3.507555 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 38714 > https [ACK] Seq=1395 Ack=2961 Win=32767 Len=0 TSval=525565623 TSecr=525565623 |
| 39 | 3.541174 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | 38713 > https [ACK] Seq=1548 Ack=8367 Win=32767 Len=0 TSval=525565657 TSecr=525565617 |
| 40 | 6.037880 | 127.0.0.1 | 127.0.0.1 | HTTP | 511 | GET /test HTTP/1.1 |
| 41 | 6.037932 | 127.0.0.1 | 127.0.0.1 | TCP | 66 | https > 38713 [ACK] Seq=8367 Ack=1993 Win=32767 Len=0 TSval=525568154 TSecr=525568154 |
| 42 | 6.041185 | 127.0.0.1 | 127.0.0.1 | HTTP | 644 | HTTP/1.1 301 Moved Permanently (text/html) |

# Attacks on SSL (a little harder…)

- BEAST Attack (2011)
  - Browser Exploit Against SSL/TLS
  - CBC vulnerability discovered in 2002
  - Fixed in TLS 1.1
- CRIME Attack (2012)
  - Compression Ratio Info-leak Made Easy
  - Exploit against TLS compression
  - ‘fixed’ by disabling TLS Compression
- BREACH Attack (2013)
  - Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext
  - Presented at BlackHat 2013 (Aug)
  - Attacks HTTP responses using HTTP Compression

# Encrypted Communications

- Use encrypted communications whenever you need to keep information confidential
- Verify via network sniffer (e.g. wireshark) that your communication is indeed encrypted
- An important aspect is credential management (creating, distributing, storing, revoking, renewing)
- Understand if/when credentials are lost that you may not be able to recover the data
- Have a plan in place in case you forget your password that protects your private keys

# Thank You. Questions?

# IPSec and SSL Virtual Private Networks

ITU/APNIC/MICT IPv6 Security Workshop

23rd – 27th May 2016

Bangkok