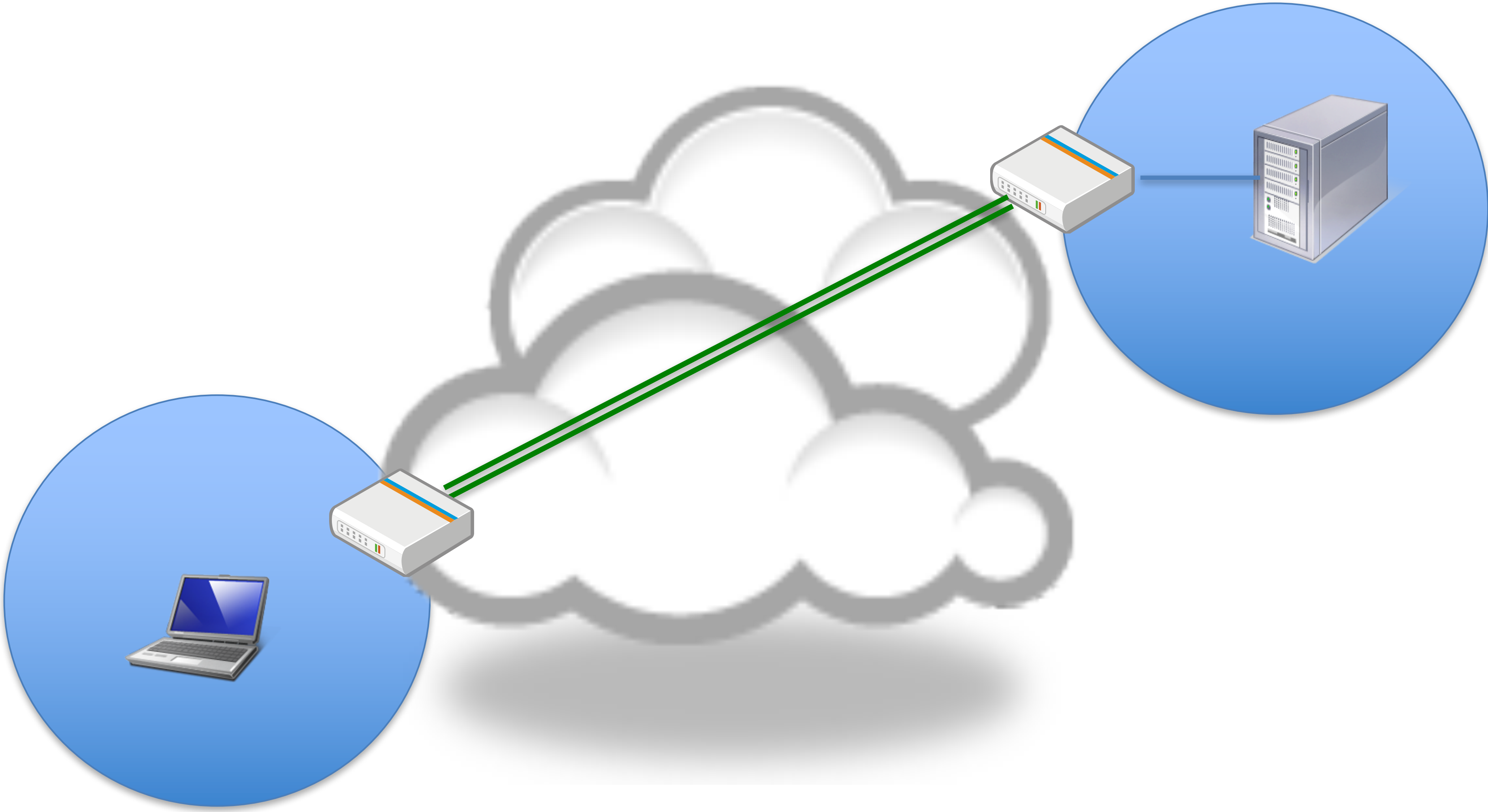
VPN, IPsec and TLS

<maz@iij.ad.jp>
stole slides from
Merike Kaeo
<merike@doubleshotsecurity.com>

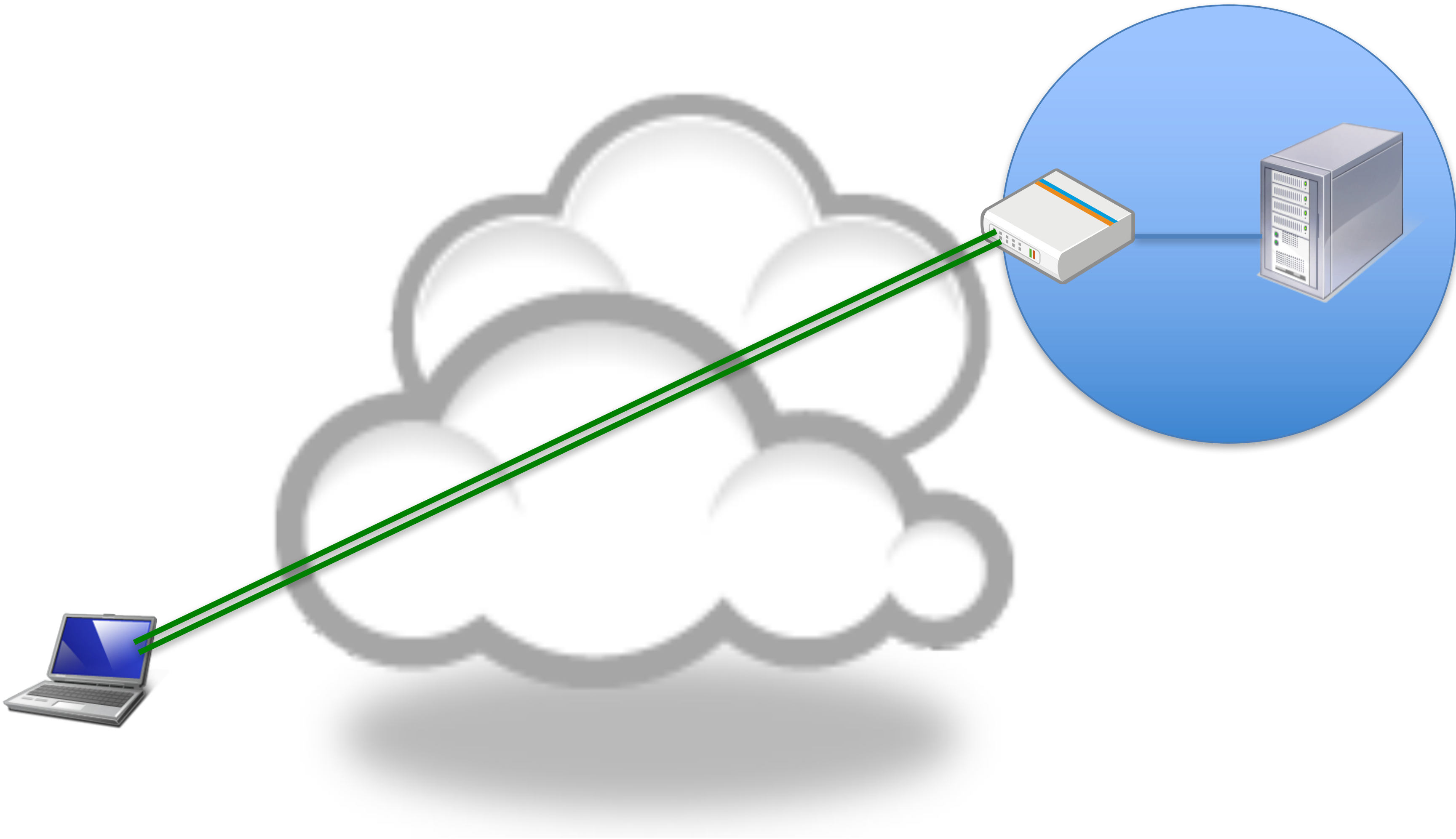
Virtual Private Network

- Overlay Network
  - a VPN is built on top of a public network (Internet)
- Cost effective
  - You don’t need to expand your network
- Rapidly deployable
  - An underlay network just carries IP packets as usual
  - Only your nodes need to agree about VPN
- Control
  - You can enforce your own policy in the VPN

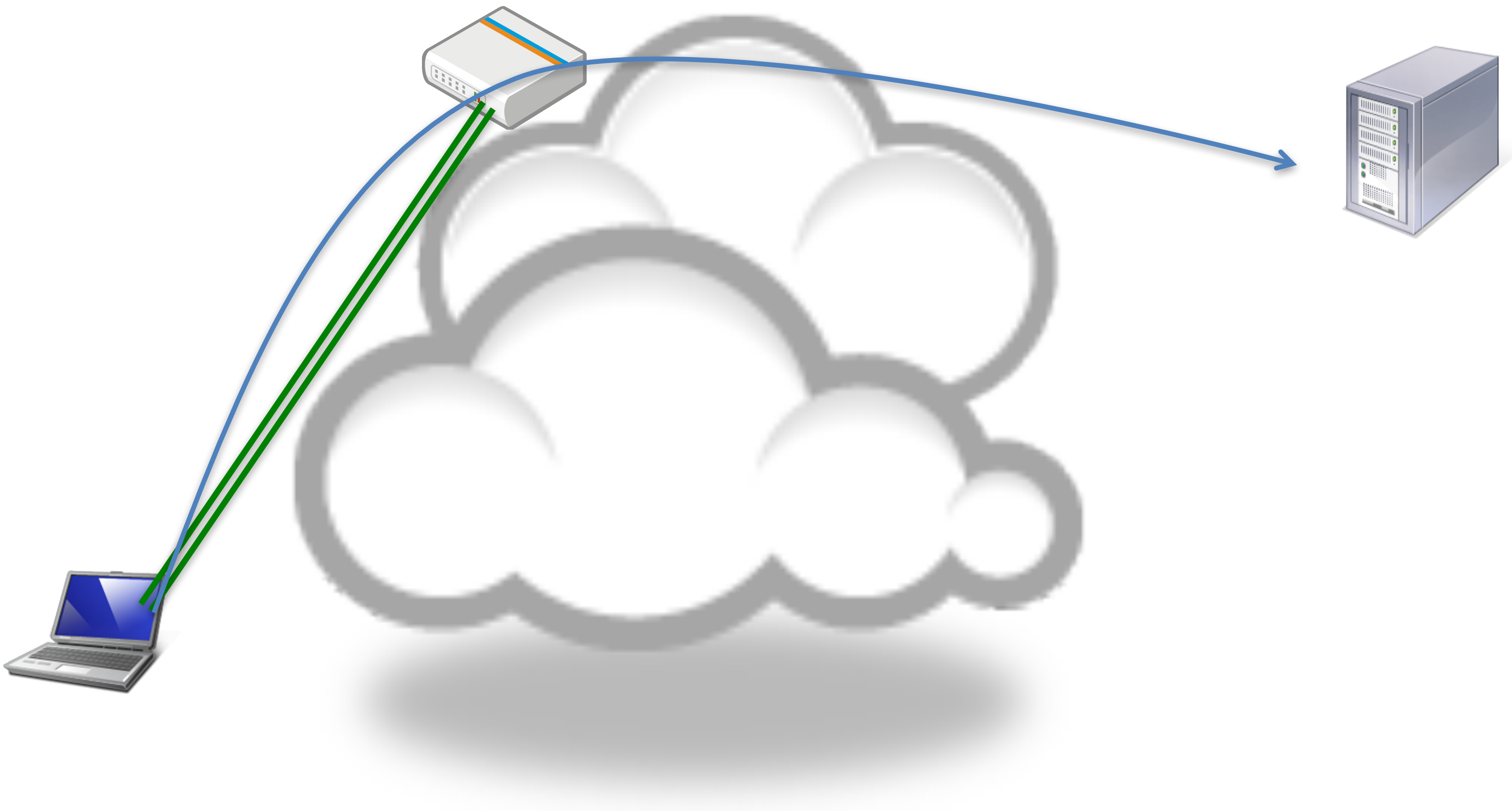
satellite office



access to intranet from outside

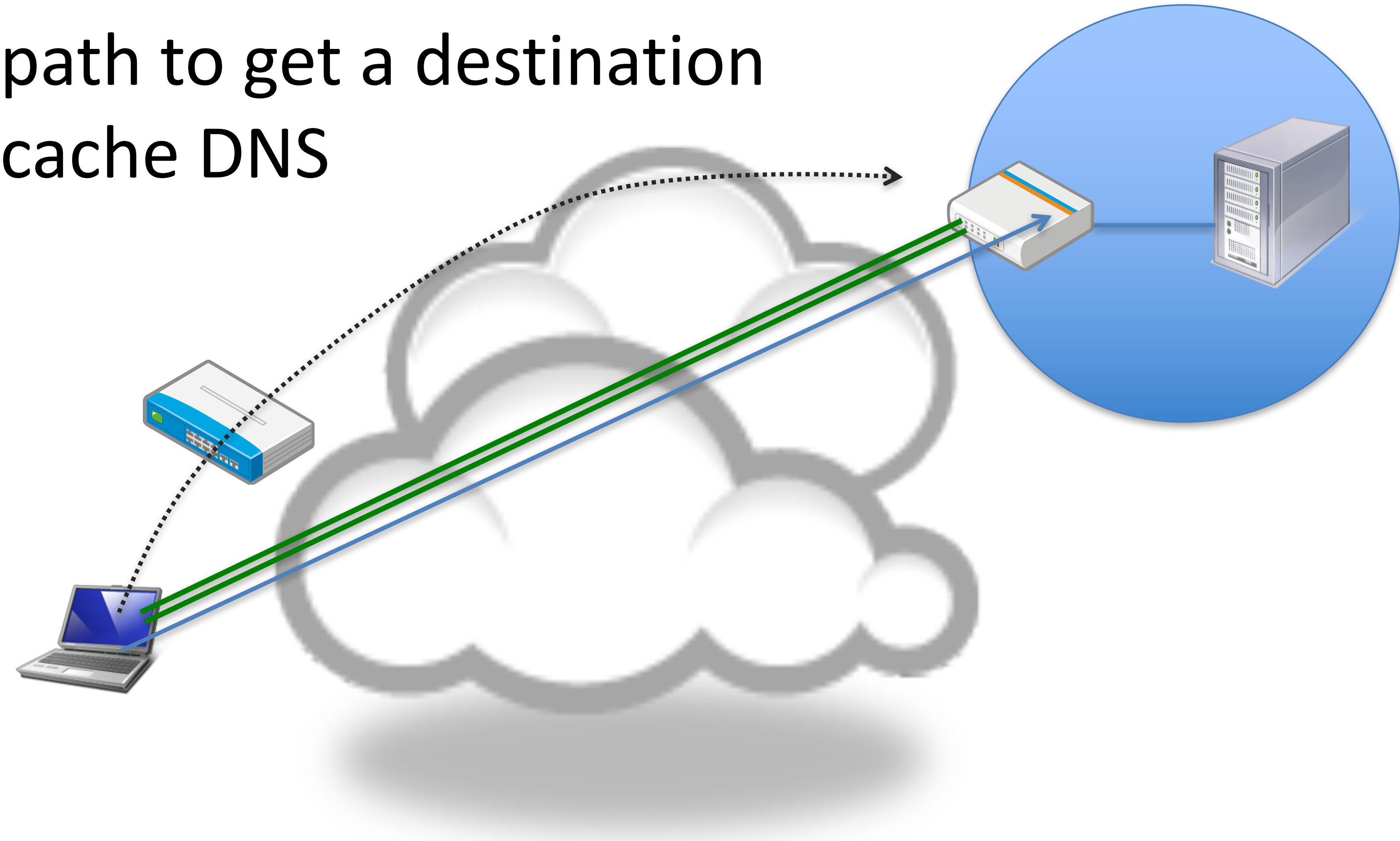


over an untrusted network



IPv4 and IPv6

- path to get a destination
- cache DNS



VPN and security

- 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

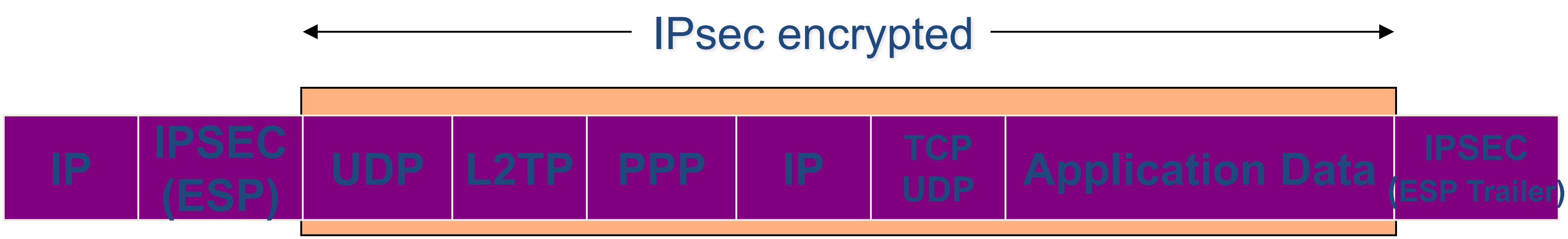
- PPTP
  - IP over PPP over GRE
  - possible password leakage by MS-CHAPv2 weakness
- OpenVPN
  - IP over TLS over TCP/UDP
- MS-SSTP
  - IP over PPP over SSTP over HTTPS over TCP
- L2TP/IPsec
  - IP over PPP over L2TP over UDP over ESP
- IPsec
  - IP over ESP
  - IP over ESP over UDP (NAT traversal)

Layer 2 Tunneling Protocol

- Designed in IETF PPP Extensions working group
  - Combination of Cisco L2F & PPTP features
  - L2TP RFC 2661, Aug 1999
  - Uses UDP port 1701 for control and data packets
  - Uses PPP for packet encapsulation – carries most protocols (also non-IP protocols)
- Security Functionality
  - Control session authentication, keepalives
  - EAP for a broader authentication mechanisms
  - IPsec ESP for confidentiality and integrity
  - IKE for key management

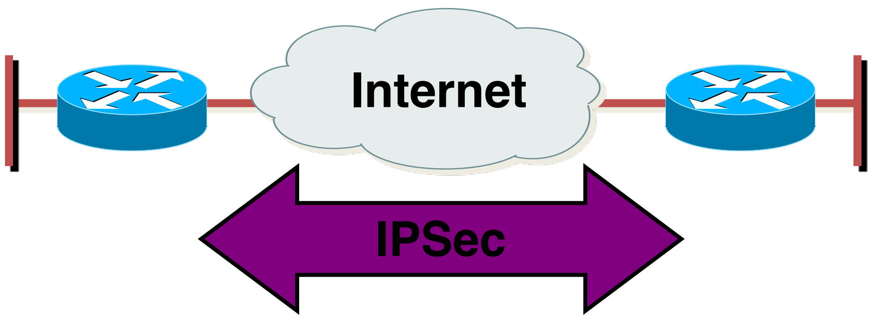
L2TP and IPsec

Multiple Encapsulations
…..careful of packet size!!



Ping with large MTU size….help discover fragmentation issues!!

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

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

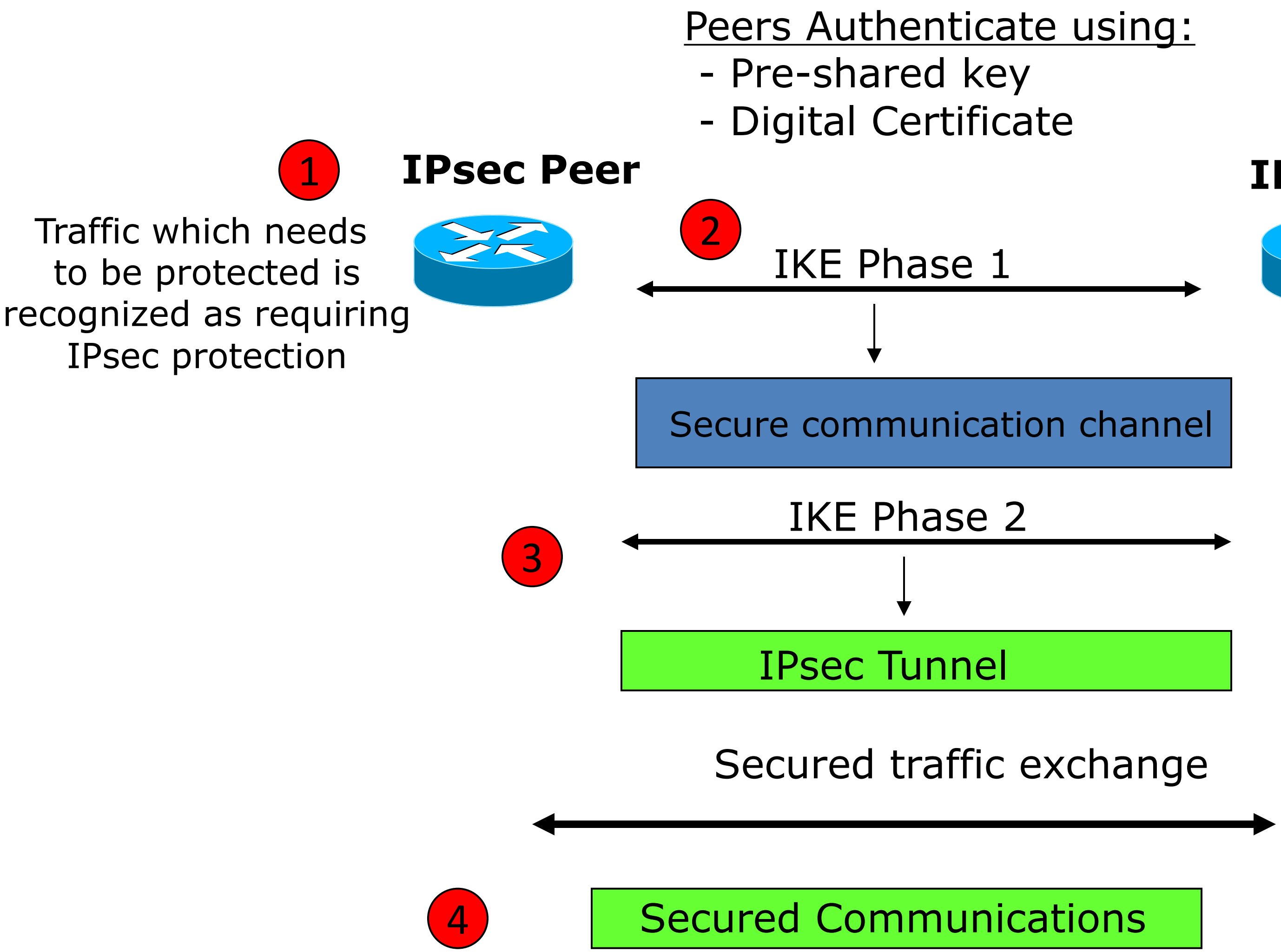
Interoperable Defaults For SAs

- 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 ?

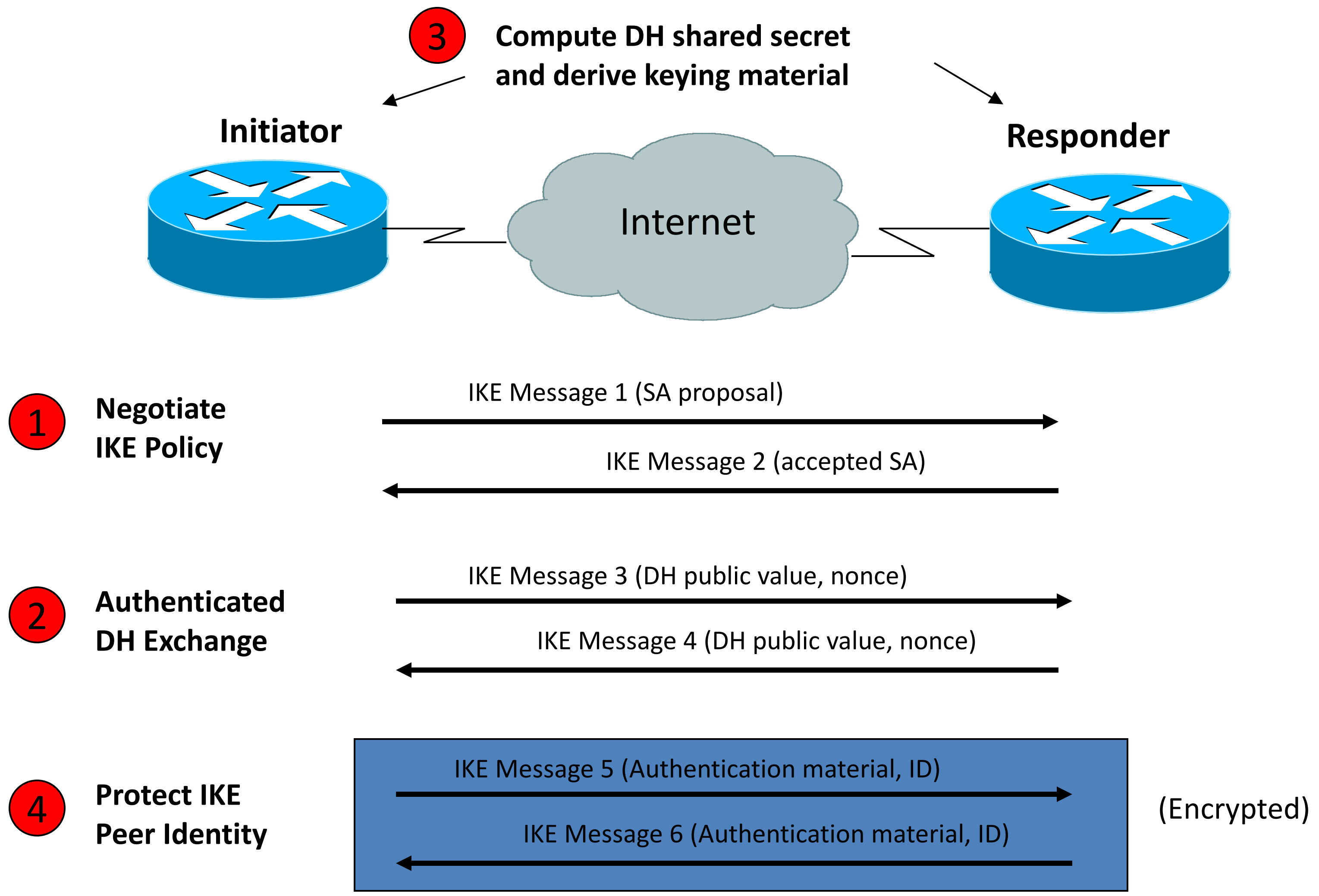
IPsec with IKE



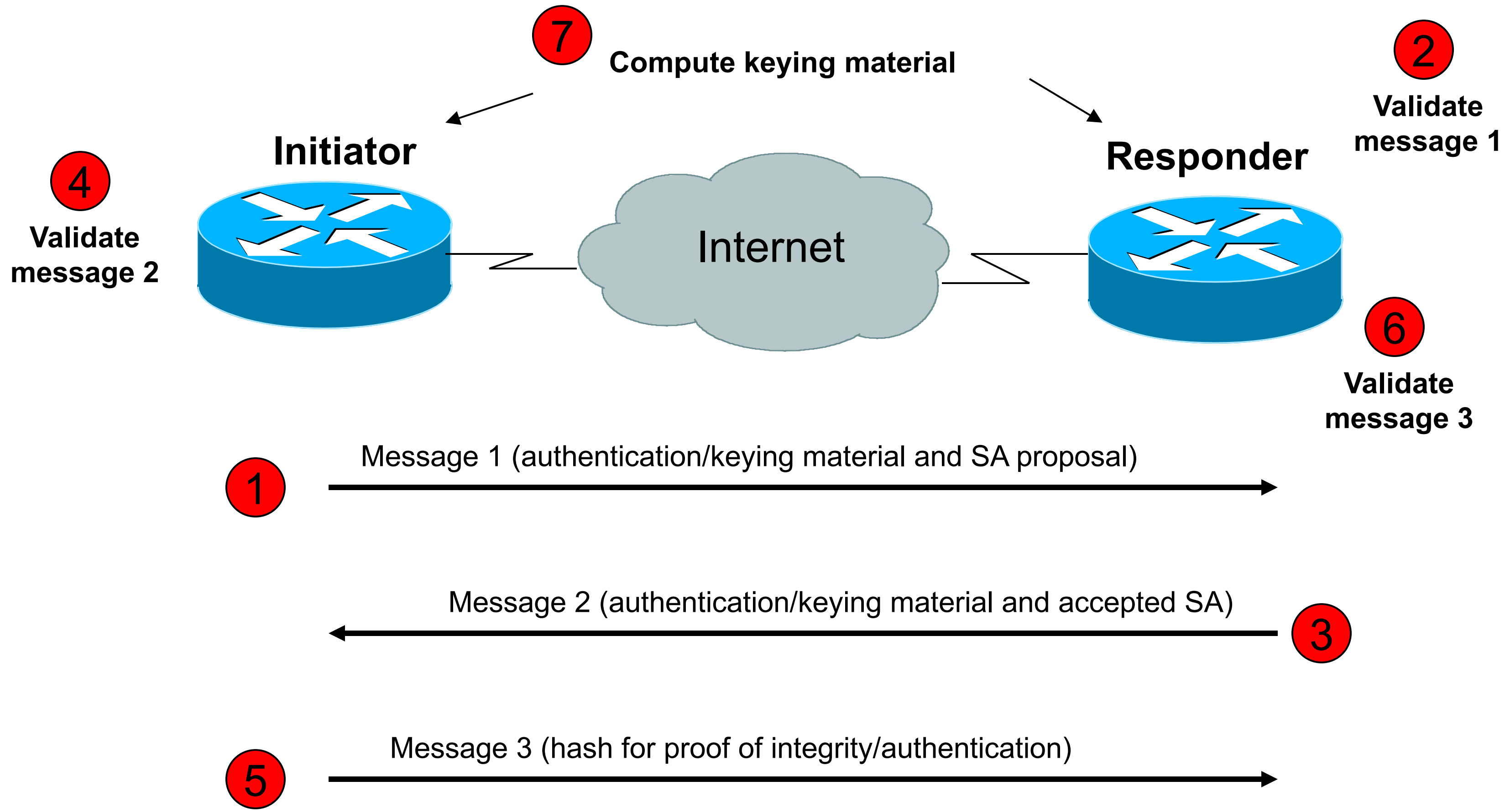
IPsec IKE Phase 1 Uses DH Exchange

- 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



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

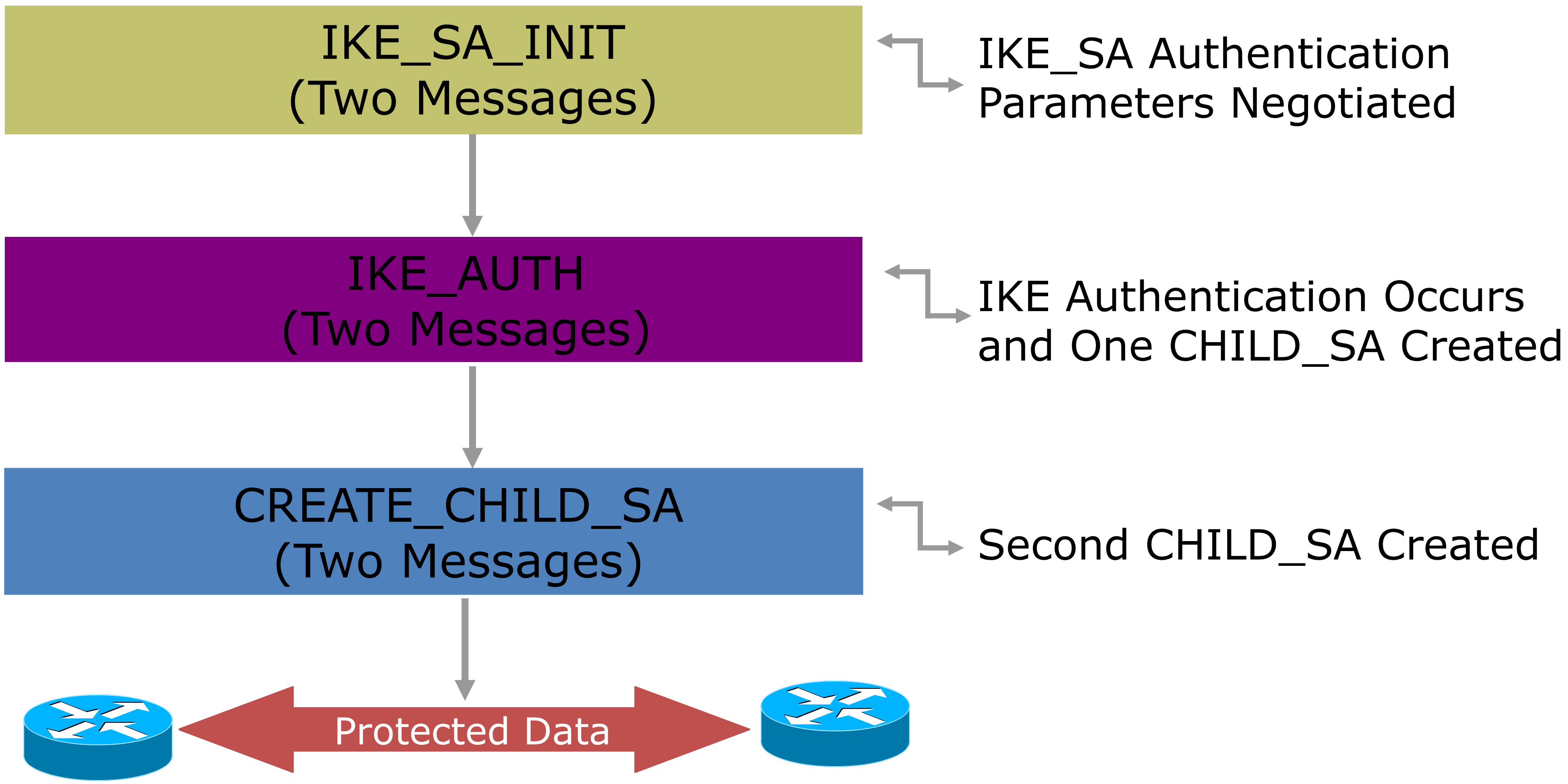
IKE v2: What Is Not Changing

- 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?



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

Considerations For Using IPsec

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

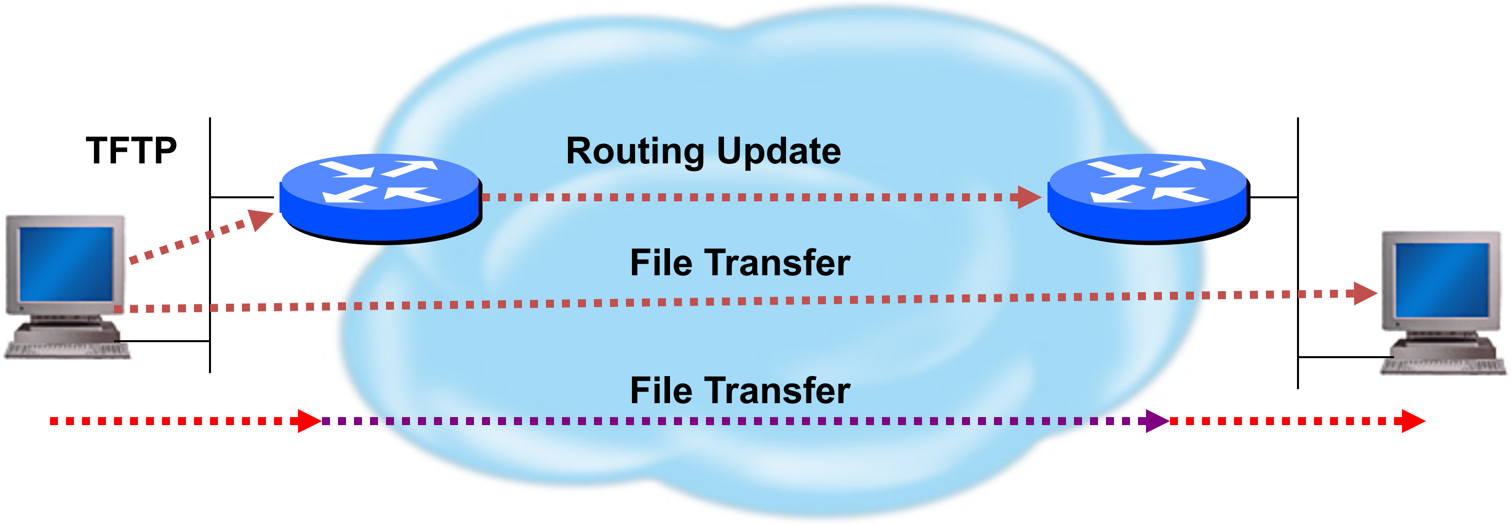
Non-Vendor Specific Deployment Issues

- 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

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 Concerns

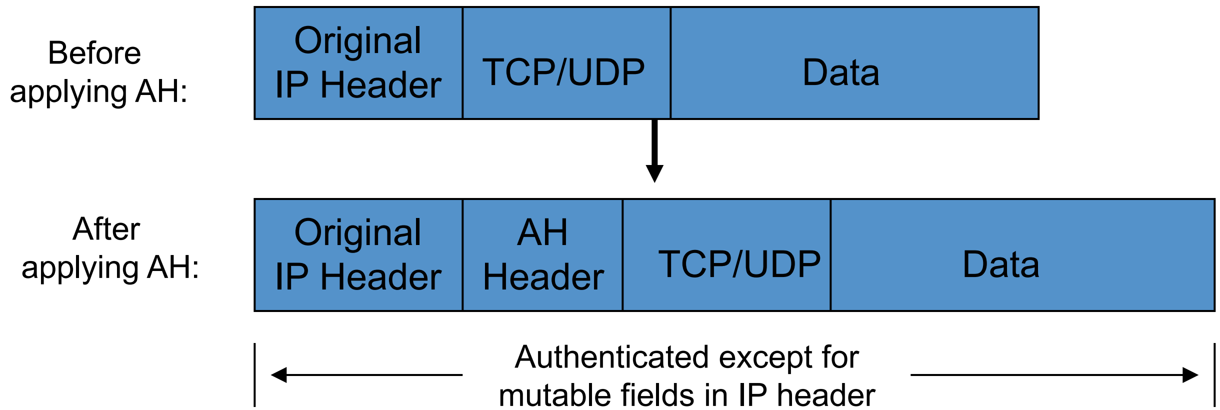
- Are enough people aware that IKEv2 is not backwards compatible with IKEv1?
  - IKEv1 is used in most IPsec implementations
  - Will IKEv2 implementations first try IKEv2 and then revert to IKEv1?
- Is IPsec implemented for IPv6?
  - Some implementations ship IPv6 capable devices without IPsec capability 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 determine if ESP-Null vs Encrypted

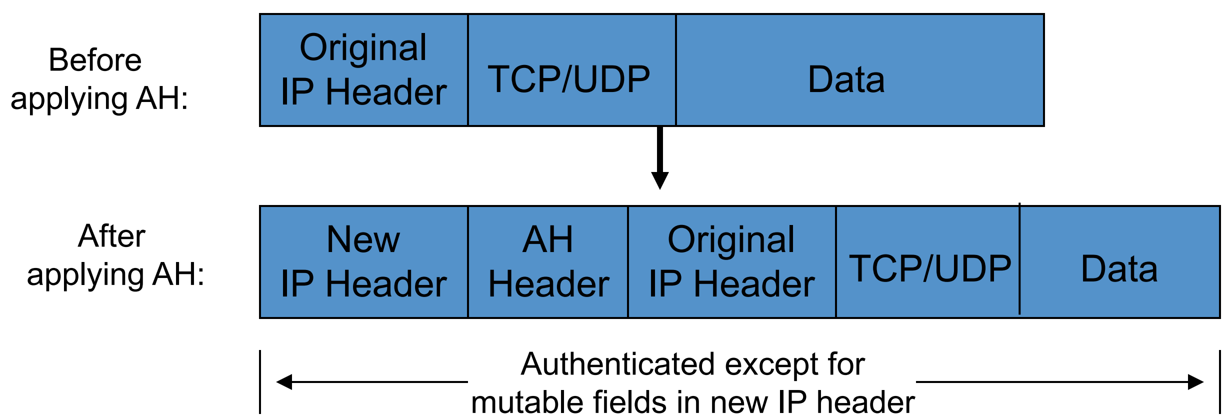
IPv4 IPsec AH

IPv4 AH Transport Mode:



Mutable Fields:
- ToS
- TTL
- Hdr Checksum
- Offset
- Flags

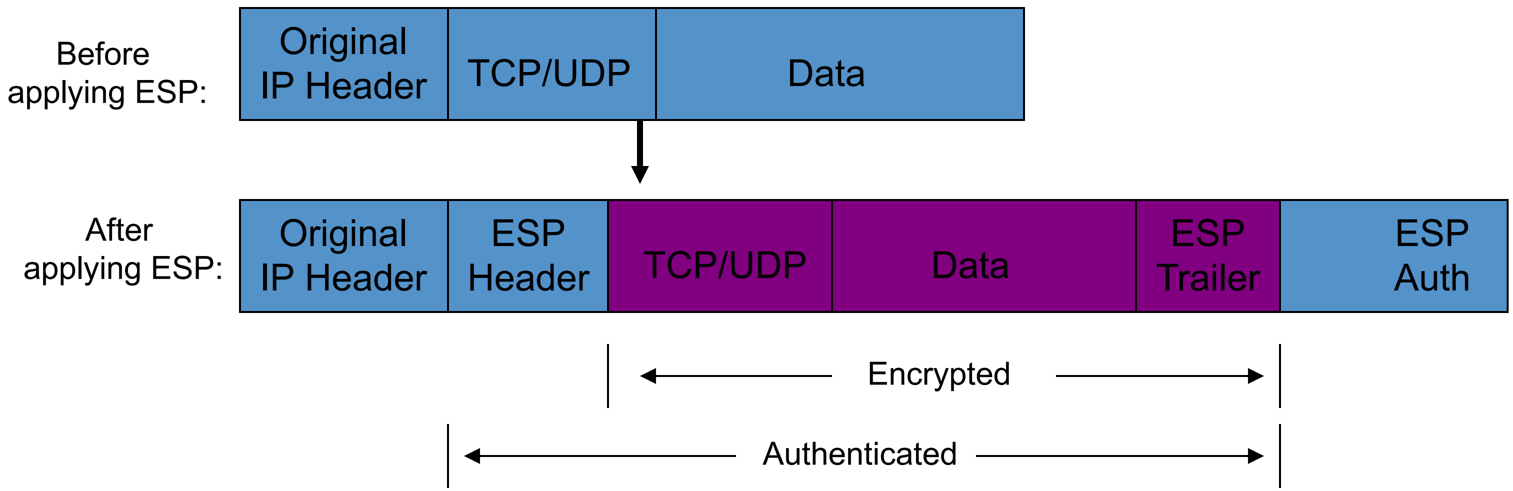
IPv4 AH Tunnel Mode:



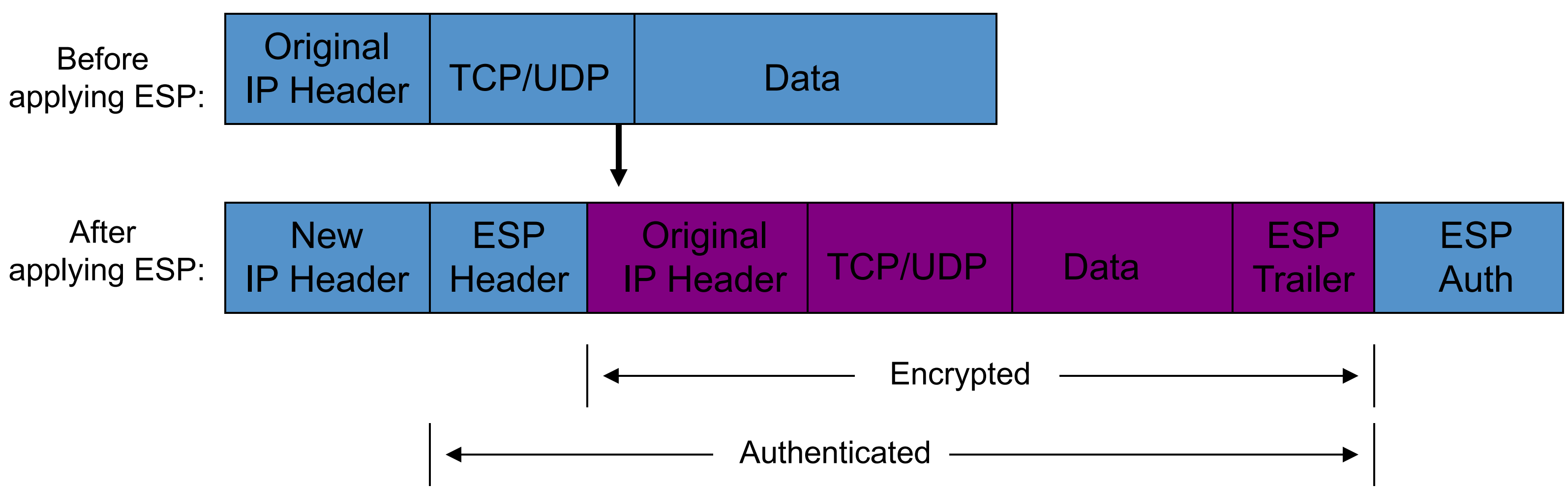
Mutable Fields:
- ToS
- TTL
- Hdr Checksum
- Offset
- Flags

IPv4 IPsec ESP

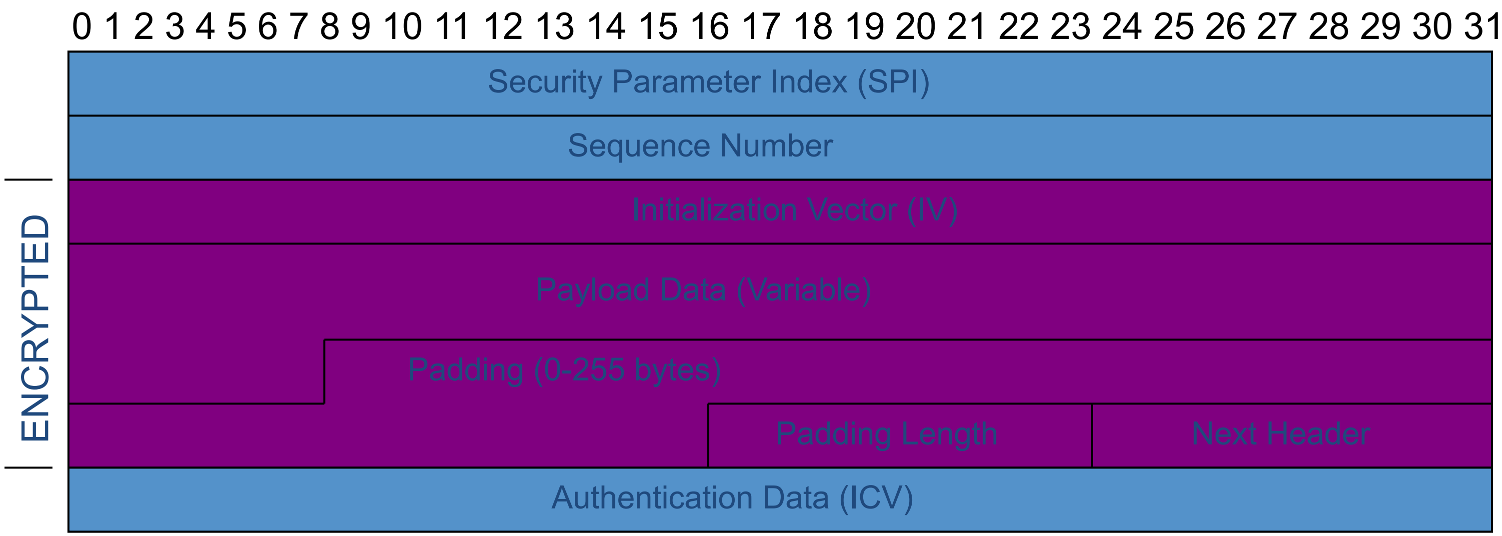
IPv4 ESP Transport Mode:



IPv4 ESP Tunnel Mode:

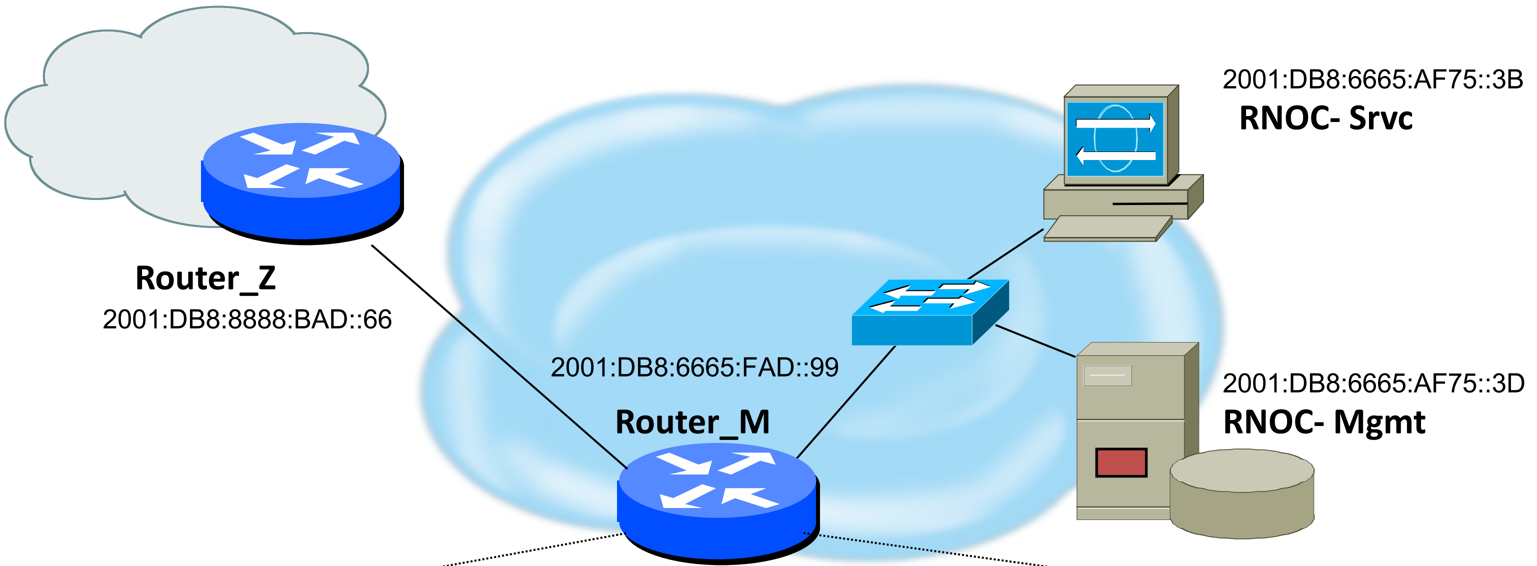


ESP Header Format



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)

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’

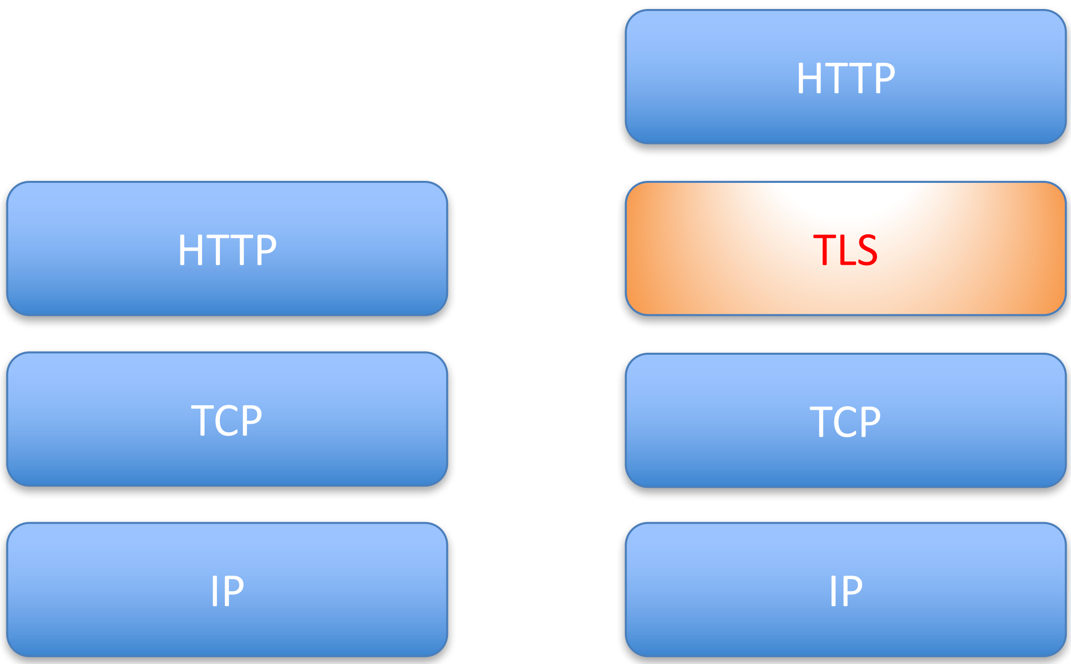
Pretty Good IPsec Policy

- 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)

Help With Configuring IPsec

- 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

HTTP and Secure Channel



SSL/TLS

- SSL and TLS
  - SSL v3.0 specified in an I-D in 1996 (draft-freier-ssl-version3-02.txt) and now in RFC6101
  - TLS v1.0 specified in RFC2246
    - TLS v1.0 = *SSL v3.1* ≈ SSL v3.0
  - TLS v1.1 specified in RFC4346
  - TLS v1.2 specified in RFC5246
- Goals of protocol
  - Secure communication between applications
  - Data encryption
  - Server authentication
  - Message integrity
  - Client authentication (optional)

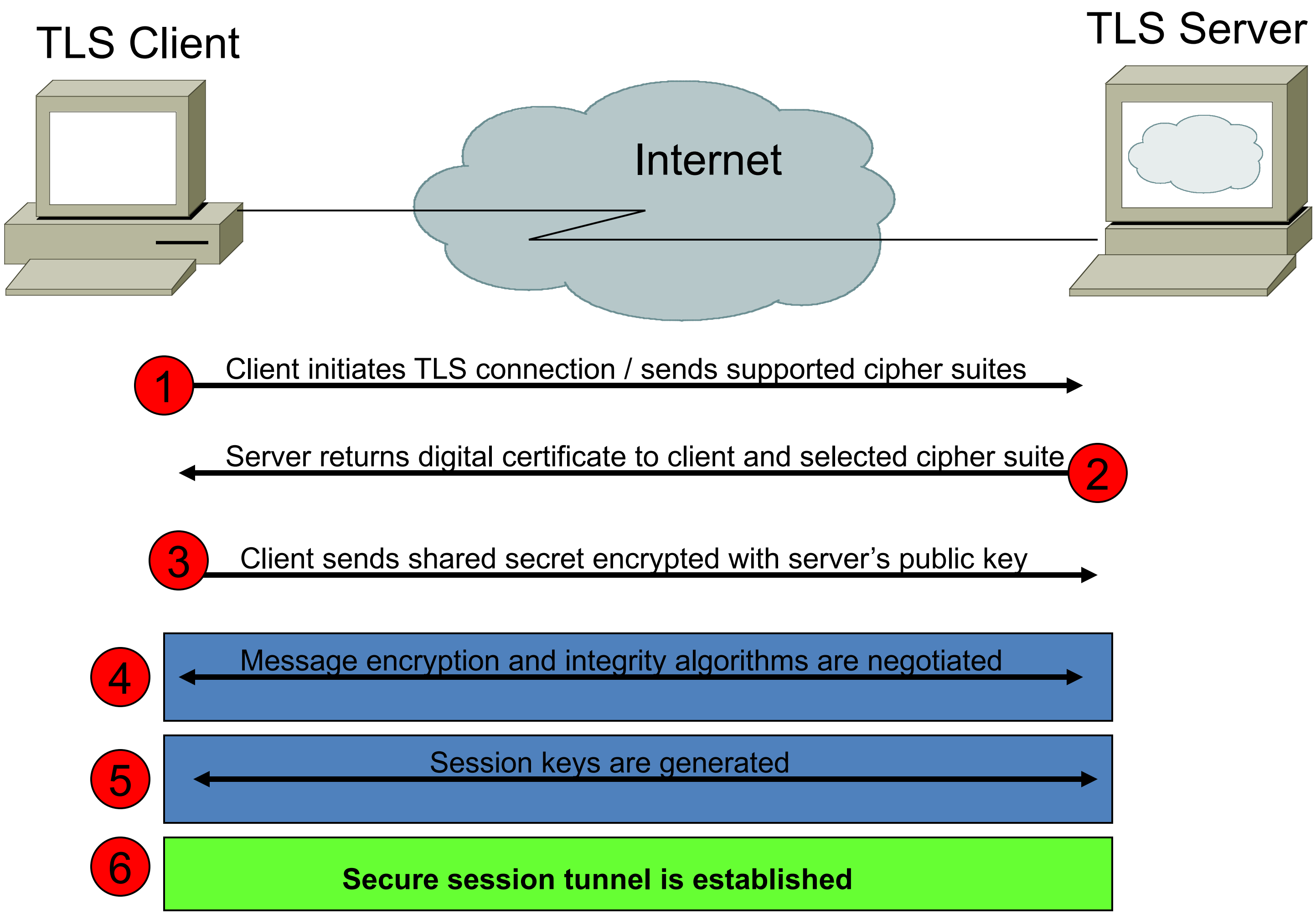
SSL is not secure any more

- SSL2.0 and SSL3.0 have known vulnerabilities in protocol specifications
  - downgrade attack
  - POODLE attack
  - RFC6176 - Prohibiting Secure Sockets Layer (SSL) Version 2.0
  - RFC7568 - Deprecating Secure Sockets Layer Version 3.0
- Use TLS instead

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 ECDSA)
- Connection is reliable
  - Message transport includes a message integrity check using a keyed MAC.
  - Secure hash functions (such as SHA384, SHA256) are used for MAC computations.

The TLS Handshake Process



TLS 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 TLS

TLS IANA Assigned Port #s

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

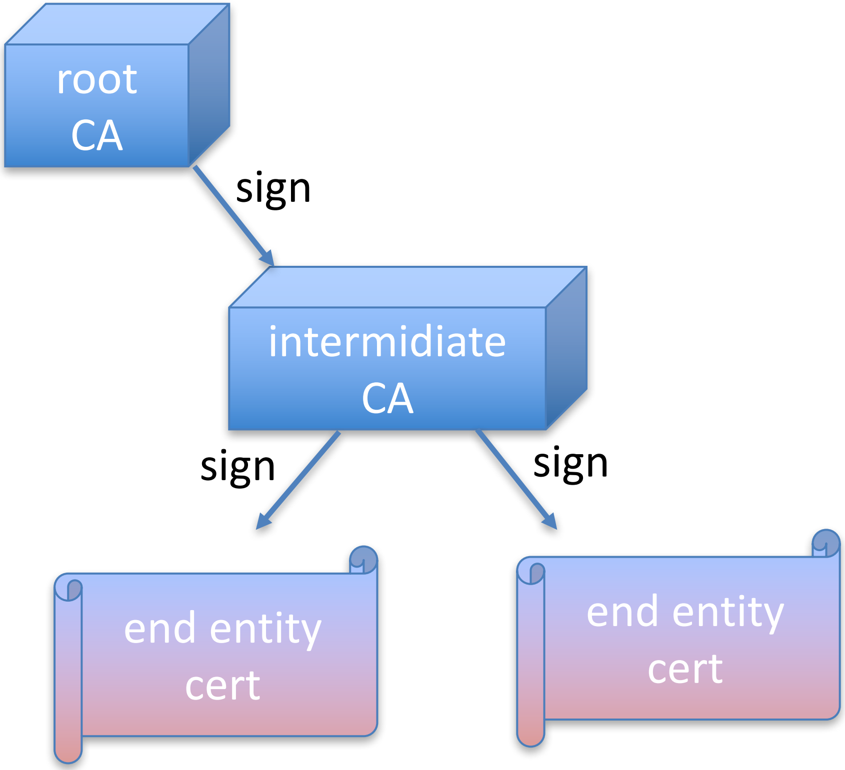
TLS policy example

- Server Key
  - RSA 2048bit or more
  - ECDSA 256bit or more
- Protocols
  - enable TLS1.2, TLS1.1, TLS1.0 and disable SSL
- Ciphers Suites
  - TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
    - 1024bit or more key length
  - TLS_RSA_WITH_AES_128_GCM_SHA256
  - TLS_RSA_WITH_AES_128_CBC_SHA
    - 2048bit or more key length

Certificate Authority

- issues a digital certificate which is signed by the CA's private key
- You can verify the certificate using the corresponding public key
  - if you trust the public key
- …and CA can have hierarchical trust model

Trust chain

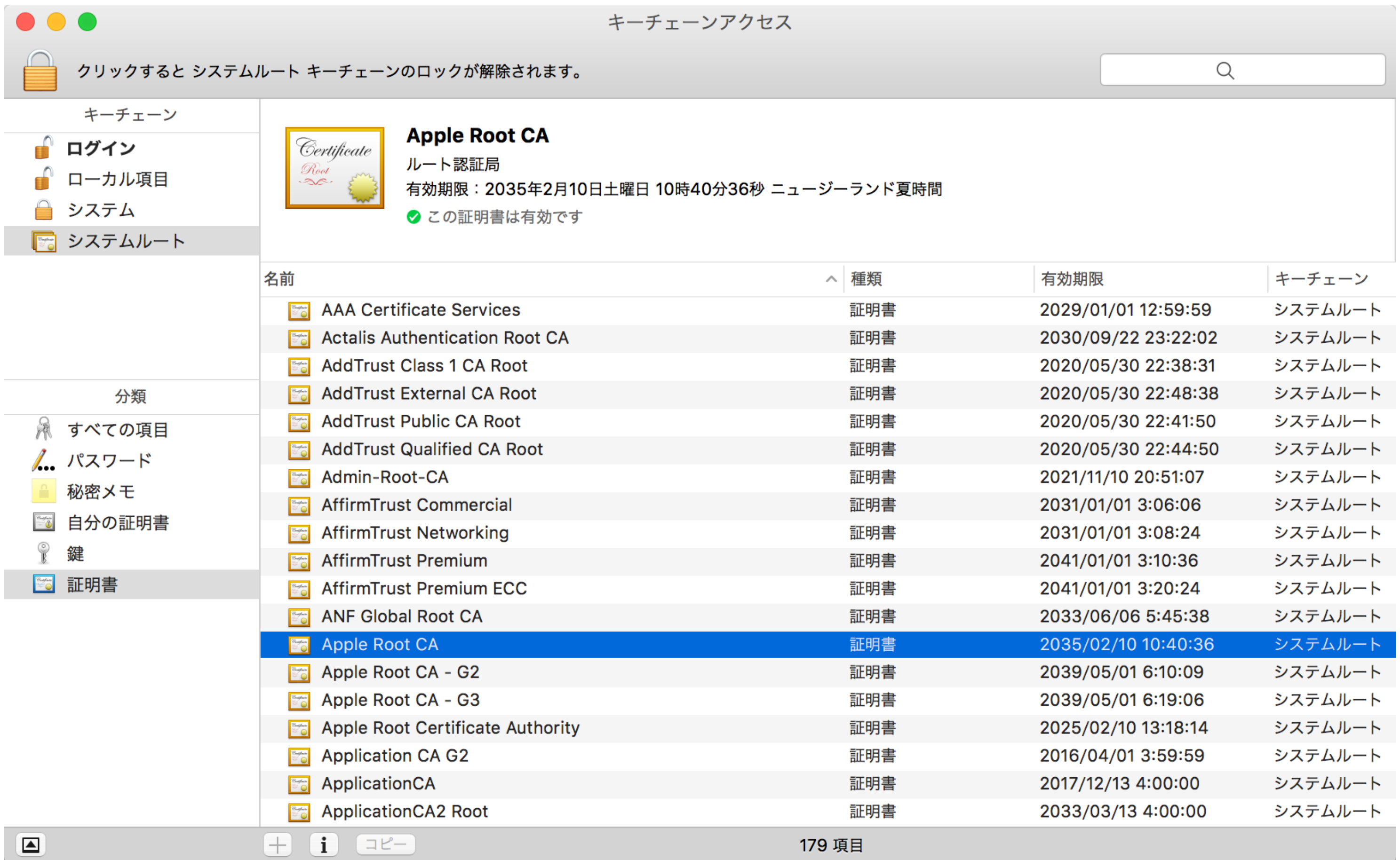


root
CA
sign
intermidiate
CA
sign
sign
end entity
cert
end entity
cert

https://www.apricot.net

https://wiki.rg.net/

trusted CA


キーチェーンアクセス
クリックすると システムルート キーチェーンのロックが解除されます。
キーチェーン
ログイン
ローカル項目
システム
システムルート
分類
すべての項目
パスワード
秘密メモ
自分の証明書
鍵
証明書
Apple Root CA
ルート認証局
有効期限：2035年2月10日土曜日 10時40分36秒 ニュージーランド夏時間
この証明書は有効です
名前
種類
有効期限
キーチェーン
AAA Certificate Services 証明書 2029/01/01 12:59:59 システムルート
Actalis Authentication Root CA 証明書 2030/09/22 23:22:02 システムルート
AddTrust Class 1 CA Root 証明書 2020/05/30 22:38:31 システムルート
AddTrust External CA Root 証明書 2020/05/30 22:48:38 システムルート
AddTrust Public CA Root 証明書 2020/05/30 22:41:50 システムルート
AddTrust Qualified CA Root 証明書 2020/05/30 22:44:50 システムルート
Admin-Root-CA 証明書 2021/11/10 20:51:07 システムルート
AffirmTrust Commercial 証明書 2031/01/01 3:06:06 システムルート
AffirmTrust Networking 証明書 2031/01/01 3:08:24 システムルート
AffirmTrust Premium 証明書 2041/01/01 3:10:36 システムルート
AffirmTrust Premium ECC 証明書 2041/01/01 3:20:24 システムルート
ANF Global Root CA 証明書 2033/06/06 5:45:38 システムルート
Apple Root CA 証明書 2035/02/10 10:40:36 システムルート
Apple Root CA - G2 証明書 2039/05/01 6:10:09 システムルート
Apple Root CA - G3 証明書 2039/05/01 6:19:06 システムルート
Apple Root Certificate Authority 証明書 2025/02/10 13:18:14 システムルート
Application CA G2 証明書 2016/04/01 3:59:59 システムルート
ApplicationCA 証明書 2017/12/13 4:00:00 システムルート
ApplicationCA2 Root 証明書 2033/03/13 4:00:00 システムルート
コピー
179 項目

CA and certificates

- CA can issue a certificate for any domainname
  - if you trust the CA, the certificate looks legitimate
- if you have a malicious CA in your trusted keychain, an attacker can monitor/modify your TLS session data
- Yes, we have cases
  - https://support.lenovo.com/nz/en/product_security/superfish
  - https://www.dell.com/support/article/us/en/19/SLN300321

Check your trusted CA

- Windows
  - certlm.msc
- Mac OS X
  - Keychain Access.app
- Firefox
  - Setting -> Advanced -> Certificates -> View Certificates

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