Network Security Protocols

DNSSEC, VPN, IPSec, TLS, and HTTPs

DNSSEC

DNSSEC

- DNSSEC – Domain Name System Security Extension
- DNSSEC are a suite of IETF specifications.
- It provides security to the information provided by DNS.
- There are several threats to the DNS (Recall DNS Attacks), a few of which are specific to peculiarities of the DNS protocol.
- DNSSEC is a useful tool in defending against these threats.

DNS Vulnerabilities

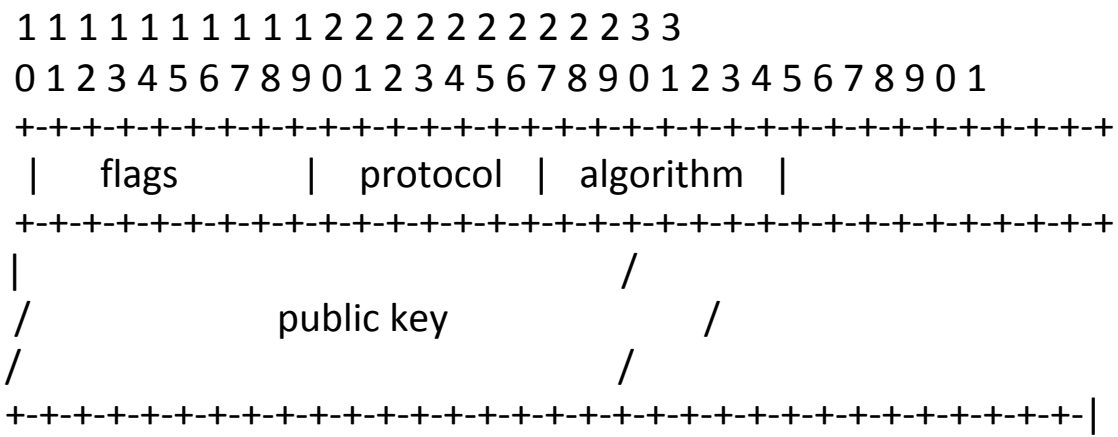
- Packet Interception
- ID Guessing and Query Prediction
- Name Chaining
- Betrayal by Trusted Server
- Denial of Service
- Authenticated Denial of Domain Names
- Wildcards

DNSSEC Services

- Security Extensions to the Domain Name System provide security to the resolvers and applications through the use of cryptographic digital signatures.
- Digital Signatures are included in secured zones as resource records.
- The Domain Name System Security Extensions provide three distinct services:
  - Key Distribution
  - Data Origin Authentication Service
  - DNS Transaction and Request Authentication

Key Distribution

- Every DNS name is associated with public key.
- A KEY resource record (RR) is used to store a public key.
- Every KEY RR has RDATA, which consists of flags, a protocol octet, the algorithm number octet, and the public key.
- The format of RDATA is:



Data Origin Authentication Service

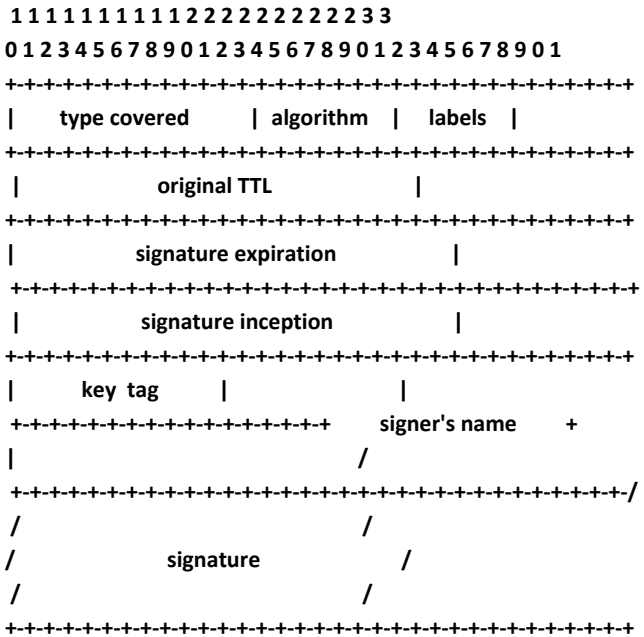
- The data origin authentication key(s) are associated with the zone (not with the servers that store copies of the data).
- Authentication is provided by associating the resource record sets in the DNS with cryptographically generated digital signatures.
- Commonly, there will be a single private key that authenticates an entire zone.
- A resolver could learn a public key of a zone either by reading it from the DNS or by having it statically configured.
- When security aware resolver reliably learns a public key of the zone, it can authenticate the signed data read from that zone.

DNS Transaction and Request Authentication

- The data origin authentication service protects retrieved resource records but provides no protection for DNS requests or for message headers.
- The SIG or "signature" resource record is the fundamental way that data is authenticated in the secure DNS.
- The SIG RR authenticates an RRset of a particular type, class, and name and binds it to a time interval and the signer's domain name.

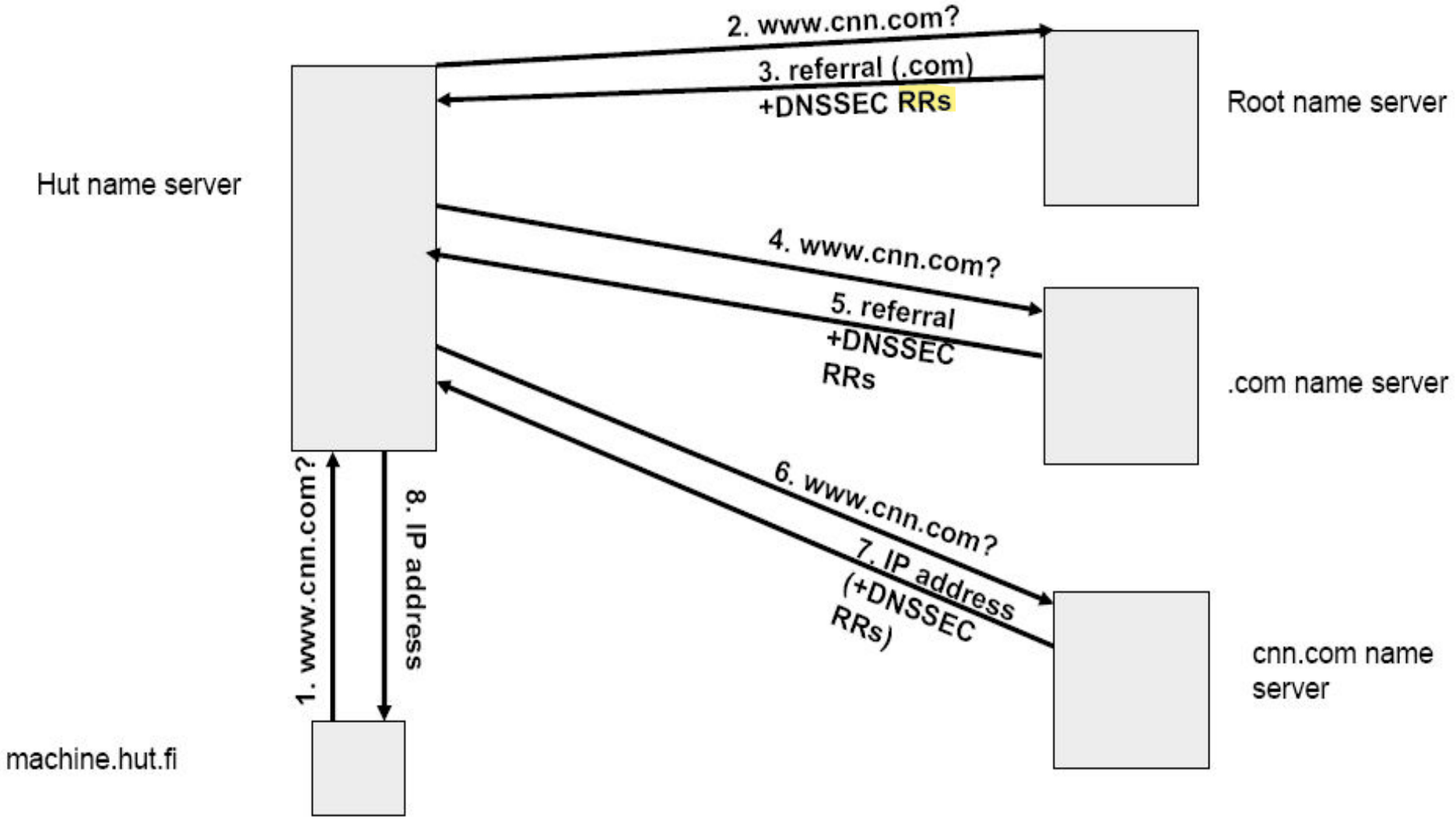
DNS Transaction and Request Authentication

The format of SIG RR :



DNS Transaction and Request Authentication

- Transaction Authentication is accomplished by adding a special SIG resource record at the end of the reply or the request.
- The SIG RR digitally signs the concatenation of the server's response and the resolver's query.
- The private keys used in transaction security belong to the entity composing the reply, not to the zone involved.
- Requests and replies are highly variable hence message authentication SIGs cannot be pre-calculated.
- Adding data origin authentication and integrity requires no change to the DNS protocol beyond the addition of the signature resource type and the key resource type needed for key distribution.
- This service can be supported by existing resolver and caching server implementations so long as they can support the additional resource types.


2. www.cnn.com?
3. referral (.com)
+DNSSEC RRs
Root name server
Hut name server
4. www.cnn.com?
5. referral
+DNSSEC
RRs
.com name server
6. www.cnn.com?
7. IP address
(+DNSSEC
RRs)
cnn.com name
server
1. www.cnn.com?
8. IP address
machine.hut.fi

Services Not Provided

- DNS gives the same answers to all inquirers. It does not include any sort of access control lists or other means to differentiate inquirers.
- It does not provide any confidentiality for queries or responses. This service may be available via IPSEC, TLS, or other security protocols.
- Cannot handle Zone transfers when there is a break in the authentication chain.

Weakness of DNSsec

- Complete Authentication not achieved: The various receptions which have been authenticated by cryptographically generated digital signature with DNS RRSets are not encrypted. It does not protect against DoS attacks directly.
- It just authenticates that the owner of the domain is valid or not, it doesn't matter to DNSSec if that data is not fully correct or from which source it is coming. It might come in from some hacker (man in middle), but it will still be authenticated.
- An attacker can query the NSEC RRs in sequence to obtain all the names in a zone.

Weakness of DNSsec

- The record number in the database grows roughly by a factor of three (NSEC, RRSIG records needed). In the case of servers over a large area, it becomes really difficult to make modifications in the implementations of DNSSec. Thus, this limitation makes DNSSec suitable for only short ranges.
- There is a main source of contention whether users should be allowed access to the main root .com. Currently, this root is only provided if the user wants a large domain space.
- Roots are provided without any authentication which can be used by the owner to perform illegal or malicious activities. For example, it is easy to obtain .tk root xtension domain, so an owner of such a domain can easily make a false site to deceive users and breach their security.

DNSSEC Conclusion

- DNSSec is designed to provide some security for DNS. It provides data authentication and data integrity.
- DNSSec uses cryptographically generated digital signatures with DNS RRsets.
- There are many challenges which make it difficult to properly implement DNSSec.
- Since DNSSec was not designed to meet specific design goals, it is difficult to measure the success of DNSSec.
- Despite various drawbacks with DNSSec, it still provides a great deal of security for DNS.

IP Security (IPSec)

IP Security (IPSec)

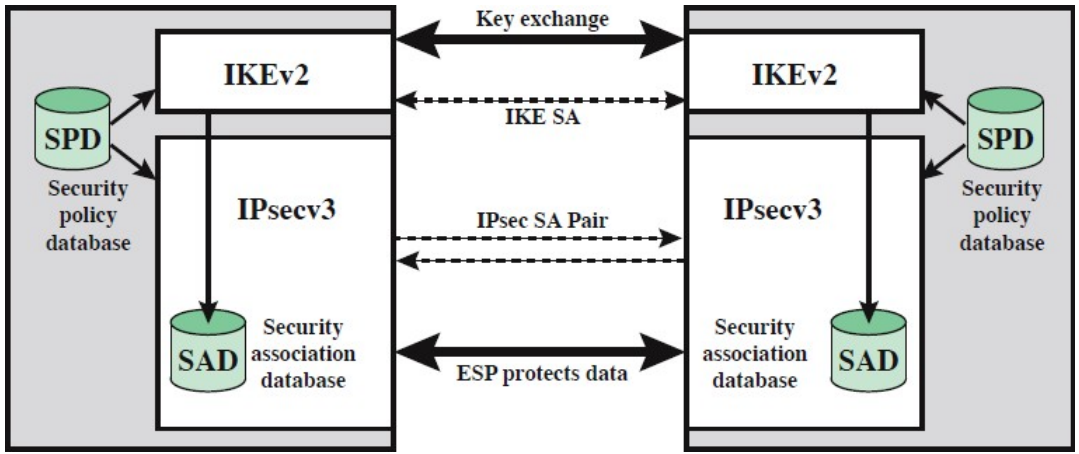
- IPSec provides
  - Access control: User authentication
  - Data integrity
  - Data origin authentication
  - Rejection of replayed packets
  - Confidentiality (encryption)
  - Limited traffic flow confidentiality
- Benefits:
  - Security at Layer 3 ⇒ Applies to all transports/applications
  - Can be implemented in Firewall/router
    ⇒ Security to all traffic crossing the perimeter
  - Transparent to applications and can be transparent to end users
  - Can provide security for individual users
- Applications: VPNs, Branch Offices, Remote Users, Extranets

IPSec services

Services	*AH*	*ESP*
Access control	Yes	Yes
Message authentication (message integrity)	Yes	Yes
Entity authentication (data source authentication)	Yes	Yes
Confidentiality	No	Yes
Replay attack protection	Yes	Yes

IPSec Architecture

- Internet Key Exchange (IKE)
- IPSec
- Security Association Database
- Security Policy database



Security Association (SA) Database

- Each host has a database of Security Associations (SAs)
- SA = One-way security relationship between sender & receiver Two-way may use different security Two SA's required
- Defined by 3 parameters:
  - Security Parameters Index (SPI)
  - IP Destination Address
  - Security Protocol Identifier: AH or ESP
- For each SA, the database contains:
  - SPI
  - Sequence number counter and counter overflow flag
  - Anti-replay window
  - AH Information and ESP information
  - Lifetime of the SA
  - Mode: Transport or tunnel or wildcard
  - Path MTU

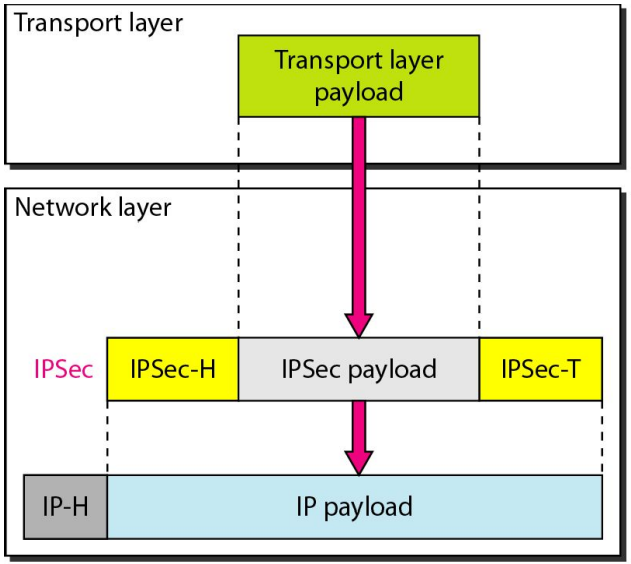
Security Policy Database

- Relates IP traffic to specific SAs
  - Match subset of IP traffic to relevant SA
  - Use selectors to filter outgoing traffic to map
  - Based on: local & remote IP addresses, next layer protocol, name, local & remote ports

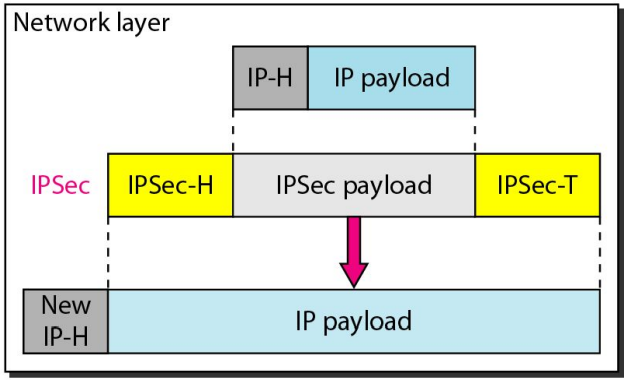
Protocol	Local IP	Port	Remote IP	Port	Action	Comment
UDP	1.2.3.101	500	*	500	BYPASS	IKE
ICMP	1.2.3.101	*	*	*	BYPASS	Error messages
*	1.2.3.101	*	1.2.3.0/24	*	PROTECT: ESP intransport-mode	Encrypt intranet traffic
TCP	1.2.3.101	*	1.2.4.10	80	PROTECT: ESP intransport-mode	Encrypt to server
TCP	1.2.3.101	*	1.2.4.10	443	BYPASS	TLS: avoid double encryption
*	1.2.3.101	*	1.2.4.0/24	*	DISCARD	Others in DMZ
*	1.2.3.101	*	*	*	BYPASS	Internet

Transport mode and tunnel modes of IPSec protocol

- IPSec in the transport mode does not protect the IP header; it only protects the information coming from the transport layer.



a. Transport mode

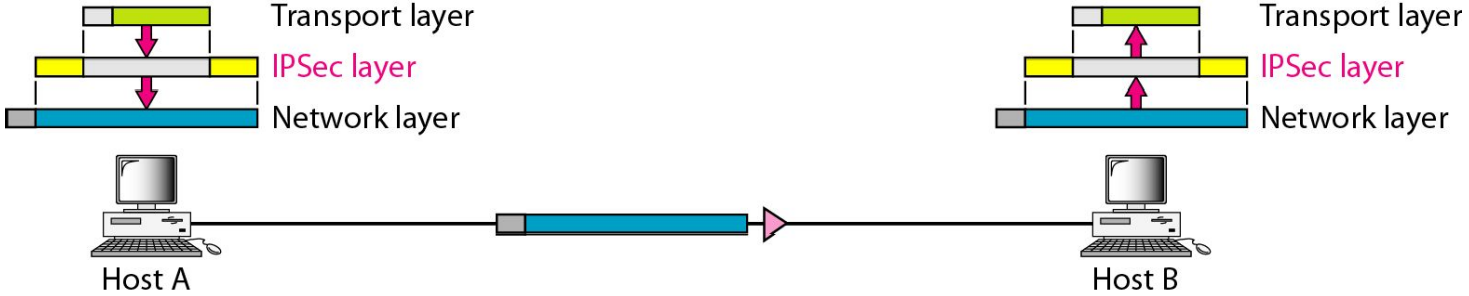


b. Tunnel mode

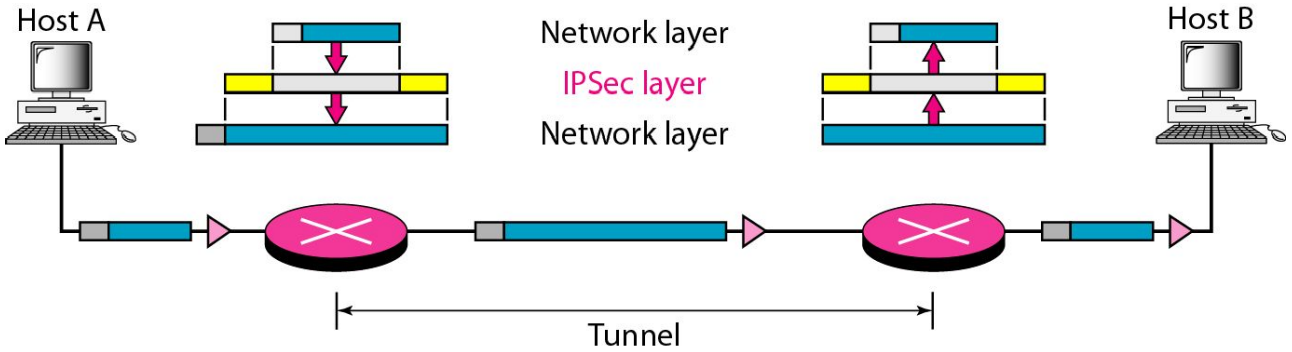
IPSec in tunnel mode protects the original IP header.

Transport and Tunnel Mode in Action

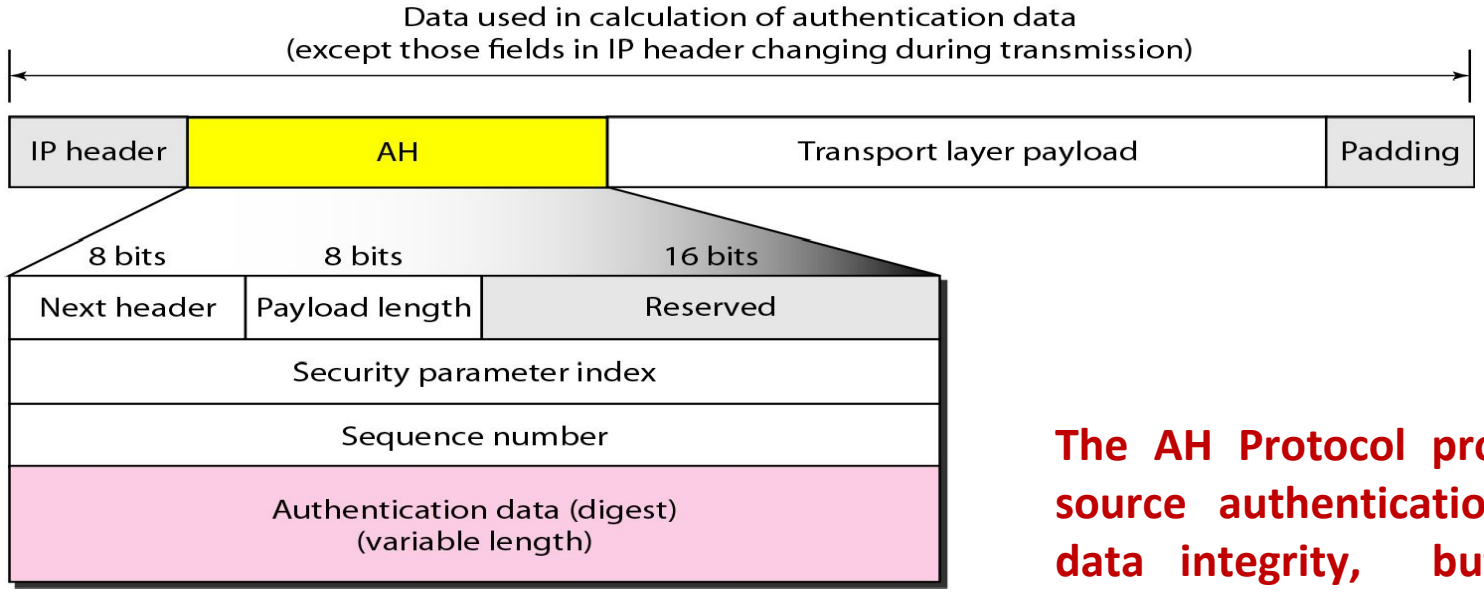
- Transport mode in action



- Tunnel Mode in action



Authentication Header (AH) Protocol in transport mode

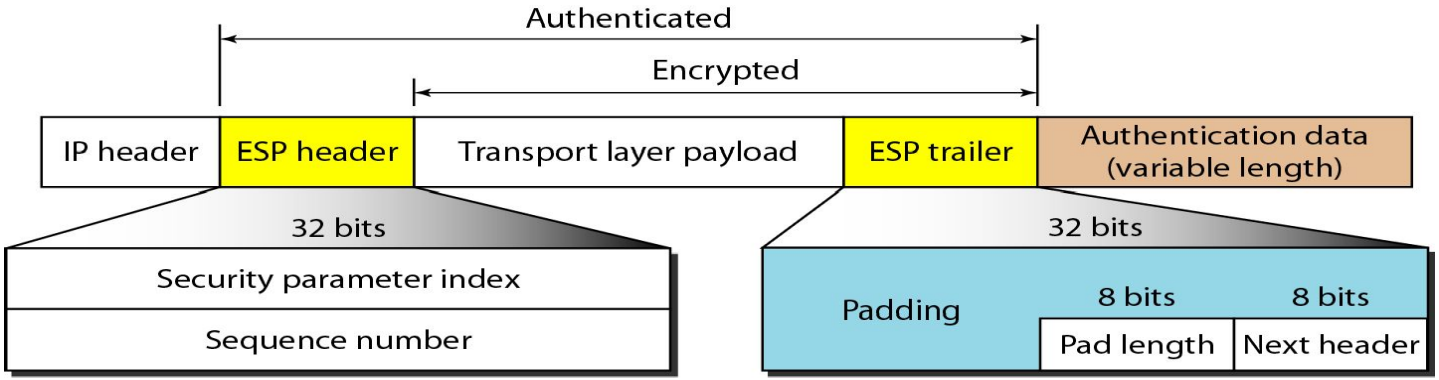


The AH Protocol provides source authentication and data integrity, but not privacy.

AH ICV calculation

- The AH ICV is computed over:
  - IP header fields that are either immutable in transit or that are predictable in value upon arrival at the endpoint for the AH SA, e.g., source address (immutable), destination address with source routing (mutable but predictable)
  - The AH header (Next Header, Payload Len, Reserved, SPI, Sequence Number, and the Authentication Data (which is set to zero for this computation), and explicit padding bytes (if any))
  - The upper level protocol data, which is assumed to be immutable in transit

Encapsulating Security Payload (ESP) Protocol in transport mode

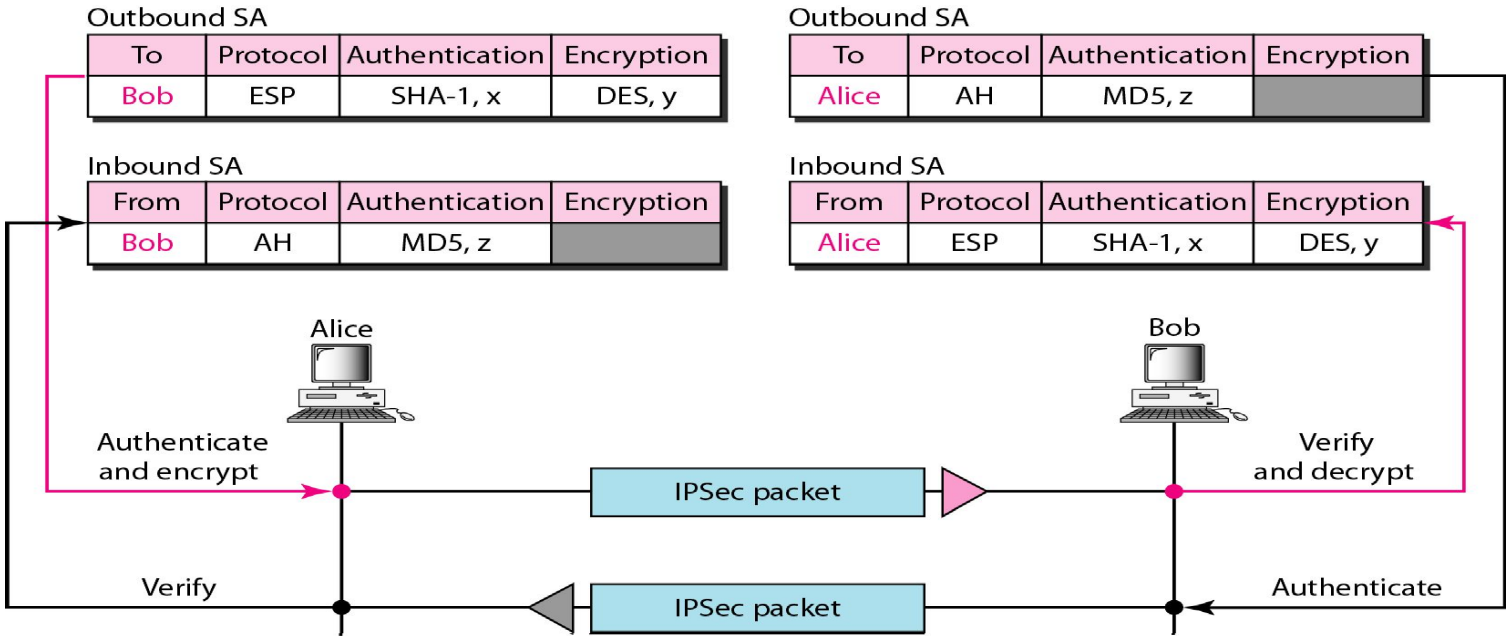


ESP provides source authentication, data integrity, and privacy.

ESP

- ESP Provides:
  - Message content confidentiality,
  - Data origin authentication,
  - Connectionless integrity,
  - Anti-replay service,
  - Limited traffic flow confidentiality
  - Services depend on options selected when establish Security Association (SA), net location
  - Can use a variety of encryption & authentication algorithms

Simple inbound and outbound security associations

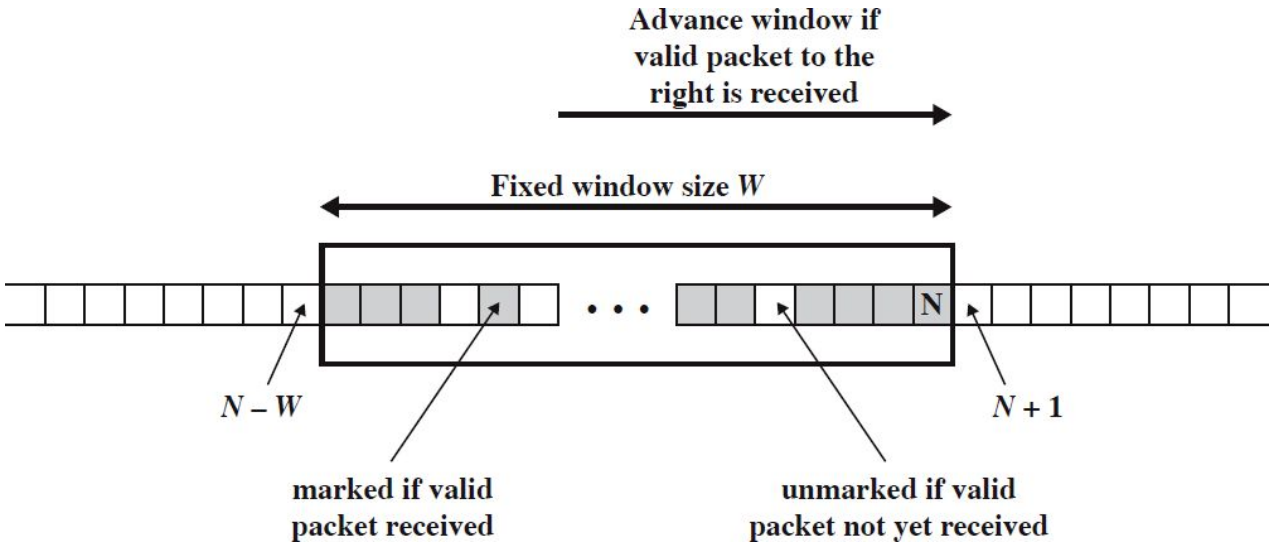


ESP

- ESP Provides:
  - Message content confidentiality,
  - Data origin authentication,
  - Connectionless integrity,
  - Anti-replay service,
  - Limited traffic flow confidentiality
  - Services depend on options selected when establish Security Association (SA), net location
  - Can use a variety of encryption & authentication algorithms

Anti-Replay Service

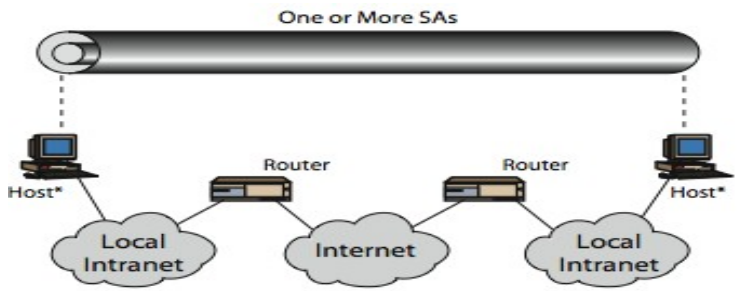
- Sender initializes sequence number to 0 when a new SA is established. Increment for each packet
- Receiver then accepts packets with sequence # within window of (N –W+1)



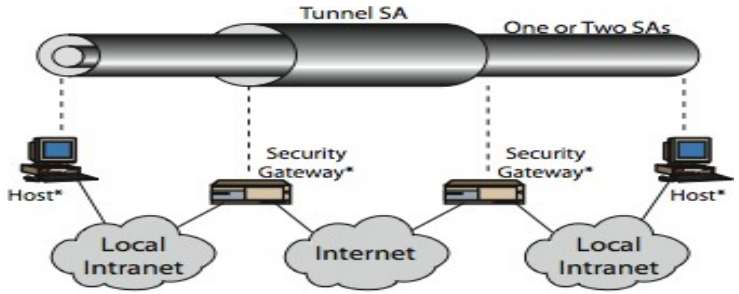
Combining Security Associations

- SAs can implement either AH or ESP
- To implement both need to combine SAs to form a security association bundle
- Transport adjacency: Outer AH over Inner ESP
- Iterated tunnelling: Multiple with different end points
  - All security between end-systems: AH Transport, ESP Transport, ESP inside AH transport, any one of the first 3 inside AH or ESP Tunnel
  - Between gateways (routers or firewalls): Single SA. No nesting.
  - Case 1 inside Case 2
  - Tunnel between a remote host and firewall. One or two SAs may be used as in Case 1.

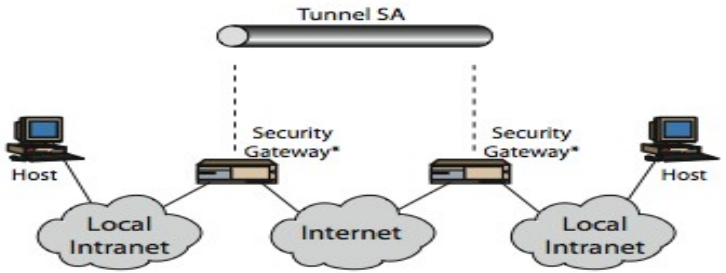
Combining Security Associations



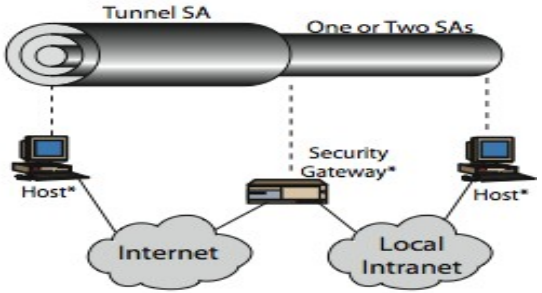
(a) Case 1



(c) Case 3

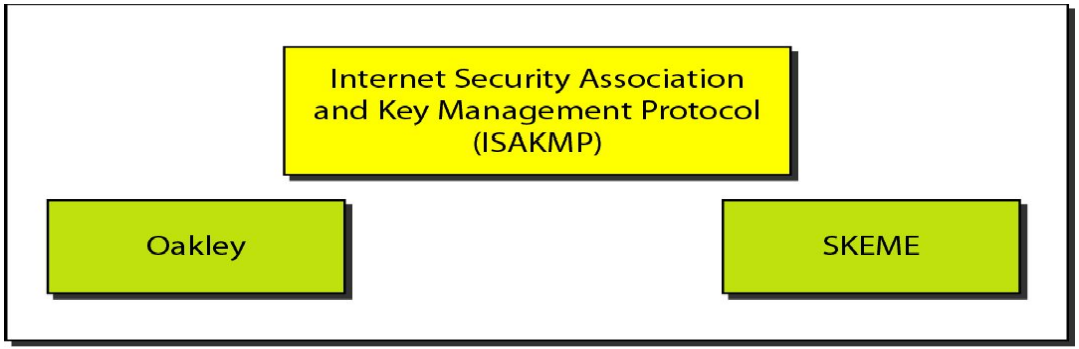


(b) Case 2



(d) Case 4

IKE components



IKE creates SAs for IPSec

IKE

- Handles key generation & distribution
- Typically need 2 pairs of keys
  - 2 per direction for integrity and confidentiality
- Manual key management
  - System admin manually configures every system
- Automated key management
  - Automated system for on demand creation of keys for SA's in large systems
  - Oakley key exchange and ISAKMP key management
  - IKEv2 no longer uses Oakley & ISAKMP terms, but basic functionality is same

Oakley

- A key determination protocol based on D-H key exchange
- Adds features to address weaknesses of D-H.
- D-H has no info on identities of parties, is subject to man-in-middle attack, is computationally expensive
- Oakley adds
  - Cookies to thwart DoS attacks
  - Several groups of pre-specified global parameters
  - Nonces to protect against replay
  - DH public key exchange with authentication using Digital signature, Public Key Encryption, or Symmetric Key Encryption
- Can use arithmetic in prime fields or elliptic curve fields

ISAKMP

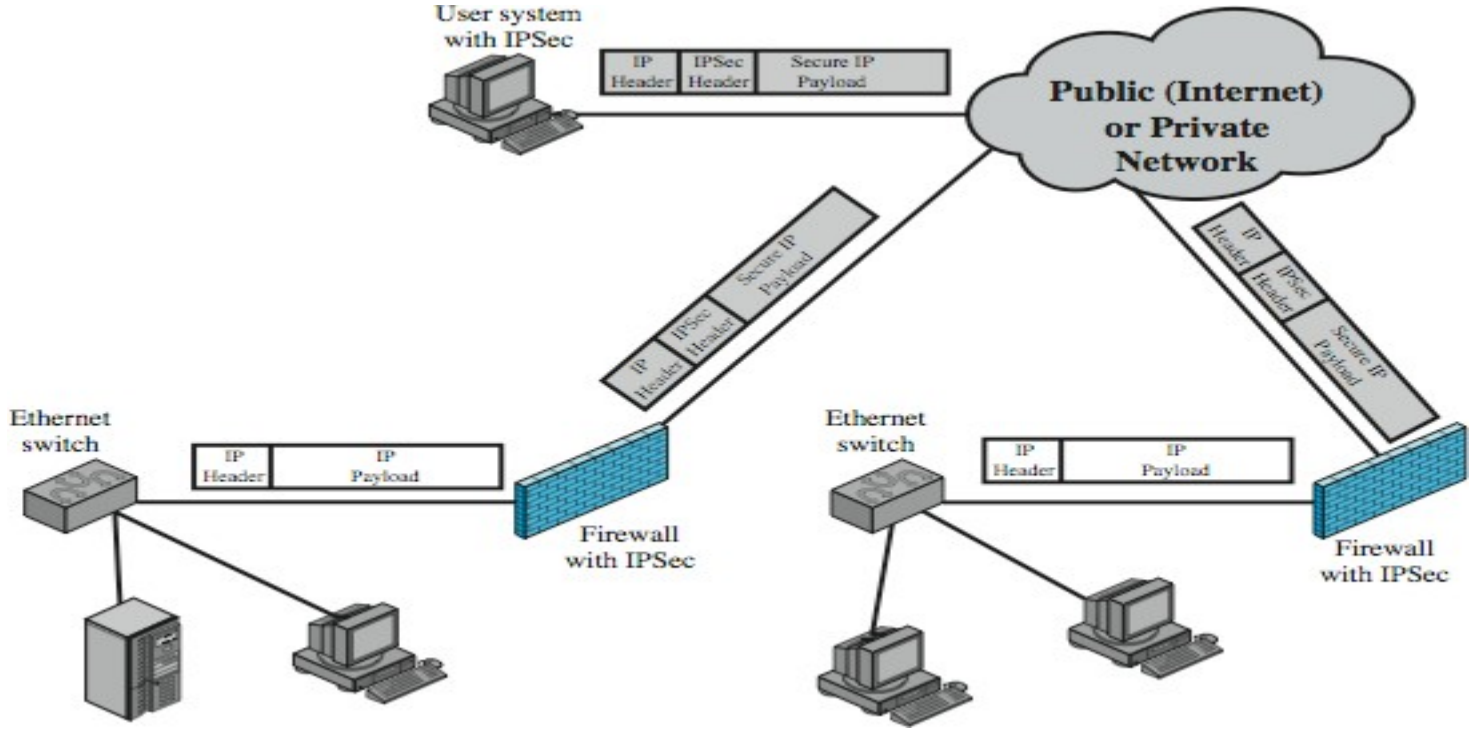
- Internet Security Association and Key Management Protocol
- Provides framework for key management
- Defines procedures and packet formats to establish, negotiate, modify, and delete SAs
- Independent of key exchange protocol, encryption algorithm, and authentication method

IPSec Conclusion

- IPSec provides authentication, confidentiality, and key management at Layer 3. Applies to all traffic.
- Security associations are one-way and can be bundled together.
- Authentication header for message authentication using HMAC
- Encapsulating security protocol (ESP) for confidentiality and/or integrity
- Both can be used end-to-end with original IP header inside (Tunnel) or without original IP header (Transport) mode
- Oakley is the IKE key determination protocol
- ISAKMP is the IKE key management protocol

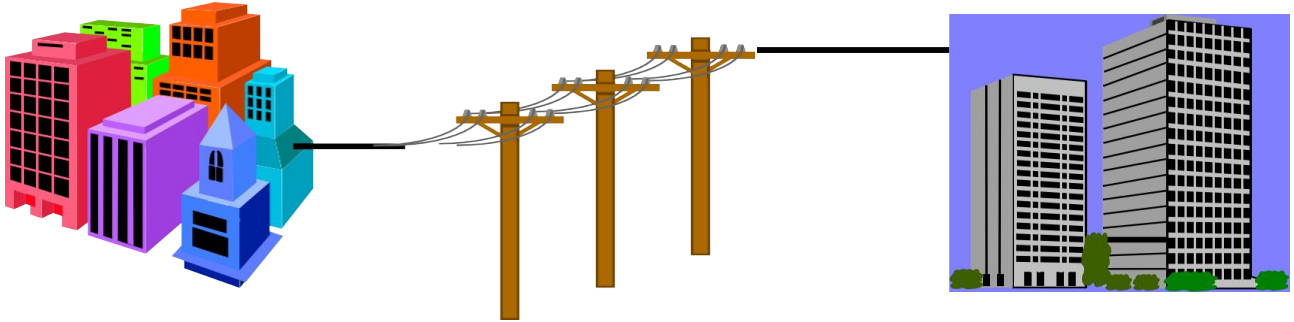
Virtual Private Network (VPN)

Virtual Private Networks

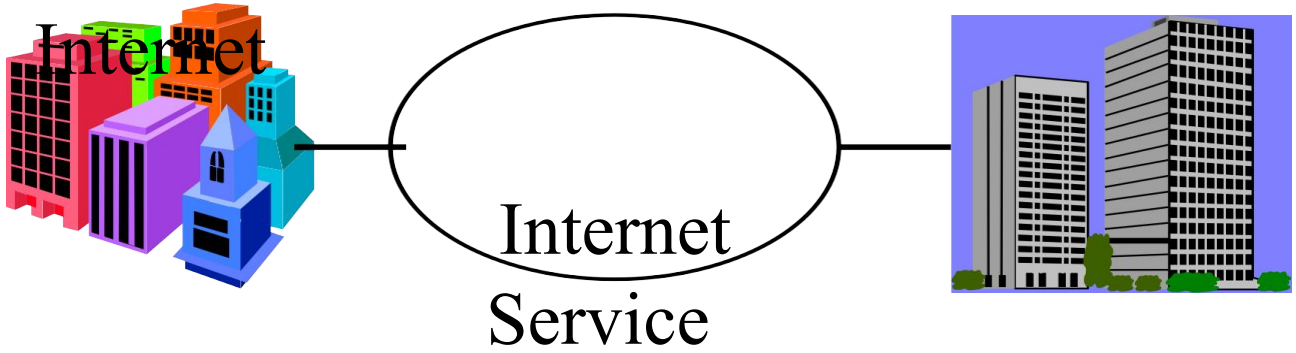


What is a VPN?

- Private Network: Uses leased lines

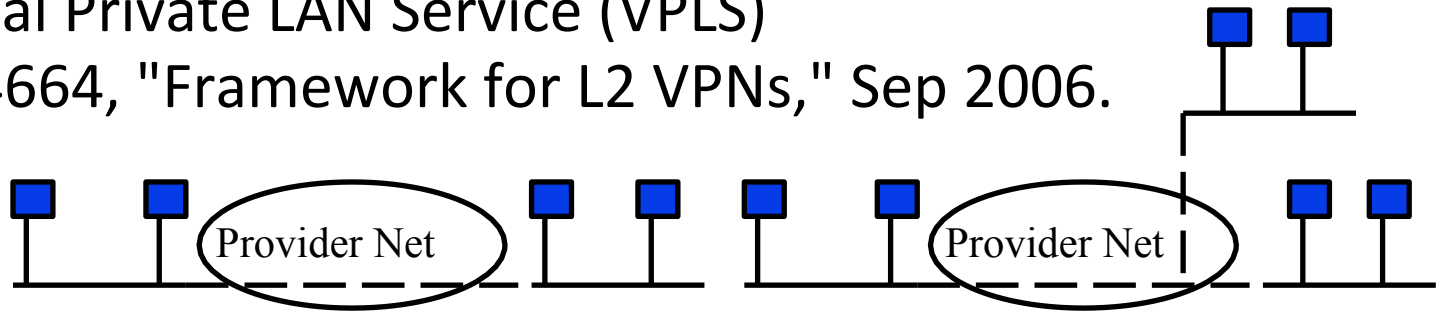


- *Virtual* Private Network: Uses public Internet



Layer 2 VPN

- Customers' Layer 2 packets are encapsulated and delivered at the other end
- Looks like the two ends are on the same LAN or same wire
- Provides Ethernet connectivity
- Works for all Layer 3 protocols
- Virtual Private Wire Service (VPWS)
- Virtual Private LAN Service (VPLS)
- RFC4664, "Framework for L2 VPNs," Sep 2006.



Layer 3 VPN

- Provides Layer 3 connectivity
- Looks like the two customer routers are connected
- Usually designed for IP packets

VPN Tunnelling Protocols

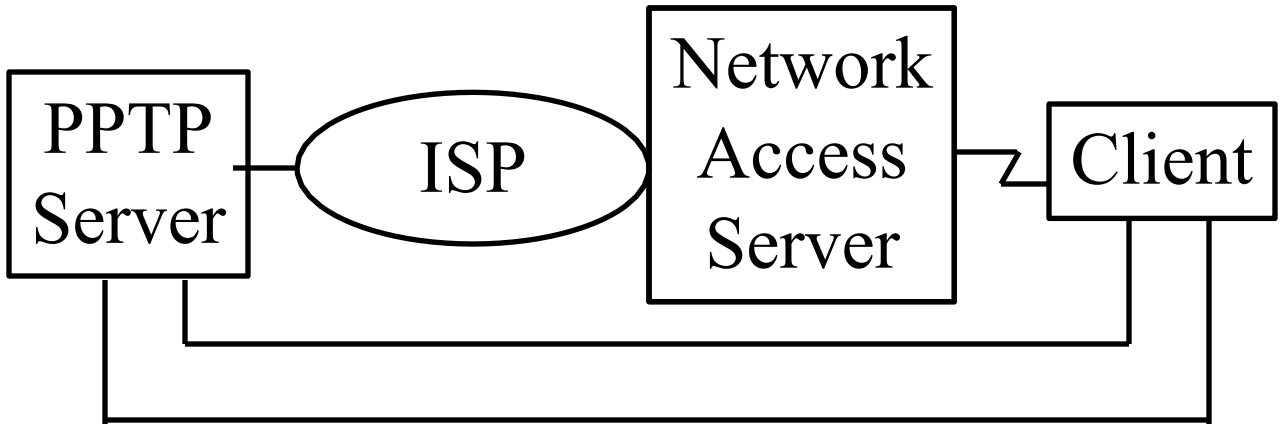
- GRE: Generic Routing Encapsulation (RFC 1701/2)
- PPTP: Point-to-point Tunnelling Protocol
- L2TP: Layer 2 Tunnelling protocol
- IPSec: Secure IP
- MPLS: Multiprotocol Label Switching

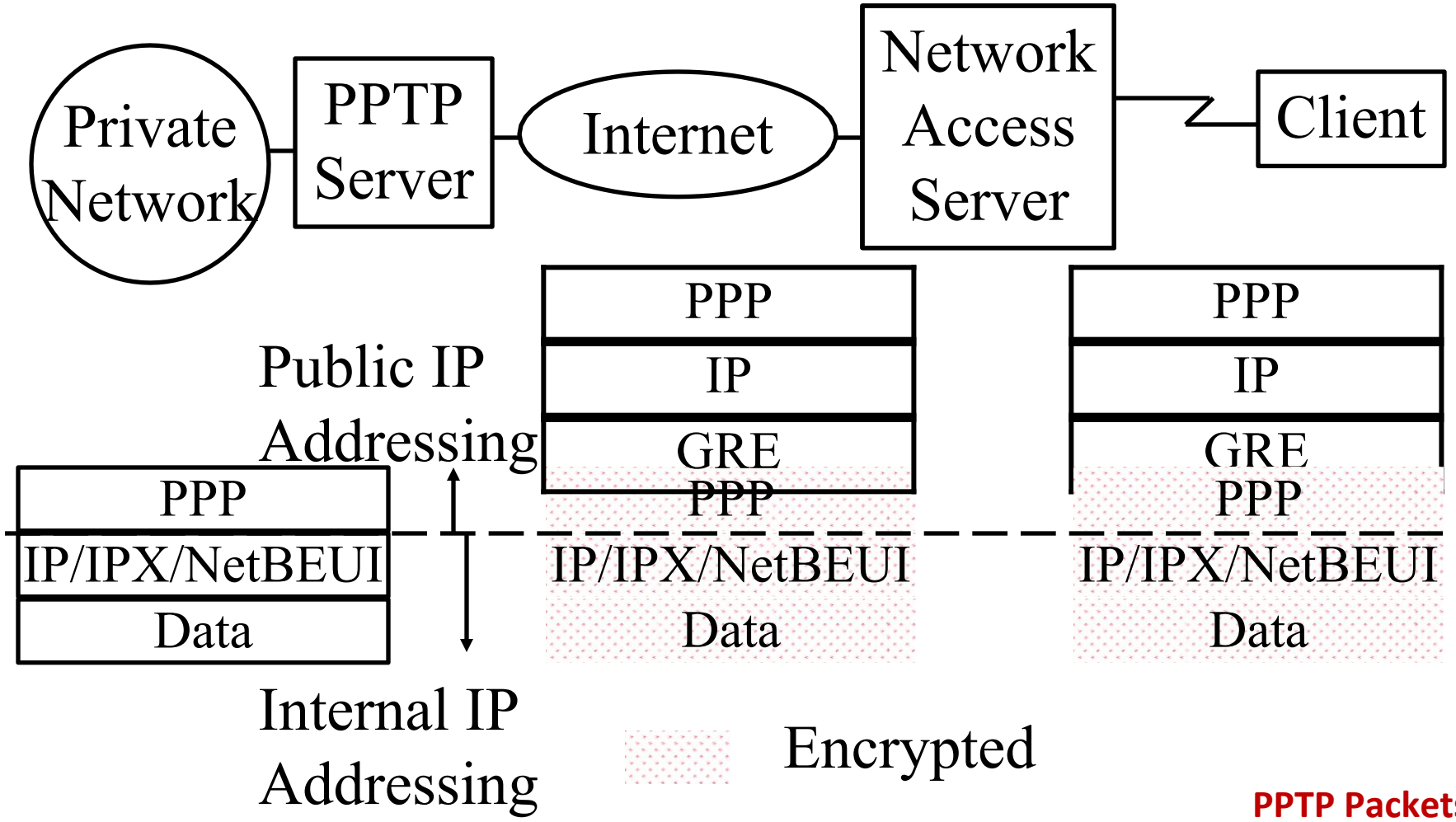
GRE

- Generic Routing Encapsulation (RFC 1701/1702)
- Generic □ X over Y for any X or Y
- Optional Checksum, Loose/strict Source Routing, Key
- Key is used to authenticate the source
- Over IPv4, GRE packets use a protocol type of 47
- Allows router visibility into application-level header
- Restricted to a single provider network □ end-to-end

PPTP Tunnel

- PPTP = Point-to-point Tunnelling Protocol
- Developed jointly by Microsoft, Ascend, USR, 3Com and ECI Telematics
- PPTP server for NT4 and clients for NT/95/98





PPTP Packets

L2TP

- Layer 2 Tunnelling Protocol
- L2F = Layer 2 Forwarding (From CISCO)
- L2TP = L2F + PPTP
- Combines the best features of L2F and PPTP
- Easy upgrade from L2F or PPTP
- Allows PPP frames to be sent over non-IP (Frame relay, ATM) networks also (PPTP works on IP only)
- Allows multiple (different QoS) tunnels between the same end-points. Better header compression. Supports flow control

L2TP v3

- Allows service providers to offer L2 VPN over IP network.
- L2TPv2 was for tunnelling PPP over packet switched data networks (PSDN)
- V3 generalizes it for other protocols over PSDN
- PPP specific header removed
- Can handle HDLC (High-Level Data Link Control), Ethernet, 802.1Q VLANs, Frame relay, packet over SONET (Synchronous Optical Network)

Open VPN

- Most popular open source VPN software for client and servers
- Can be implemented in firmware, e.g., DD-WRT, OpenWRT, …
- Available on most operating systems, e.g., Windows, Linus, Mac, IoS, Android, …
- Many routers come with OpenVPN support
- Does not use IKE, IPSec, PPTP, L2TP
- Uses OpenSSL library for SSL/TLS on TCP/UDP
- Provides all encryption/authentication methods in OpenSSL, e.g., pre-shared key, certificates, username/password, …
- OpenSSL allows servers to issue certificates to clients
- Extendable using modular plugins
- Uses a single TCP/UDP port can traverse NAT/firewalls23

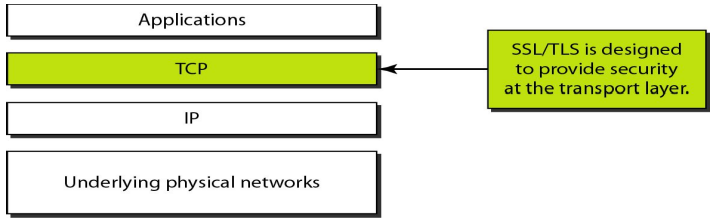
Configuring VPN

- View the video of VPN settings https://www.youtube.com/watch?v=UD03hRHcyEY
- Read about Tor (The Onion Router) from: https://en.wikipedia.org/wiki/Tor_(anonymity_network)
- Download and install Tor browser from: https://www.torproject.org/projects/torbrowser.html.en
- Open both Tor browser and your regular browser (Firefox or Internet Explorer, etc.)
- Browse to “WhatIsMyIP.com” on both browsers and capture the results.
- Repeat the previous step on both browsers and capture the results.
- Browse to http://thehiddenwiki.org/2013/08/23/list-of-onion

SSL/TLS and HTTPs

SSL/TLS

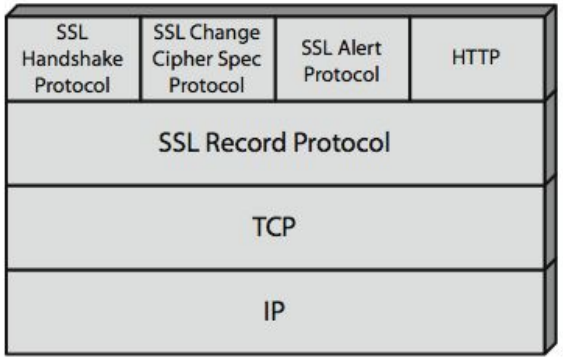
- Two protocols are dominant today for providing security at the transport layer: the Secure Sockets Layer (SSL) Protocol and the Transport Layer Security (TLS) Protocol.
- SSL/TLS provides the following services over TCP layer:
  - Crypto negotiation: Negotiate encryption and hash methods
  - Key Exchange: Secret key exchange using public key certificates
  - Privacy: Encryption using a secret key
  - Integrity: Message authentication using a keyed hash



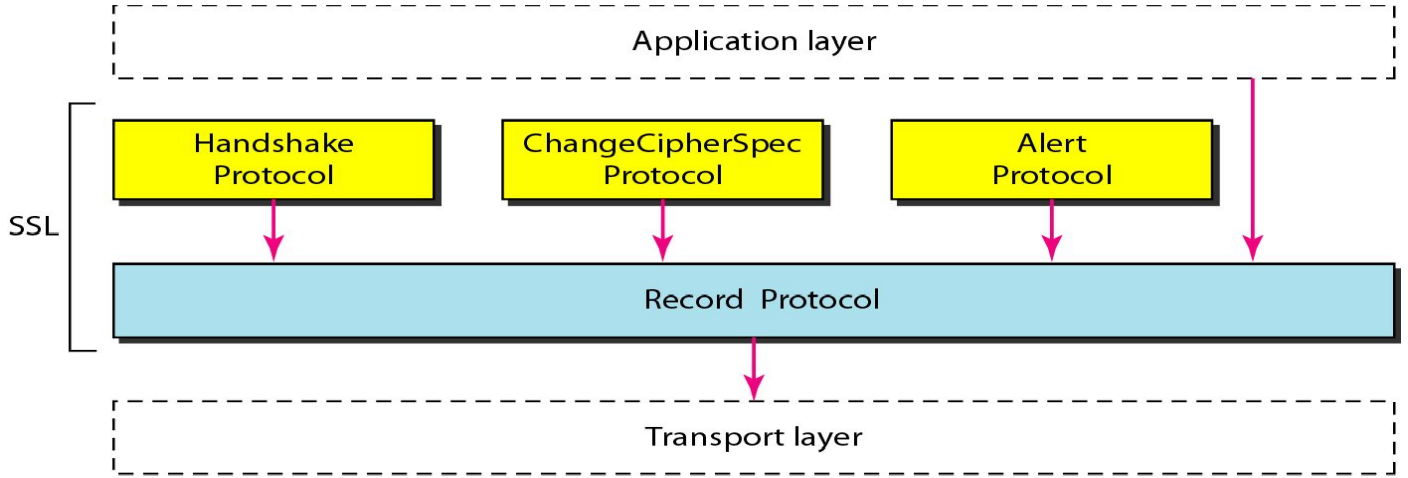
Location of SSL and TLS in the Internet model

SSL Architecture

- SSL has 4 components in two layers
  - Handshake protocol: Negotiates crypto parameters for an "SSL session" that can be used for many "SSL/TCP connections"
  - Record Protocol: Provides encryption and MAC
  - Alert protocol: To convey problems
  - Change Cipher Spec Protocol: Implement negotiated crypto parameters

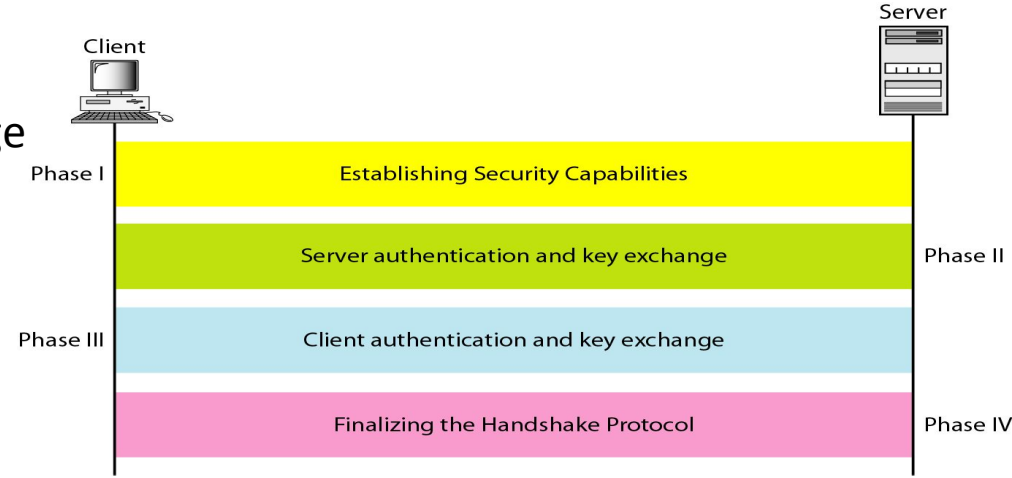


Four SSL protocols

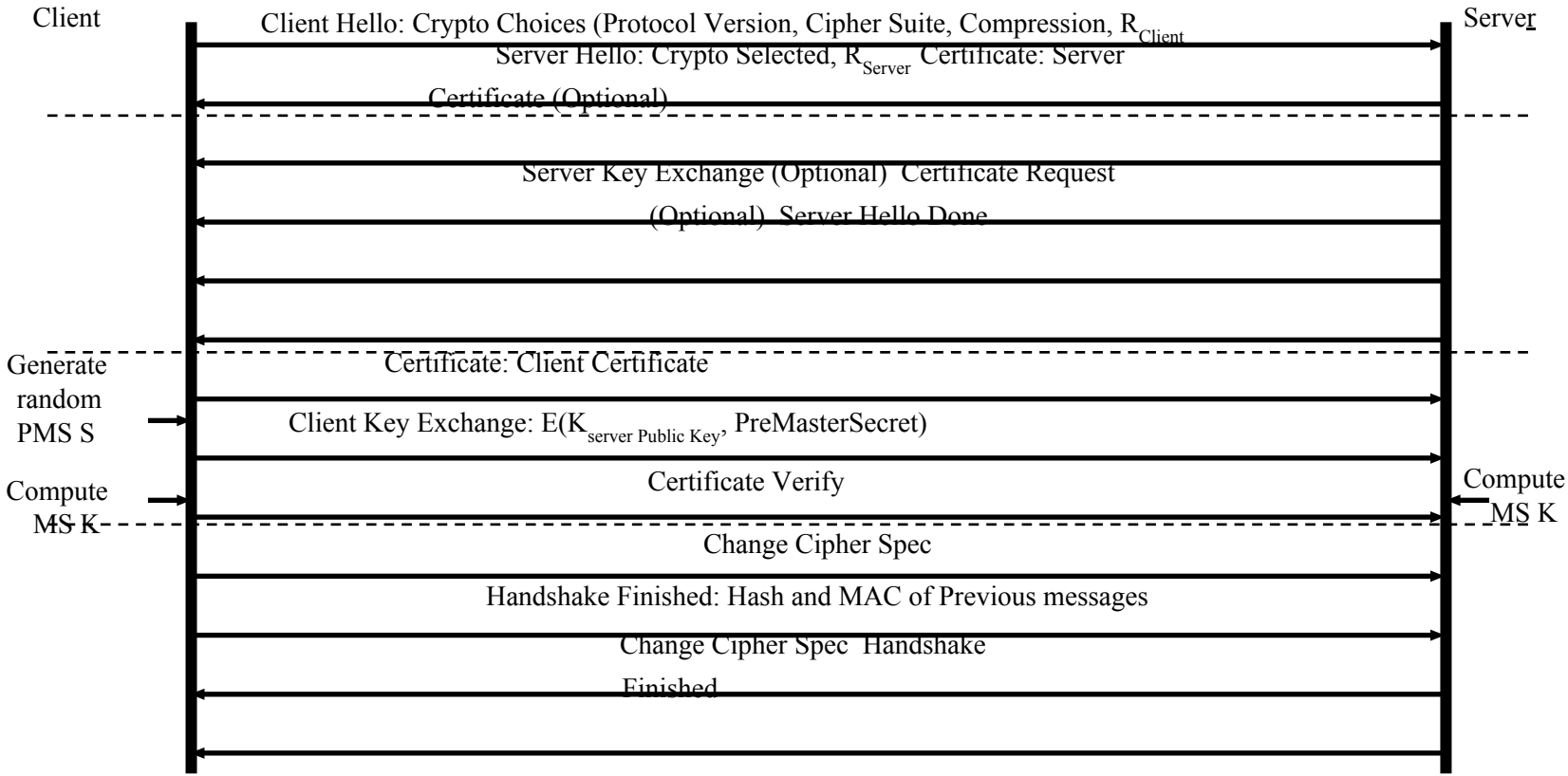


Handshake Protocol

- Allows server and client to:
  - Authenticate each other
  - To negotiate encryption & MAC algorithms
  - To negotiate cryptographic keys to be used
- Comprises a series of messages in phases
  - Establish Security Capabilities
  - Server Authentication
  - Client Authentication and Key Exchange
  - Finish



SSL Handshake Protocol Action



Handshake Message

All messages are Type-Length-Value (TLV) encoded. Types

1 = Client Hello: Highest Version Supported, RClient, Session ID, Cipher Suites, Compressions

2 = Server Hello: Version Accepted, RServer, Session ID, Chosen Cipher, Chosen Compression

14 = Server Hello Done

16 = Client Key Exchange: Encrypted pre-master key

12 = Server Key Exchange: Modulus p, Exponent g, Signature (export only) 13 = Certificate Request: CA Names (requested by the server)

11 = Certificate: sent by the server

15 = Certificate Verify: Signature of Hash of messages

20 = Handshake Finished: MD5 and SHA Digest of message halves

Security Capability Negotiation

- Key-Exchange Methods:
  - RSA
  - Fixed D-H: Shared secret generated using fixed public keys
  - Ephemeral D-H: Ephemeral = Temporary, one-time secret key is generated after certificate exchange and authentication
  - Anonymous D-H: No authentication. Only public key exchange. Subject to MITM attack
  - Fortezza: Using PC-Cards (http://en.wikipedia.org/wiki/Fortezza)
- CipherSpec:
  - Cipher Algorithm: RC4, RC2, DES, 3DES, DES40, IDEA, or Fortezza
  - MAC Algorithm: MD5 or SHA-1
  - CipherType: Stream or Block
  - IsExportable: True or False
  - HashSize: 0, 16 (for MD5), or 20 (for SHA-1) bytes
  - Key Material: info used to generate keys
  - IV Size: Size of IV for CBC

Cryptographic Computations

- Master secret creation
  - A one-time 48-byte value based on nonces
  - A 48-byte pre-master secret is exchanged/generated using secure key exchange (RSA / Diffie-Hellman) and then hashing:
  - **Master_Secret = MD5(Pre_master_Secret || SHA ('A' || pre_master_secret || clientHello.random || ServerHello.random)) || MD5(Pre_master_Secret || SHA('BBB' || pre_master_secret || clientHello.random || ServerHello.random)) || MD5(Pre_master_Secret || SHA('CCC' || pre_master_secret || clientHello.random || ServerHello.random))**
- Generation of cryptographic parameters
  - Client write MAC secret, a server write MAC secret, a client write key, a server write key, a client write IV, and a server write IV
  - Generated by hashing master secret

SSL Change Cipher Spec Protocol

- A single 1-byte message
- Causes negotiated parameters to become current
- Hence updating the cipher suite in use

(a) Change Cipher Spec Protocol

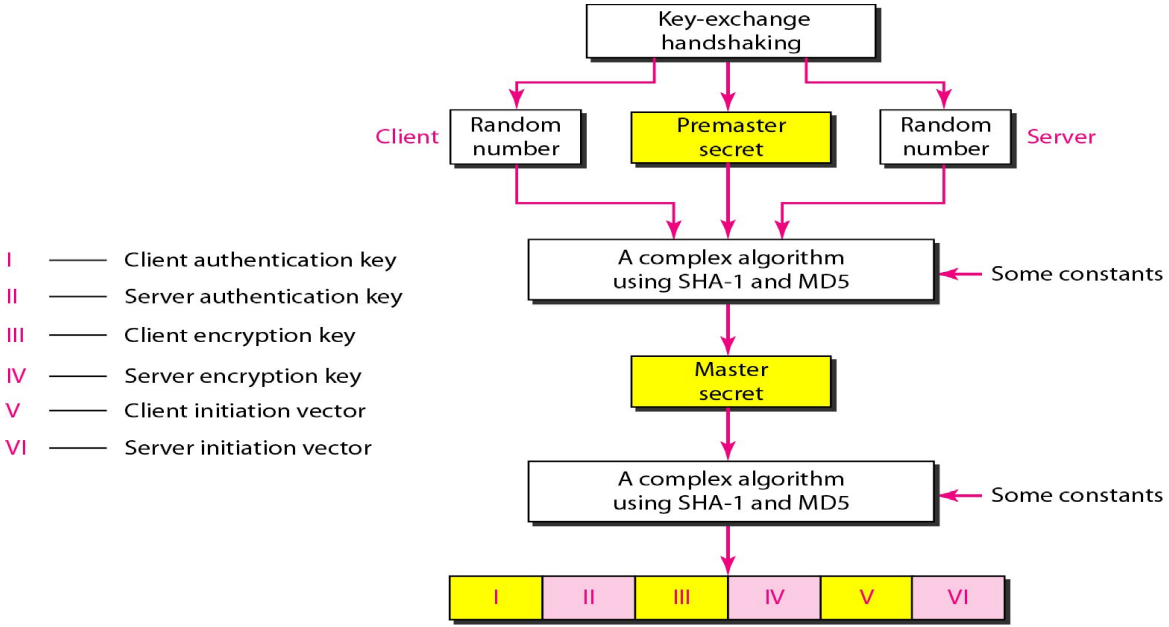
SSL cipher suite list

Cipher Suite	*Key Exchange Algorithm*	*Encryption Algorithm*	*Hash Algorithm*
SSL_NULL_***WITH***_**NULL**_NULL	NULL	NULL	NULL
SSL_RSA_***WITH***_**NULL**_MD5	RSA	NULL	MD5
SSL_RSA_***WITH***_**NULL**_SHA	RSA	NULL	SHA
SSL_RSA_***WITH***_**RC4_128**_MD5	RSA	RC4_128	MD5
SSL_RSA_***WITH***_**RC4_128**_SHA	RSA	RC4_128	SHA
SSL_RSA_***WITH***_**IDEA_CBC**_SHA	RSA	IDEA_CBC	SHA
SSL_RSA_***WITH***_**DES_CBC**_SHA	RSA	DES_CBC	SHA
SSL_RSA_***WITH***_**3DES_EDE_CBC**_SHA	RSA	3DES_EDE_CBC	SHA
SSL_DH_anon_***WITH***_**RC4_128**_MD5	DH_anon	RC4_128	MD5
SSL_DH_anon_***WITH***_**DES_CBC**_SHA	DH_anon	DES_CBC	SHA
SSL_DH_anon_***WITH***_**3DES_EDE**_CBC_SHA	DH_anon	3DES_EDE_CBC	SHA

SSL cipher suite list (continued)

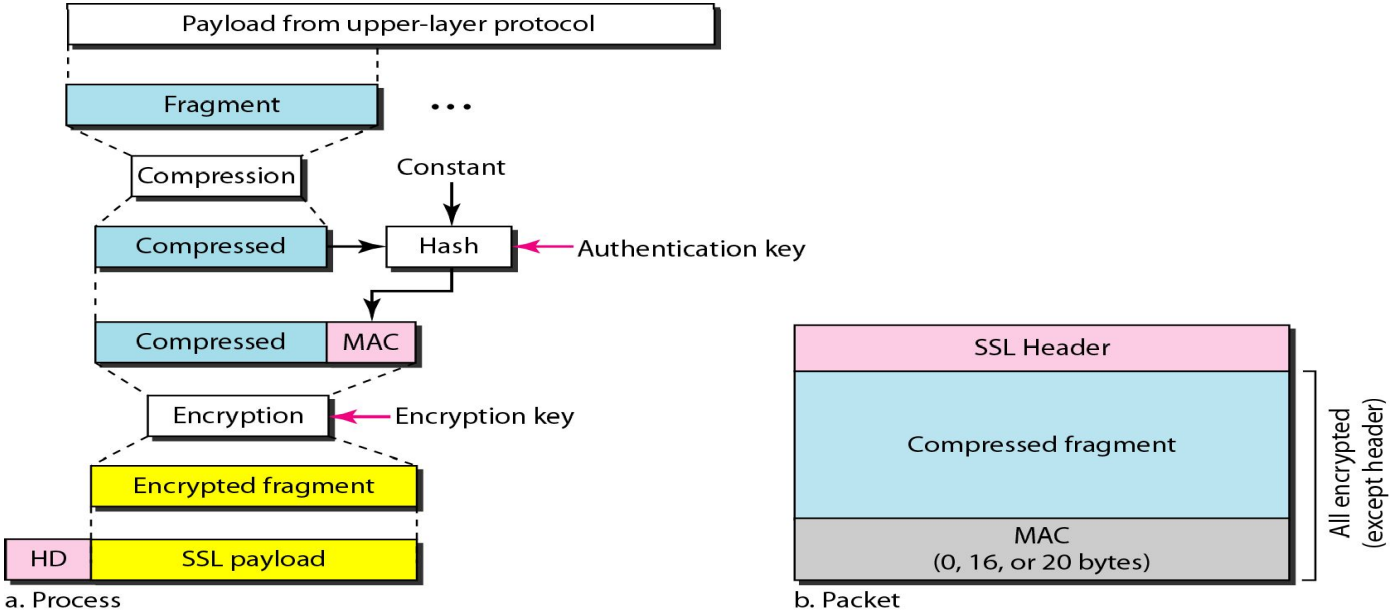
Cipher Suite	*Key Exchange Algorithm*	*Encryption Algorithm*	*Hash Algorithm*
SSL_DHE_RSA_*WITH*_DES_CBC_SHA	DHE_RSA	DES_CBC	SHA
SSL_DHE_RSA_*WITH*_3DES_EDE_CBC_SHA	DHE_RSA	3DES_EDE_CBC	SHA
SSL_DHE_DSS_*WITH*_DES_CBC_SHA	DHE_DSS	DES_CBC	SHA
SSL_DHE_DSS_*WITH*_3DES_EDE_CBC_SHA	DHE_DSS	3DES_EDE_CBC	SHA
SSL_DH_RSA_*WITH*_DES_CBC_SHA	DH_RSA	DES_CBC	SHA
SSL_DH_RSA_*WITH*_3DES_EDE_CBC_SHA	DH_RSA	3DES_EDE_CBC	SHA
SSL_DH_DSS_ *WITH*_DES_CBC_SHA	DH_DSS	DES_CBC	SHA
SSL_DH_DSS_ *WITH*_3DES_EDE_CBC_SHA	DH_DSS	3DES_EDE_CBC	SHA
SSL_FORTEZZA_DMS_*WITH*_NULL_SHA	FORTEZZA_DMS	NULL	SHA
SSL_FORTEZZA_DMS_*WITH*_FORTEZZA_CBC_SHA	FORTEZZA_DMS	FORTEZZA_CBC	SHA
SSL_FORTEZZA_DMS_*WITH*_RC4_128_SHA	FORTEZZA_DMS	RC4_128	SHA

Creation of cryptographic secrets in SSL



The client and the server have six different cryptography secrets.

Processing done by the Record Protocol



TLS (Transport Layer Security)

- IETF standard RFC 2246 similar to SSLv3
- With minor differences
  - In record format version number
  - Uses HMAC for MAC
  - A pseudo-random function expands secrets
  - Based on HMAC using SHA-1 or MD5
  - Has additional alert codes
  - Some changes in supported ciphers
  - Changes in certificate types & negotiations
  - Changes in crypto computations & padding

HTTPs

- HTTPS (HTTP over SSL)
  - Combination of HTTP & SSL/TLS to secure communications between browser & server
    - Documented in RFC2818
    - No fundamental change using either SSL or TLS
- Use https:// URL rather than http://
  - And port 443 rather than 80
- Encrypts URL, document contents, form data, cookies, HTTP headers

HTTPs Uses

- Connection initiation
  - TLS handshake then HTTP request(s)
- Connection closure
  - Have “Connection: close” in HTTP record
  - TLS level exchange close_notify alerts
  - Can then close the TCP connection
  - Must handle abnormal TCP close before alert exchange sent or completed

References

- http://www.cse.wustl.edu/~jain/cse571-17
- Forouzan, B. A., COOMBS, C., & FEGAN, S. C. (1998). Data communications and networking. Language, 32(908), 23cm.
- Internet security http://www.cis.syr.edu/~wedu/Teaching/cis758/readings.html
- Open Learn https://www.open.edu/openlearn/ocw/mod/oucontent/view.php?id=48261
- IDPS https://cloudacademy.com/course/intrusion-detection-and-prevention-on-amazon-web-services/ids-ips-in-detail-1/
- G. Dileep Kumar, "Network Security Attacks and Countermeasures", IGI Global, 2016.