Unit 1
Network layer Security & Transport Layer Security

Chamundeswari Arumugam
Professor
SSN College of Engineering, Chennai

June 2017

Outline

- Network layer security
  1. IPSec Protocol
  2. Security Association
  3. HMAC
  4. IP Authentication Header
  5. IP ESP
  6. Key Management Protocol for IPSec
- Transport layer Security
  1. SSL protocol
  2. Cryptographic Computations
  3. TLS Protocol
- Reference

IPsec protocol

- IPSec is an Internet standard for network layer security

Basic Documents

- Architecture, ESP, AH, Encryption algorithm, Authentication algorithm, Key management, DOI

Basic components

- Security Protocols for AH and ESP
- Security Associations for policy management and traffic processing
- Manual and automatic key management for the Internet Key Exchange (IKE), the Oakley key determination protocol and ISAKMP.
- Algorithms for authentication and encryption

Fig : IPsec Document overview

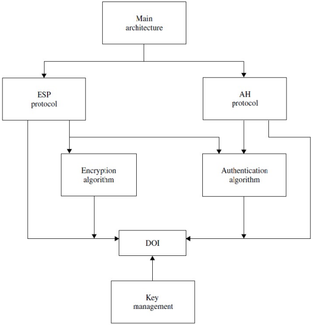
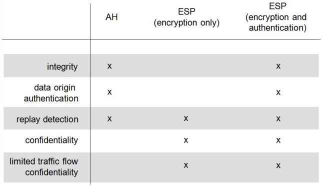


Fig : IPsec Services



	AH	ESP (encryption only)	ESP (encryption and authentication)
integrity	x		x
data origin authentication	x		x
replay detection	x	x	x
confidentiality		x	x
limited traffic flow confidentiality		x	x

Security Associations (SAs)

- An SA is a simplex connection between a sender and receiver that affords security services.
- An SA is used either for AH or for ESP but never for both

SA Three parameters

- Security Parameters Index (SPI) a bit string assigned to the SA carried in AH and ESP headers to allow the receiving party to select the SA which must be used to process the packet
- IP destination address of an end-system or a network element (e.g., router)
- Security protocol identifier indicates whether the SA is an AH or an ESP SA

Two nominal databases

- Security Policy Database (SPD) : specifies the policies that determine the disposition of all IP traffic inbound or outbound from a host or security gateways
- Security Association Database (SAD) : contains parameters that are associated with each security association.

Two modes of operation

- Transport
  1. Provides protection primarily for upper-layer protocols.
  2. AH in transport mode authenticates the IP payload and the protection is also extended to selected portions of the IP header, selected portions of IPv6 extension headers and the selected options.
  3. ESP in transport mode primary encrypts and optionally authenticates the IP payload but not the IP header.
- Tunnel
  1. Provides protection to the entire IP packet.
  2. AH in tunnel mode authenticates the entire inner (original) IP packet and selected portions of the outer (security field of AH)IP header.
  3. ESP in tunnel mode encrypts and optionally authenticates the entire inner IP packet, including the inner IP header.

Hashed Message Authentication Code (HMAC)

Introduction

- HMAC is a secret-key authentication algorithm which provides both data integrity and data origin authentication for packets sent between two parties.
- MD5, and SHA-1 are examples of such hash functions, which works faster. SHA-1 appears to be a cryptographically stronger function.

HMAC Equation Explanation

- Append zeros to the end of K to create a b-byte string (i.e. if $K = 160$ bits in length and $b = 512$ bits, then K will be appended with 352 zero bits or 44 zero bytes 0x00).
- XOR (bitwise exclusive-OR) K with ipad to produce the b-bit block computed in step 1.
- Append M to the b-byte string resulting from step 2.
- Apply H to the stream generated in step 3.
- XOR (bitwise exclusive-OR) K with opad to produce the b-byte string computed in step 1.
- Append the hash result H from step 4 to the b-byte string resulting from step 5.
- Apply H to the stream generated in step 6 and output the result.

HMAC Structure

- H denotes a hash function where the message is hashed by iterating a basic compression function on data blocks
- Let K denote secret key, and b denote the block length of 64 bytes or 512 bits for all hash functions such as MD5 and SHA-1.
- Let h denotes the length of hash values, i.e. h = 16 bytes or 128 bits for MD5 and 20 bytes or 160 bits for SHA-1.
- To compute HMAC over the message, the HMAC equation is expressed as follows.
- $HMAC = H[(K \oplus opad) \parallel H[(K \oplus ipad) \parallel M]]$
- $ipad = 00110110(0x36)$ repeated 64 times (512 bits)
- $opad = 01011100(0x5c)$ repeated 64 times (512 bits)
- ipad is inner padding opad is outer padding

Hashed Message Authentication Code (HMAC)(contd..)

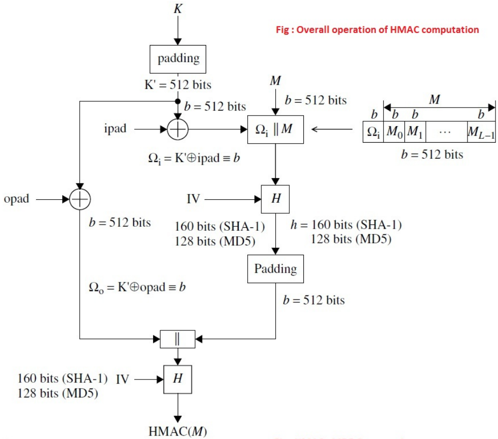
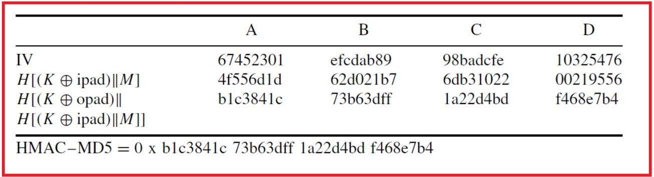


Fig : Overall operation of HMAC computation

Fig : HMAC - MD5 Computation



	A	B	C	D
IV	67452301	efcdab89	98badcfe	10325476
$H[(K \oplus \text{ipad})\|M]$	4f556d1d	62d021b7	6db31022	00219556
$H[(K \oplus \text{opad})\|$ $H[(K \oplus \text{ipad})\|M]]$	b1c3841c	73b63dff	1a22d4bd	f468e7b4

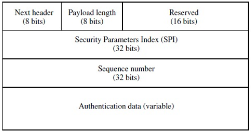
HMAC−MD5 = 0 x b1c3841c 73b63dff 1a22d4bd f468e7b4

IP Authentication Header

Autentication Header

- IP AH is used to provide data integrity and authentication for IP packets
- AH provides authentication for the IP header, as well as for upper-level protocol (TCP, UDP) data.
- Security services can be provided between a pair of hosts, between a pair of security gateway or between a security gateway and a host.
- Authentication is based on the use of an MAC or the Integrity Check Value (ICV) computation so that two hosts must share a secret key.

Fig : IPsec AH format



Next header (8 bits)	Payload length (8 bits)	Reserved (16 bits)
Security Parameters Index (SPI) (32 bits)		
Sequence number (32 bits)		
Authentication data (variable)		

AH Format

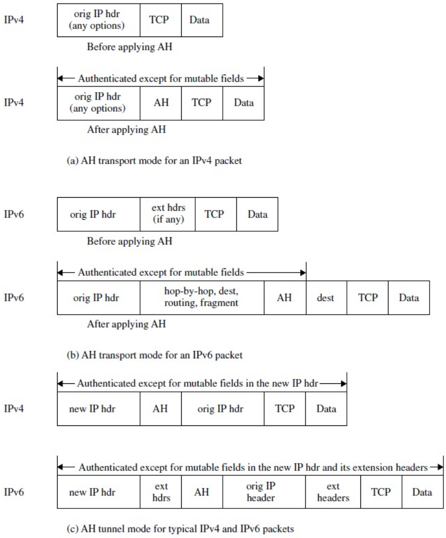
Type	Description
Next header (8 bits)	type of header immediately following this header (e.g., TCP, IP, etc.)
Payload length	length of AH (in 32 bit words) minus 2. e.g., 4 if Authentication data is 3x32 bits long
Reserved (16 bits	This field is reserved for future use. It must be set to zero.
SPI (32 bits)	identifies the SA used to generate this header.
Sequence number (32 bits)	The first packet has sequence number 1 using a given SA. If anti-replay is enabled, the sender checks the counter has not cycled before inserting the new value in the sequence number field. If the counter has cycled, the sender will set up a new SA and key. If the anti-replay is disabled, when the counter reaches the maximum value, the counter rolls over to zero.
Authentication data (variable):	This variable length field must be an integral multiple of 32-bit words that contains the Integrity Check Value (ICV) or MAC for this packet. It may include explicit padding to ensure that the length of AH is an integral multiple of 32 bits (IPv4) or 64 bits (IPv6).

IP Authentication Header (contd..)

AH Location

- Transport model AH
  - AH is inserted after the IP header and before an upper layer protocol (TCP, UDP or ICMP)
  - IPv4 context : Authentication covers the entire packet, excluding mutable fields in the IPv4 header that are set to zero for MAC computation. It is illustrated in Figure (a).
  - IPv6 context : Authentication again covers the entire packet, excluding mutable fields that are set to zero for MAC computation. It is illustrated in Figure (b)
- Tunnel mode AH
  - AH protects the entire inner IP packet, including the entire inner IP header.
  - Figure.(c) illustrates AH tunnel mode positioning for typical IPv4 and IPv6 packets.

Fig : Transport mode and tunnel mode for AH authentication



IPv4: orig IP hdr (any options) | TCP | Data

Before applying AH

Authenticated except for mutable fields

IPv4: orig IP hdr (any options) | AH | TCP | Data

After applying AH

(a) AH transport mode for an IPv4 packet

IPv6: orig IP hdr | ext hdrs (if any) | TCP | Data

Before applying AH

Authenticated except for mutable fields

IPv6: orig IP hdr | hop-by-hop, dest, routing, fragment | AH | dest | TCP | Data

After applying AH

(b) AH transport mode for an IPv6 packet

Authenticated except for mutable fields in the new IP hdr

IPv4: new IP hdr | AH | orig IP hdr | TCP | Data

Authenticated except for mutable fields in the new IP hdr and its extension headers

IPv6: new IP hdr | ext hdrs | AH | orig IP header | ext headers | TCP | Data

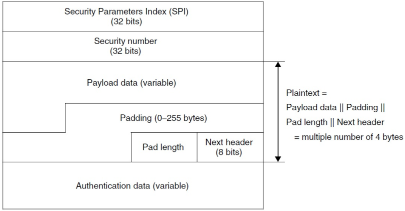
(c) AH tunnel mode for typical IPv4 and IPv6 packets

IP ESP

Introduction

- ESP header is designed to provide security services in IPv4 and IPv6.
- Security services are provided between a pair of hosts, between a pair of security gateways or between a security gateway and a host.
- ESP header is inserted before the upper-layer protocol header (transport mode) or before an encapsulated IP header (tunnel mode).
- Provide confidentiality (encryption), data authentication, integrity and anti-replay service, and limited traffic flow confidentiality.

Fig : IPsec ESP format



Security Parameters Index (SPI)
(32 bits)

Security number
(32 bits)

Payload data (variable)

Padding (0–255 bytes)

Pad length

Next header
(8 bits)

Authentication data (variable)

Plaintext =
Payload data || Padding ||
Pad length || Next header
= multiple number of 4 bytes

ESP Packet Format

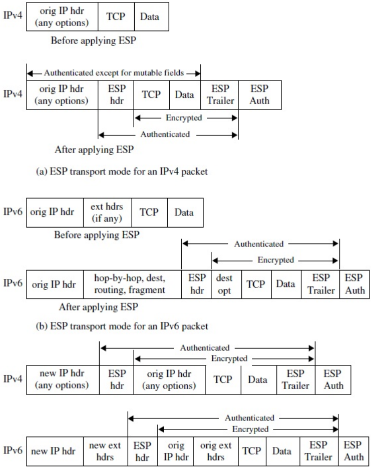
Type	Description
SPI (32 bits)	identifies an SA for this datagram.
Sequence number(32 bits)	Increasing counter value that provides an anti-replay function. Anti-replay is manadatory enabled, the transmitted sequence number must not be allowed to cycle. Senders counter and the receivers counter must be reset prior to the transmission of the 232nd packet on an SA.
Payload data (variable)	This variable-length field contains data described by the next header field.
Padding	padding field is used to fill the plaintext to the size required by the algorithm.
Pad length	This field indicates the number of pad bytes immediately preceding it.
Next header (8 bits)	This field identifies the type of data contained in the payload data field
Authentication data (variable)	Length of this optional field will be included if authentication service has been selected for the SA in question.

IP ESP (Contd..)

ESP Header Location

- Transport mode
  - Provides protection for upper protocols, but not the IP header.
  - IPv4 context, ESP is placed after the IP header, but before the upper-layer protocol. Figure.(a) illustrates this concept.
  - IPv6 context, the ESP appears after hop-by-hop, routing and fragmentation extension headers.ESP protects only fields after the ESP header. Figure (b) illustrates this concept.
- Tunnel mode
  - Employed in either hosts or security gateways
  - security gateway : protect subscriber transit traffic
  - Inner IP header carries the ultimate source and destination addresses, while an outer IP header may contain different IP addresses such as addresses of security gateways.
  - ESP protects the entire inner IP packet, including the entire inner IP header.
  - Figure (c) illustrates for typical IPv4 and IPv6 packets.

Fig : Transport mode and tunnel mode for ESP authentication.



(a) ESP transport mode for an IPv4 packet

(b) ESP transport mode for an IPv6 packet

(c) ESP tunnel mode for typical IPv4 and IPv6 packets

Key Management Protocol for IPsec

Key Management

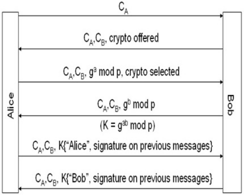
- Automated : Oakley Key Determination Protocol, Internet Security Association and Key Management Protocol (ISAKMP)

Oakley Key Determination Protocol

- A key exchange protocol based on Diffie-Hellman
- Provides added security (e.g., authentication)
- Oakley can be used directly over the IP protocol or over UDP protocol.
- Added security features of Oakley
  - cookie exchange to thwart clogging

    attacks :
    - hash(src IP addr, dst IP addr, src UDP port, dst UDP port, local secret)
    - local secret is periodically changed
  - uses nonces to detect replay attacks
- Oakley message fields correspond to ISAKMP message payloads.
- Oakley is the actual instantiation of ISAKMP framework for IPsec key and SA generation.

Fig : Sample Oakley message



C_A
C_A,C_B, crypto offered
C_A,C_B, g^a mod p, crypto selected
Alice
Bob
C_A,C_B, g^b mod p
(K = g^{ab} mod p)
C_A,C_B, K{"Alice", signature on previous messages}
C_A,C_B, K{"Bob", signature on previous messages}

I → R: CKY_i | 0 | OK_KEYX | GRP | g^x | EHAO

R → I: CKY_r | CKY_i | OK_KEYX | GRP | g^y | EHAS

I → R: CKY_i | CKY_r | OK_KEYX | GRP | g^x | NIDP | ID_i | ID_r | $\{N_i\}_{K_r}$

R → I: CKY_r | CKY_i | OK_KEYX | GRP | NIDP | $\{ N_r \mid N_i \}_{K_i}$ | ID_r | ID_i |
MAC(K_{ir}, ID_r | ID_i | GRP | g^y | g^x | EHAS)

I → R: CKY_i | CKY_r | OK_KEYX | GRP | NIDP |
MAC(K_{ir}, ID_i | ID_r | GRP | g^x | g^y | EHAS)

where

- CKY: cookie
- OK_KEYX: message type is Oakley key exchange
- GRP: group
- EHAO/EHAS: encryption, hash, authentication alg. offered/selected
- NIDP: no ID protection
- N: nonce

and

- K_{ir} = hash(N_i | N_r)
- shared secret key = f(N_i, N_r, g^{xy}, CKY_i, CKY_r)

Key Management Protocol for IPsec (Contd..)

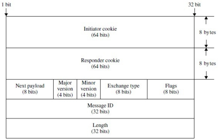
ISAKMP

- ISAKMP defines a framework for SA management and cryptographic key establishment for the Internet
- ISAKMP defines procedures and packet formats to establish, negotiate, modify and delete SAs.
- It also defines payloads for exchanging key generation and authentication data.

ISAKMP Payloads

- ISAKMP Payloads: ISAKMP Header, ISAKMP generic header
- Each A generic header which provides a payload chaining capability
- ISAKMP generic payload header 32 bits. Next Payload (8 bits), Reserved(8 bits), and Payload Length(16 bits).

Fig : ISAKMP Header



Payload Types for ISAKMP

Type	Description
SA	Used to negotiate security attribute
Proposal	Used in SA negotiation; for securing the communications channel
Transform	Consists of a specific security mechanism to be used to secure the communications channel.
Key Exchange	Oakley, Diffie-Hellman, the enhanced D-H key exchange, and the RSA-based key exchange (PGP)
Identification	Determine the identities of communication partners and may be used for determining authenticity of information
Certificate	Provides a mean to transport certificates via ISAKMP and can appear in any ISAKMP message
Certificate request	Provides a mean to request certificate via ISAKMP and can appear in any message.
Hash	Contains data generated by the hash function over some part of the message and/or ISAKMP state.
Signature	Verify the integrity of the data in the ISAKMP message
Nonce	Protect against replay attacks
Notification	Used to transmit information data
Delete	valid protocol-specific security association identifier that the sender has removed from its SA database

Key Management Protocol for IPsec (Contd..)

ISAKMP Exchanges

- ISAKMP allows the creation of exchanges for SA establishment and key exchange.
- Currently five default Exchange Types defined for ISAKMP.
- The primary difference between exchange types is the ordering of messages and the payload ordering within each message

Type	Description
Base Exchange	Designed to allow the Key Exchange and Authentication-related information to be transmitted together.
Identity Protection Exchange	Designed to separate the Key Exchange information from the Identity and Authentication-related informatio
Authentication Only Exchange	designed to allow only Authentication-related information to be transmitted.
Aggressive Exchange	designed to allow the Security Association, Key Exchange and Authentication-related payloads to be transmitted together
Informational Exchange	Designed as a one-way transmittal of information that can be used for security association management.

ISAKMP Payload Processing

Type	Description
Generalmsg	Minimize threats - denial of services and replay attacks
Header	Initiator construct, responder verify its validity.
GPH	Inititator creates, responder confirms its validity
SA	Initiator construct, responder, if not accepted, sends notification
Proposal	Initiator construct, and responder process it.
Transform	Initiator construct, responder process it.
Key exchange	Initiator construct, responder checks key exchange, fails then sends notification
Identification	initiator constructs, responder checks idetification, fails msg is discarded.
Certificate	Initiator constructs, responder if certificate data improper format, payload discarded
Certificate request	Initiator constructs, responder, if certificate authority is improperly formatted, the payload is discarded.
hash	Initiator constructs, responder checks hash fn, fails discard msg.
Signature	Initiator constructs, responder checks signature, fails discard msg.
Nonce	responder checks the exchange types
Notification	Initiator constructs, responder checks validity of msg and process it
Delete	Inititator construct, responder process it.

Transport Layer Security

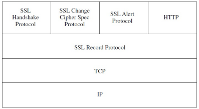
- SSL protocol
- Cryptographic Computations
- TLS Protocol

SSL Protocol

Introduction

- Secure Sockets Layer version 3 (SSLv3) was introduced by Netscape Communications Corporation in 1995.
- SSL is a two layered protocol.
- Eg. HTTP can operate on the top of the SSL layered protocol.
- The SSL protocol is composed of two layers: SSL Record Protocol, SSL Handshake Protocol.
- There are two defined specifications: SSL session and SSL connection

Fig : Two layered protocol



SSL Session

- An SSL session is an association between a client and a server.
- Sessions are created by the Handshake Protocol.
- SSL session coordinates the states of the client and server.
- When the handshake negotiation is completed, the client and server exchange change cipher spec messages, and they then communicate using the newly agreed-upon cipher spec.
- Defined by the following elements :Session identifier, peer certificate, compression method, cipher spec, master secrrt, is resumable

SSL Connection

- A connection is a transport (in the OSI layering model definition) that provides a suitable type of service.
- Every connection is associated with one session
- Defined by the following elements : Server and client random, Server write MAC secret, Client write MAC secret, Server write key, Client write key, Initialisation vectors, Sequence numbers

SSL Protocol - Record Protocol (Contd..)

Introduction

- SSL Record Protocol provides basic security services to various higher-layer protocols, the Handshake Protocol, the Change Cipher Spec Protocol and the Alert Protocol
- The SSL Record Protocol takes an application message to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies an MAC, encrypts, adds a header, and transmits the result in a TCP segment.
- The received data is decrypted, verified, decompressed, reassembled and then delivered to higher-level clients
- Overall operation of SSL protocols : Fragmentation, Compression and decompression, MAC, Append SSL record header.

Overall operation

- Fragmentation : A higher-layer message is fragmented into blocks.
- Compression and decompression : Compression is optionally applied in the SSL Record Protocol, but, if applied, it must be done before encryption and MAC computation.
- MAC : The MAC is computed before encryption. The compressed message plus the MAC are encrypted using symmetric encryption.
- Append SSL record header : The final processing of the SSL Record Protocol is to append an SSL record header.

Fig : SSL Record Protocol

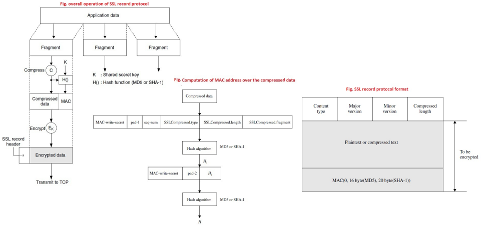


Fig. overall operation of SSL record protocol

Fig. Computation of MAC address over the compressed data

Fig. SSL record protocol format

SSL Protocol (Contd..)

Change Cipher Spec Protocol

- Change Cipher Spec Protocol consists of a single message, which is compressed and encrypted under the current CipherSpec.
- The message consists of a single byte of value 1.
- The client sends a change cipher spec message following handshake key exchange and certificate verify messages (if any), and the server sends one after successfully processing the key exchange message it received from the client.

Alert Protocol

- Alert messages convey the severity of the message and a description of the alert.
- Alert messages consist of 2 bytes.
- First byte takes the value warning or fatal to convey the seriousness of the message
- Second byte contains a code that indicates the specific alert.
- Alert messages are compressed and encrypted.
- Fatal related alerts : decompression failure, bad-certificate, etc.

SSL Protocol - SSL Handshake Protocol (Contd..)

Phase 1: Hello Messages for Logical Connection

- Hello request: sent by the server at any time, but may be ignored by the client if the Handshake Protocol is already underway.
- Client hello :A client sends a client hello message using the session ID of the session to be resumed.
- Server hello: response to a client hello message when it has found an acceptable set of algorithms

Phase 2: Server Authentication and Key Exchange

- Server certificate: If the server is to be authenticated, it must send a certificate immediately after server hello message
- Server key exchange msg: It is sent by the server, only when it is required.
- Certificate request message: server can optionally request a certificate from the client.
- Server hello done message: indicate the end of the server hello and associated msgs

Phase 3:Client Authentication and Key Exchange

- Client certificate message: This message is sent only when the server requests a certificate.
- Client key exchange message: With this message a premaster key is set.
- Certificate verify message: provide explicit verification of a client certificate

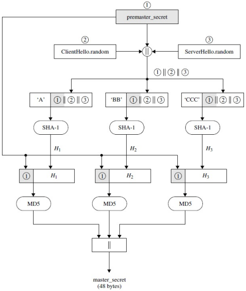
Phase 4:End of Secure Connection

- Change cipher spec messages: This message is immediately sent after the certificate verify message that is used to provide explicit verification of a client certificate.
- Finished message: to verify that the key exchange and authentication processes were successful.

Cryptographic Computations

Computing the Master Secret

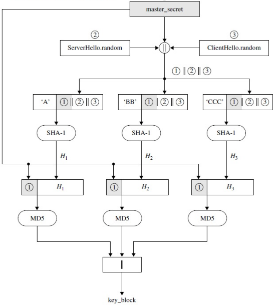
- To create the master secret, a premaster secret is first exchanged between two parties and then the master secret is calculated from it.
- The length of the premaster secret is not fixed and will vary depending on the key exchange method.
- Two ways for the exchange of the premaster secret: RSA, DiffieHellman.
- Generation of the master secret from the premaster secret is shown in Figure.



Converting the Master Secret into Cryptographic Parameters

- The generation of the key block from the master secret uses the same format for generation of the master secret from the premaster secret.
- Figure illustrates the steps for generation of the key block from the master secret.

Fig. Generation of Key block

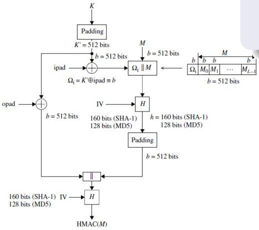


TLS Protocol

HMAC Algorithm

- HMAC is a secure digest of some data protected by a secret using hash function HMAC_SHA-1, HMAC_MD5
- Fig illustrates overall operation of HMACMD5 or HMACSHA-1.

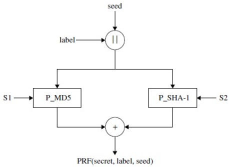
Fig : Overall Operation of HMAC-MD5 or HAMC-SHA-1



Pseudo-random Function

- TLS utilizes a pseudo-random function (PRF) to expand secrets into blocks of data for the purposes of key generation or validation
- Small values such as a secret, a seed and an identifying label as input and generates an output of arbitrary longer blocks of data.
- Data expansion function, P_hash(secret, seed)

Fig : A pseudo-random function (PRF) generation scheme



Error Alerts

- Error handling immediately close the connection. Eg. decrypt_error, protocol_version

Certificate Verify Message

- The MD5 and SHA-1 hashes are calculated only over handshake messages.

Finished Message

- A finished message is always sent immediately to verify that the key exchange and authentication processes were successful

Cryptographic Computations

- To establish connection, the TLS Record Protocol requires authentication, encryption and MAC algorithms, a master secret, and the client and server random values
- Compute the master secret and the key block.
- The length of the premaster secret will vary depending on key exchange method.