Demystifying - IPSec VPN's

By Abhishek Singh, CISSP# 82044

In this article I will cover the basics of IPSec and will try to provide a window into the mystical world of the IPSec VPNs. The intended audience is anyone who wants to have a quick go through of the IPSec VPNs. This article will suite to readers of range Beginners to Intermediate.

Gentle Intro to the IPSec

The need to converse privately, securely and cost-effectively gave rise to the technology of VPN's. IPSec is one such technology (a framework to be precise) allowing people to achieve the former. IPSec is a framework comprising of the following RFC's:

- IP Authentication Using Keyed MD5 (RFC 1828)
- The ESP DES-CBC Transform (RFC 1829)
- HMAC: Keyed-Hashing for Message Authentication (RFC 2104)
- HMAC-MD5 IP Authentication with Replay Prevention (RFC 2085)
- Security Architecture for the Internet Protocol (RFC 2401)
- The NULL Encryption Algorithm and Its Use with IPsec (RFC 2410)
- IP Security Document Roadmap (RFC 2411)
- IP Authentication Header (RFC 2402)
- The OAKLEY Key Determination Protocol (RFC 2412)
- The ESP CBC-Mode Cipher Algorithms (RFC 2451)
- The Use of HMAC-MD5-96 Within ESP and AH (RFC 2403)
- The Use of HMAC-SHA-1-96 Within ESP and AH (RFC 2404)
- The ESP DES-CBC Cipher Algorithm with Explicit IV (RFC 2405)
- IP Encapsulating Security Payload (ESP) (RFC 2406)
- The Internet IP Security Domain of Interpretation for ISAKMP (RFC 2407)
- Internet Security Association and Key Management Protocol (ISAKMP) (RFC 2408)
- The Internet Key Exchange (IKE) (RFC 2409)

IPsec is designed to provide interoperable, high quality, cryptographically-based security for IPv4 and IPv6. The set of security services offered includes access control, connectionless integrity, data origin authentication, protection against replays (a form of partial sequence integrity), confidentiality (encryption), and limited traffic flow confidentiality. These services are provided at the IP layer, offering protection for IP and/or upper layer protocols.

These objectives are met through the use of two traffic security protocols, the Authentication Header (AH) and the Encapsulating Security Payload (ESP), and through the use of cryptographic key management procedures and protocols. The set of IPsec protocols employed in any context, and the ways in which they are employed, will be determined by the security and system requirements of users, applications, and/or sites/organizations.

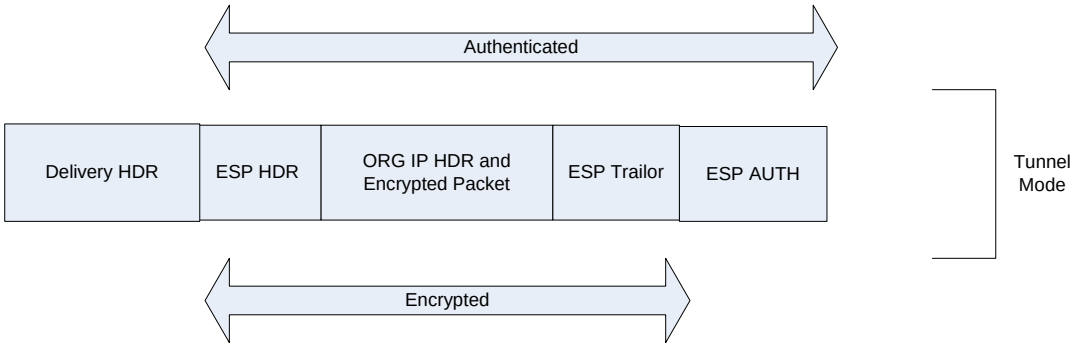
So, we can say that the major components of the IPSec are:

- **Security Protocols** Encapsulation Security Protocol (ESP) and Authentication Header (AH).
- **Key Management Protocols** ISAKMP, IKE, OAKLEY
- **Encryption Algorithms**

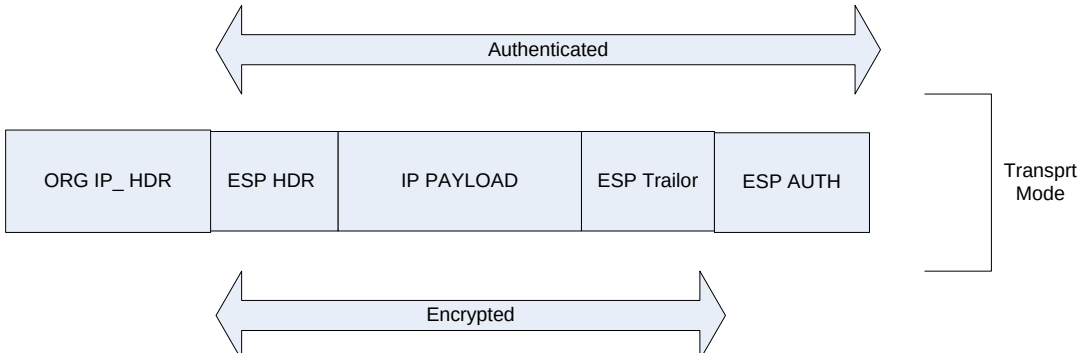
IPSec Modes

IPSec operates in two modes:

Tunnel mode: Compelete packet is ecrypted and a new header (Delivery Header) is attached to the encrypted packet.



Transport Mode: Encryption done only to the payload of the packet and the original header is retained.



IPSec Phases

IPSec Operates in two phases:

Phase I:

It lays the ground work for the actual transfer of data in the subsequent phase II and leads to a mutual agreement called Security Association. It contains ISAKMP (Internet Security Association Key Management Protocol) and Keymanagement protocols like IKE (Internet Key Exchange), OAKLEY, SKEME. Though these terms are used as synonyms but infact there is a difference between them ISAKMP is the protocol which defines:

- How the peer is to be authenticated.
- Key material Exchange.
- Doesn't define how authenticated key exchange is done which is actually the job of IKE.

Traditionally the IKE messages are carried in the Payload of the ISAKMP Packets.

The Internet Security Association and Key Management Protocol (ISAKMP) define procedures and packet formats to establish, negotiate, modify and delete Security Associations (SA). SAs contain all the information required for execution of various network security services, such as the IP layer services (such as header authentication and payload encapsulation), transport or application layer services, or self-protection of negotiation traffic. ISAKMP defines payloads for exchanging key generation and authentication data. These formats provide a consistent framework for transferring key and authentication data which is independent of the key generation technique, encryption algorithm and authentication mechanism.

ISAKMP is distinct from key exchange protocols in order to cleanly separate the details of security association management (and key management) from the details of key exchange. There may be many different key exchange protocols, each with different security properties. However, a common framework is required for agreeing to the format of SA attributes, and for negotiating, modifying, and deleting SAs. ISAKMP serves as this common framework.

ISAKMP is intended to support the negotiation of SAs for security protocols at all layers of the network stack (e.g., IPSEC, TLS, TLSP, OSPF, etc.). By centralizing the management of the security associations, ISAKMP reduces the amount of duplicated functionality within each security protocol. ISAKMP can also reduce connection setup time, by negotiating a whole stack of services at once.

ISAKMP is not bound to any specific cryptographic algorithm, key generation technique, or security mechanism. This flexibility provides independence from specific security mechanisms and algorithms provides a forward migration path to better mechanisms and algorithms. When improved security mechanisms are developed or new attacks against current encryption algorithms, authentication mechanisms and key exchanges are iscovered, ISAKMP will allow the updating of the algorithms and mechanisms without having to develop a completely new KMP or patch the current one.

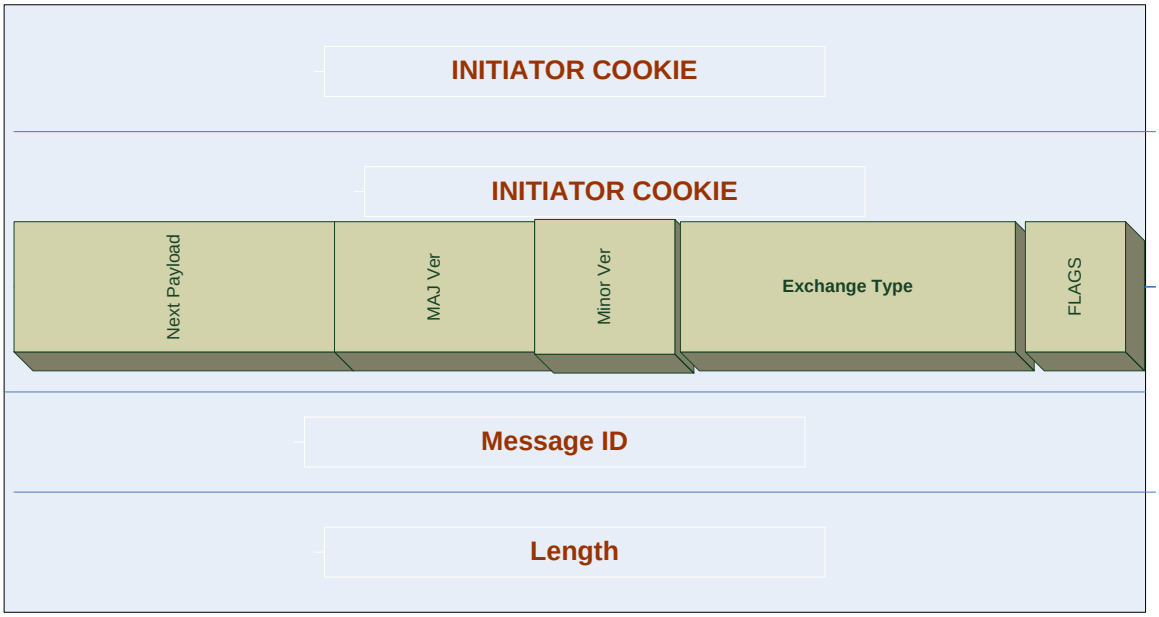


Fig. ISAKMP Header

A brief discussion of the fields:

Initiator Cookie (8 octets) - Cookie of entity that initiated SA establishment, SA notification, or SA deletion.

Responder Cookie (8 octets) - Cookie of entity that is responding to an SA establishment request, SA notification, or SA deletion.

Next Payload (1 octet) - Indicates the type of the first payload in the message. Common ones are:

- NONE
- Security Association (SA)
- Key Exchange (KE)
- Identification (ID)
- Certificate (CERT)
- Certificate Request (CR)
- Hash (HASH)
- Signature (SIG)
- Nonce (NONCE)
- Delete (D)
- Vendor ID (VID)

Major Version (4 bits) - indicates the major version of the ISAKMP protocol in use (it's fixed and is 1).

Exchange Type (1 octet) - indicates the type of exchange being used. This dictates the message and payload orderings in the ISAKMP exchanges. Some common types:

- NONE
- Base
- Identity Protection
- Authentication Only

Flags (1 octet) - indicates specific options that are set for the ISAKMP exchange. There are 3 flags:

Bit 0 (Encryption Bit): if set, signifies that all the subsequent payloads are encrypted by the algorthim chosen in SA.

Bit 1 (Commit Bit): Used for Key exachange synchronization to signal that the encrypted material is not received before the completion of Key Establishement.

Bit 2 (Authentication Only Bit): This bit is intended for use with the Informational Exchange with a Notify payload and will allow the transmission of information with integrity checking, but no encryption.

So far for ISAKMP, let's move on to the beast of transfer Internet Key exchange, IKE; IKE is the actual work horse traditionally used to establish the ground work for the Phase II by setting up the *Security Associations (SA)* thereby defining the security policies – keys, encryption, authentication that are to be applied to the data (not the user but to the control data) IKE results in establishment of a control connection between two peers which will lay the framework of actual data transfer.

The following attributes are used by IKE and are negotiated as part of the ISAKMP Security Association. (These attributes pertain only to the ISAKMP Security Association and not to any Security Associations that ISAKMP may be negotiating on behalf of other services.)

Following are the attributes which will be negotiated during the IKE

- Encryption algorithm
- Hash algorithm
- Authentication method
- Information about a group over which to do Diffie-Hellman.

Deffie-Hellman Algorithm:

Postulated in 1976 by two mathematicians Bailey W. Diffie from Berkeley and Martin E. Hellman, defined the Diffie-Hellman Agreement Protocol (also known as exponential key agreement) and published it in a paper entitled, "New Directions in Cryptography." The provided a method to achieve a secret key between two peers without actually transmitting the key and using the relationship between prime numbers. Details of the Algorithm are beyond the scope of this article.

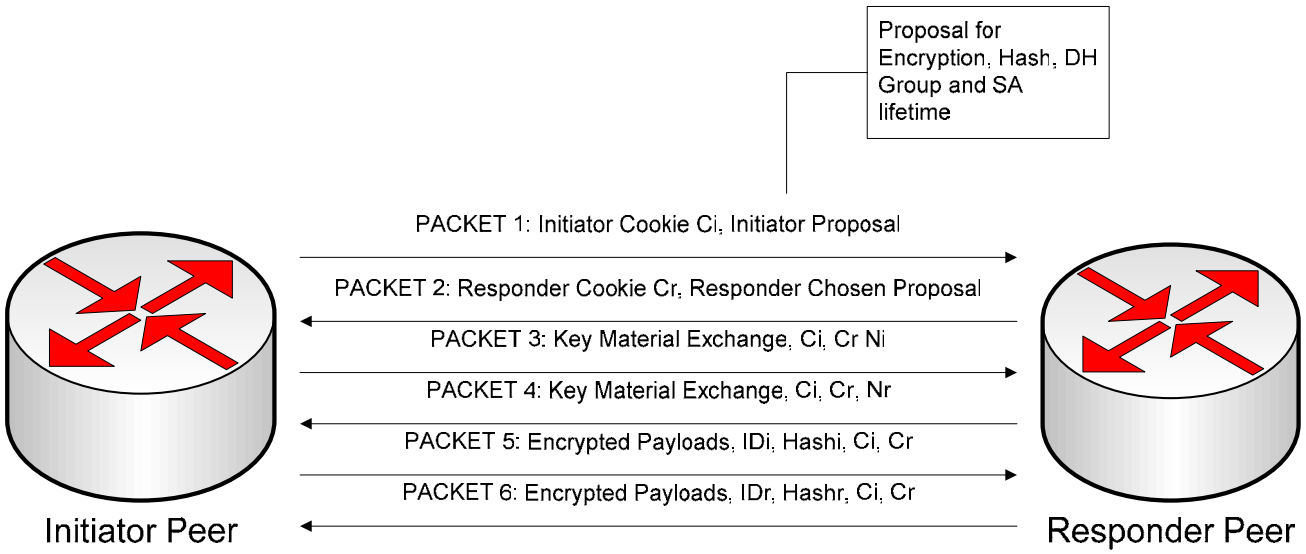


Fig Phase I of IPSec

IKE implementations must support

- DES in CBC mode (one of the 4 modes of DES) for encryption.
- MD5 and SHA.

Might Support

- 3DES for encryption.
- Tiger ([TIGER]) for hash.

There are two basic methods used to establish an authenticated key exchange: **Main Mode** and **Aggressive Mode**. Each generates authenticated keying material from an ephemeral Diffie-Hellman exchange. Main Mode is mandatory for any application; Aggressive Mode may also be implemented. In addition, Quick Mode MUST be implemented as a mechanism to generate fresh keying material and negotiate non-ISAKMP security services.

Main Mode

Main Mode is an instantiation of the ISAKMP Identity Protect Exchange: The first two messages negotiate policy; the next two exchange Diffie-Hellman public values and ancillary data (e.g. nonces) necessary for the exchange; and the last two messages authenticate the Diffie-Hellman Exchange. The authentication method negotiated as part of the initial ISAKMP exchange influences the composition of the payloads but not their purpose.

Packet 1: The 1st payload of the 1st packet has to be the Security Association which inturn contains Proposal Payload containg further payloads Transform (Transform payload is sent for each of the encryption, hash algorithms configured on the initiator) apart from the 1st payload being SA payload there can be any order of the payloads:

- Encryption Algorithm: DES-CBC, AES-CBC, 3DES-CBC
- Hash Algorithm: MD5, SHA-1
- Authentication Method: pre-shared key or Cetificate
- Group Description: alternate 1024-bit MODP group
- Life Type: Seconds or KB of data exchanged
- Life Duration: contains data based on the value in previous step e.g 86400 s

Packet 2 will be an answer to the proposals sent to the responder, with a chosen proposal or sends no proposal chosen. So the contents of the Security Association payload of second packet would be:

Encryption Algorithm:	3DES-CBC
Hash Algorithm:	SHA1
Authentication Method:	pre-shared key
Group Description:	alternate 1024-bit MODP group
Life Type:	Seconds
Life Duration:	86400

Contents of the Packet 3 would have the key data and the nonce. It would encompass two payloads:

1. KEY EXCHANGE: This payload contains the following (i.e. typically the most evident items)

   Length: 00 84 (132)

Key Data: (e.g. see below)

7e ab 64 b5 7d 31 8d 97 43 d1 e5 23 e8 44 be 53 a1 8d 92 55 d9 d4 0a 34 49 e1 18 e9 16 29 a8 db ee 3f df 89 2d 79 30 4f eb 6a 6a ae 57 89 71 e1 ad 1f c4 f5 ba 4c 65 c6 e5 c6 ba d2 56 80 28 6a f3 57 e6 9d c0 da 1c 58 31 8b bc eb de 45 d8 50 0f cb 8f 29 55 2d 81 56 69 67 fe 3d e0 2f 1e cf c8 28 fe f9 94 aa 6e e6 a8 d1 e5 0a 0f 0f db bf 28 88 54 10 7f e7 16 5c d2 cb 25 1d 59 45 ce b9

2. NONCE:

Length: 00 18 (24)

Nonce Data: (e.g. see below)

b9 98 ad a7 ac 36 be 43 c6 65 8e 51 51 4a fa a7 6c 3b 3a ba

Contents of Packet 4 would be the response of the responding peer and would contain the fields same as in packet 3 but the values are of course of the responder for both Key Data and the Nonce.

Note: Now that the Key data is exchanged and the algorithms are agreed upon further communication would be encrypted. Since the keys can now be formed using the DH.

Packet 5 would have the following fields typically:

Header would have the following:

InitCookie: 1e f4 50 03 5f 56 6d 98
RespCookie: ea bb 50 b7 fc 4f d9 3e
Next Payload: Identification
MjVer: 1
MnVer: 0
ExcType: 02 (2)
EncFlag: 1 # Notice that the ENC bit is set meaning encryption ahead
CommitFlag: 0
ResFlag: 0
MsgID: 00 00 00 00
Length: 00 00 00 44 (68)

Payload would have Identification information of the Initiator peer:

Next Payload: Hash
Reserved: 0
Length: 00 0c (12)

ID type: ID_IPV4_ADDR
Reserved2: 00 00 00 (0)
ID Data: 42 e3 8d b3 (1x.1x.1x.1x)

ID is identified by the means of the IP address of the initiator which the responder wills verfify by decrypting the payload.

HASH Payload contains the hash of the above:

Hash Data: (e.g. below)

39 64 b8 c2 8b 3c 38 71 e0 15 ef 35 bd 80 53 f9 17 6c 04 92

Packet 6 similar fields as of 5 but contains values of responding peer.

It should be noted that initator can send any number of the potential *Security Associations* to responder but the responder must limit itself so that it will inspect only certain number of proposals.

Four different authentication methods are allowed with either Main Mode or Aggressive Mode:

- Digital Signature,
- Two forms of authentication with public key encryption, or pre-shared key.

At the end of the Phase 1 we will have an IKE Security Association setup between the two interested peers. These SA are duplex in nature i.e. same SA will work in both inbound as well as outbound direction.

It should be noted that since the ID payload of the 5th and 6th packets are encrypted so while using the pre-shared key method of authentication we have to do it with IP Address only, also if the configuration involves a client to site scenario gateway has no means of knowing the IP Address of the client before hand so we have Aggressive mode (as discussed below) as a suitable option since the IDs are passed in clear. Also if the certificates based authentication is used the request for the certificates has to be done in 3rd and 4th messages while the actual exchange of the certificates is done in 5th and 6th messages.

Aggressive mode

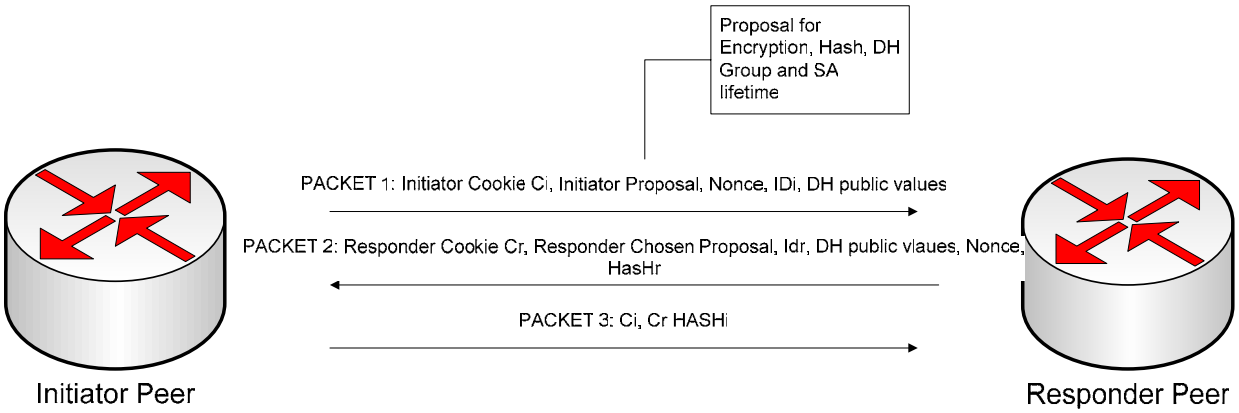


Fig. Aggressive mode

Aggressive mode the process:

- The first two messages negotiate policy, exchange Diffie-Hellman public values and ancillary data necessary for the exchange, and identities.
- In addition the second message authenticates the responder.
- The third message authenticates the initiator and provides a proof of participation in the exchange.

It should be noted that the ID are sent in clear during the first 2 packet exchanges and the DH group values cant be negotiated also this presents a limitation on the type of the authentication methods used e.g. Public Key Encryption might not be negotiated. So if we require the rich feature set of IKE **Main Mode** is advisable.

- Security Association: contains the agreed protocols, keys, lifetime, authentication methods and other data negotiated above, also since IKE can't use ESP and AH to maintain the SA it uses special attributes called Cookies (identifying the specific SA). Thereby providing a method so that Phase 2 can be done securely.

Phase 2 Quick Mode - IPSec SA establishment

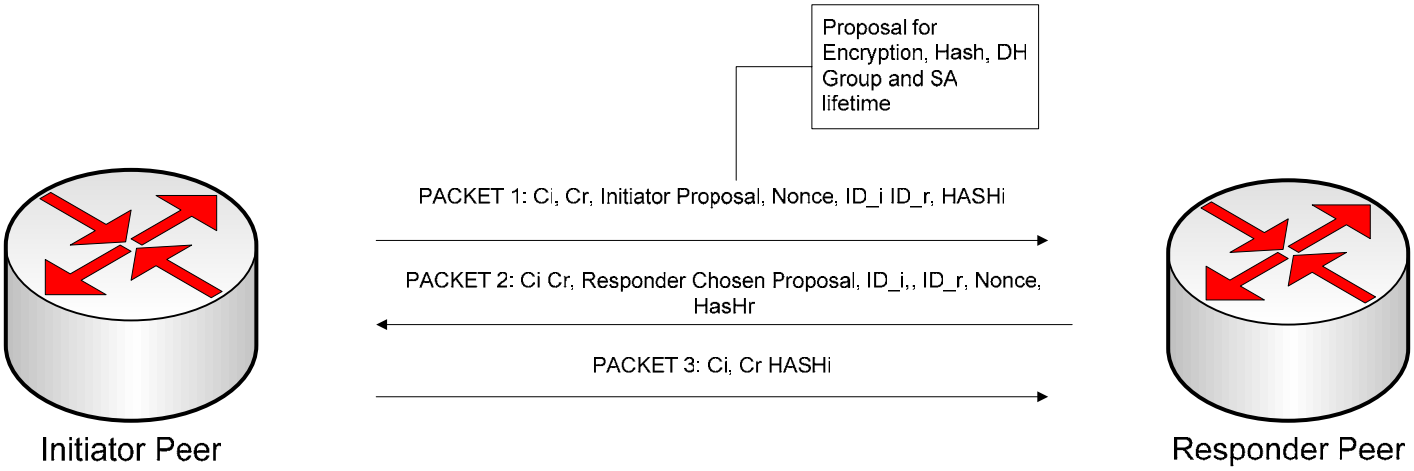


Fig Quick Mode Negotiation

Qucik mode is not the complete exchange but is based to a larger extent on the Phase 1 and is used to derive shared policy non-ISAKMP SA. Also there is a message id field in the header which attaches the quick mode negotiation to a particular SA which inturn is recognized by means of Cookies. Since each instance of the Quick Mode uses a unique initialization vector there can be many Quck Modes for a particular SA and each, thereby, being uniquely identified by a unique messge ID.

In terms of the structure of the packets Quick Mode is far much stringent in it:

- All fields except the ISAKMP header are encrypted.
- HASH payload must immediately follow Header.
- Security Association Payload (Containing Prposals) must immediately follow HASH.

HASH provides the authentication to the message and acts a proof of liveliness of the peer.

Anatomy of the Quick Mode packet by packet

Packet 1:

Contents:

Header: Major constituents

InitCookie: e.g. 26 c4 5e d0 e5 82 d9 2f
RespCookie: e.g. c4 b6 8c bd 37 62 fb 80
Next Payload: Hash
MjVer: 1
MnVer: 0
ExcType: 20 (32)
EncFlag: 1
CommitFlag: 0
ResFlag: 0
MsgID: e.g. 76 80 cf 8b
Length: e.g. 00 00 02 6c (620)

Cookies are same as the ones setup in the Phase 1
Also notice that the Enc Flag is set

Payloads:

Hash Payload

Security Association Payload containing proposals and

PropNum: 1
ProtId: e.g 3
SPI Size: e.g. 4
SPI: e.g d6 50 11 d0

Nonce

ID: ID of intiator IP_Addr in case of pre-shared authentication

ID: ID of intiator IP_Addr in case of pre-shared authentication

Packet 2:

Same as packet 1 with the proposal chosen by the responder.

Packet 3:

Used for authentication by responder to validate the channel before actual transmission of the data begins.

Header

InitCookie:	e.g. 26 c4 5e d0 e5 82 d9 2f
RespCookie:	e.g. c4 b6 8c bd 37 62 fb 80
Next Payload:	Hash
MjVer:	1
MnVer:	0
ExcType: 20 (32)	
EncFlag: 1	
CommitFlag:	0
ResFlag: 0	
MsgID:	76 80 cf 8b
Length:	00 00 00 3c (60)

Payload

HASH

It is interesting to note that after the packet 2 exchange initiator has enough material to start the encryption but if responder is still in process of the authentication/ validation it will drop the data hence there is commit bit set in the 2nd packet so that initiator has to wait for a go ahead and encrypt signal from the responder before it actually starts encrypting the traffic.

Note that the outcome of this phase is the association is establishment of a pair of SA at each peer, each one calculating its inbound SA. One SA from pair will be inbound and other outbound, as in to the ISAKMP SA's.

Nonces in Quick mode provide a mechanism of protection against the replay attack and for the generation of fresh key material. An additional payload can be defined to exchange the DH values but its not mandatory as per the standards. Basic Quick Mode is refreshes key material from knowledge derived from Phase 1, while the use of this optional payload will provide what is referred to as *Perfect Forward Secrecy (PFS)*.

The identities of the SAs negotiated in Quick Mode are implicitly assumed to be the IP addresses of the ISAKMP peers, without any implied constraints on the protocol or port numbers allowed, unless client identifiers are specified in Quick Mode. If ISAKMP is acting as a client negotiator on behalf of another party, the identities of the parties MUST be passed as IDi and then IDr. Local policy will dictate whether the proposals are acceptable for the identities specified. If the client identities are not acceptable to the Quick Mode responder (due to policy or other reasons), a Notify payload with Notify Message Type INVALID-ID-INFORMATION is sent.

The client identities are used to identify and direct traffic to the appropriate tunnel in cases where multiple tunnels exist between two peers and also to allow for unique and shared SAs with different granularities.

Also in Phase 2 each tunnel has a different SPI (and is recognized by the same) as per the exchanges done in *Packet 1* and *Packet 2* of **Quick Mode**.

In brief each of the IPSec SA is identified by the following:

1. SPI (Security Parameter Index), contained in the ESP Header (see below for a brief discussion of ESP).
2. Destination IP Address.
3. Security Protocol Identifer (e.g AH-MD5, ESP-DES etc.).

An important point to mention here is that only one protocol can be defined per SA so 2 SA are required if both Authentication and Confidentiality are needed. Each outgoing packet, if matched by the Security Policy, has a SA associated with it and an incoming packet is assigned and SA based on the SPI value contained in the Header.

Security Policy Database (SPD)

It's a database associated with SA and describes:

- What is to be encrypted?
- How it's to be encrypted?
- With whom the security related negotiations are to be made i.e. with whom the traffic is to be encrypted?

We can say that a Security Association is based on what is defined in the SPD. The process of encrypting the traffic is – when a packet arrives at the gateway its matched with the policy database which defines the action which is to be taken on the packet, Bypass, Discard, or Subject to IPSec Processing (drop/encrypt/ send in clear). Once it's decided that the packet is to be encrypted an SA is identified for the connection (if an SA doesn't exists a new one is created) and packet is now encrypted and forwarded to appropriate Gateway. For an Inbound packet SPI are matched as contained in the Header of the packet. It's important to remember here

that if no SA could be identified for an incoming packet it's ***dropped.*** Packets are matched to the SPD based on the selectors (one or more), identifying the IPSec policies with a communication, these selectors are

1. **Destination IP Address**
2. **Source IP Address**
3. **Data Sensitivity level**
4. **Upper layer protocols**
5. **Upper layer ports**

First 2 selectors Dest and Src IP address are the ones used by most of the vendors.

Security Association Database

Contains the footprints of an SA – SPI (acts an index to SADB), Destination IP Address and IPSec Protocol ESP/ AH, these are for Inbound packets. For outbound packet SADB is indexed by an appropriate pointer in SPD. Each new SA updates the SADB with SA params.

Below are the attributes contained for IPSec Processing:

1. **Sequence number** - Provided by ESP and AH Header
2. **Sequence number overflow** - A window which is constantly updated, used only for the outgoing packets
3. **32 Bit antireplay window –** Packets normally arrive at the destination peer out-of-order so this window is constantly updated with the arriving packets, thereby providing the time in which subsequent packets must arrive.
4. **Lifetime of SA –** Two type of the lifetimes are used

    - **Hard-Life Time-** Time at which the old SA is shunned and new ones (negotiated after Soft Life time) are used.
    - **Soft-Life Time –** Time, before the actual expiry time of the SA, when the negotiation for the new SA actually starts so that the traffic is not disturbed when the life time of SA is over.

5. **AH algorithm.**
6. **ESP Authentication algorithm**
7. **ESP Encryption Algorithm**
8. **IPSec mode – Tunnel or Transport**
9. **PMTU**

Encapsulating Security Payload - ESP (IP Protocol 50)

Services provided by ESP are:

- Data Confidentiality.
- Integrity
- Data Origin Authentication
- Limited Traffic flow confidentiality

The services provided by ESP are decided during establishment of *Security Association* (*SA*). Confidentialy is provided by means of Encryption and Data Integrity by Authenticator. It is highly recommended that if the Integrity is the sevice which you require use AH instead of ESP.

ESP should be considered as a standard for the encryption and defines the procedures necessary for common encryption process, but it doesnot define, or limit itself on, what can be used to encrypt the traffic.

Authentication and Confidentiality are the two important services provided by the ESP its interesting to note that according to the standard botn of them are optional, but one of them has to be mandatorily applied.

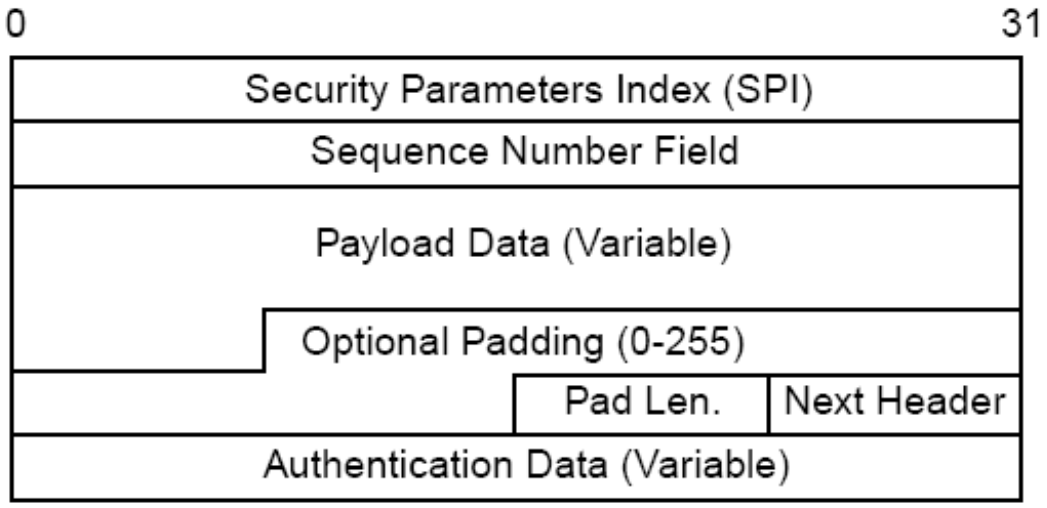


Fig. ESP Header

Discussion of the important Fields:

1. **Security Parameter Index (32 bit) -** Values 1-255 are reserved and 0 is only used during the ISAKMP SA negotiations.

2. **Sequence Number (32 bit) –** Provides protection against the replay, it is unique for the Lifetime of SA. When this is exhausted new SA is negotiated. This replay protection can be done if data origin authentication is selected but is completely based on the discretion of the Reciever.

3. **Payload Data (Variable Length) –** If the algorithm uses the Initialization Vector (IV) its contained at the beginning of the Payload, this shows that IPSec is completely independent of the encryption algorithm negotiated by the peers since the recieveing Peer will extact this IV and apply directly to the Algorithm.

4. **Optinal Padding –** This provides communication flow confidentiality as well as ensures that the Pad Length and Next Header field align to 4 Byte boundaries.

5. **Next Header –** Conatins the header following the ESP header.

Authenticaton Header - AH (IP Protocol 51)

Services provided by AH are:

- Data Integrity.
- Data Origin Authentication.
- Optional Replay Protection.

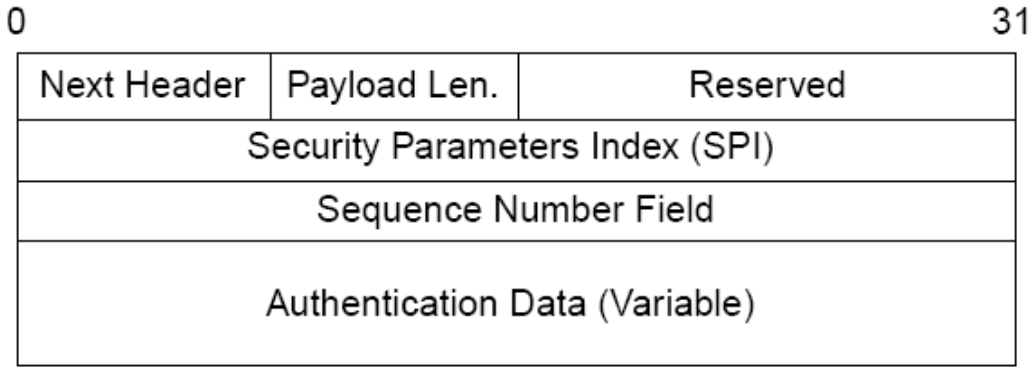


Fig. AH Header

Most of the feld have same implications as in ESP Header. Others are as discussed below:

1. **Reserved Field –** Always set to 0.

2. **Authentication Data –** Contains the message digest or authenticator used to verify the message.

What have I not Covered in this paper

Since this is very broad and difficult topic to fit into length of a single readable article, some important but advanced topic which I have not been able to accommodate in this paper are the implications of the NAT on IPSec, Common

VPN Topologies – Mesh and Star, , GRE and IPSec and Possible pros and cons of IPSec over other VPN solutions.

References

1. Works and Research of my friend Raghu Chinthoju Network Specialist – Google India.
2. IPSec VPN Design By Vijay Bollapragada, Mohamed Khalid, Scott Wainner CISCO Press ISBN 1-58705-111-7.
3. A Technical Guide to IPSec VPNs by James S. Tiller Auerbach Press ISBN 0-8493-876-3.
4. IP Authentication Using Keyed MD5 (RFC 1828).
5. The ESP DES-CBC Transform (RFC 1829).
6. HMAC: Keyed-Hashing for Message Authentication (RFC 2104).
7. HMAC-MD5 IP Authentication with Replay Prevention (RFC 2085).
8. Security Architecture for the Internet Protocol (RFC 2401).
9. The NULL Encryption Algorithm and Its Use with IPsec (RFC 2410).
10. IP Security Document Roadmap (RFC 2411).
11. IP Authentication Header (RFC 2402).
12. The OAKLEY Key Determination Protocol (RFC 2412).
13. The ESP CBC-Mode Cipher Algorithms (RFC 2451).
14. The Use of HMAC-MD5-96 Within ESP and AH (RFC 2403).
15. The Use of HMAC-SHA-1-96 Within ESP and AH (RFC 2404).
16. The ESP DES-CBC Cipher Algorithm with Explicit IV (RFC 2405).
17. IP Encapsulating Security Payload (ESP) (RFC 2406).
18. The Internet IP Security Domain of Interpretation for ISAKMP (RFC 2407).
19. Internet Security Association and Key Management Protocol (ISAKMP) (RFC 2408).
20. The Internet Key Exchange (IKE) (RFC 2409).