## IPSec: Providing Security at the Network Layer

- A more broad-based approach to security consists of providing authentication, confidentiality, and key management at the level of IP packets (the network layer).

  [The different layers of the TCP/IP protocol stack are:
  - Application Layer (HTTP, FTP, SMTP, SSH, POP3, TLS/SSL, DNS, etc.)
  - Transport Layer (TCP, UDP, etc.)
  - Network Layer (IP (IPv4, IPv6), ICMP, IGMP, etc.)
  - Link Layer (Ethernet, Wi-Fi, PPP, SLIP, etc.)

  Note that TCP and IP are two separate protocols in the stack. Nonetheless, the entire stack is commonly and loosely referred to as the TCP/IP protocol. TCP and IP were the first ones to be developed. Note that this representation of the internet protocol suite is somewhat lacking, especially with regard to the Application layer. As a case in point, it is not reasonable to think of both HTTP and TLS/SSL at the same level because the former call on the latter for security. In that sense, HTTP is above TLS/SSL. A superior layering of the protocols is provided by the OSI (Open Systems Interconnection) model that divides the application layer as shown above into three finer-grained layers: **Application**, **Presentation**, and **Session**. In this model TLS/SSL would belong to the Session layer, whereas HTTP would stay in the Application layer. The Presentation layer includes protocols such as SMB (Samba). ]

- When security is implemented at the network layer in the TCP/IP protocol, it covers all applications running over the network. That makes it unnecessary to provide security separately for, say, email exchange, running distributed databases, file transfer, remote site

administration, etc. **Thus the application-level programs are spared the computational overhead of having to provide for security.**

- IP-level authentication means that source of the packet is as stated in the packet header. Additionally, it means that the packet was not altered during transmission.

- IP-level confidentiality means that third-party packet sniffers cannot eavesdrop on the communications.

- **IPSec** is a specification for the IP-level security features that are built into the IPv6 internet protocol. These security features can also be used with the IPv4 internet protocol.

  [In addition to the built-in security, the main features of IPv6 is its much larger address space. The older and much more widely used IPv4 supports $4.3 \times 10^9$ addresses, IPv6 supports $3.4 \times 10^{38}$ addresses. (The population of the earth is only (roughly) $6 \times 10^9$.) It is interesting to note that because of features like DHCP that allows for IP addresses to be dynamically allocated to devices and NAT that allows for network address translation on the fly, the address limitations of IPv4 suddenly do not appear to be as serious as many people thought about five years back. Even though IPv6 has now been around for roughly ten years, it accounts for only a tiny fraction of the live addresses in the internet. DHCP stands for the Dynamic Host Configuration Protocol. NAT allows the devices connected on a private network (such the devices connected to your wireless router at home) to access the internet using a single public IP address. NAT is achieved

by the router rewriting the source and/or destination address in the IP packets as they pass through. Also note that the U.S. Government has specified that all federal agencies must deploy IPv6 by 2008.]

- IPSec can be used either in the **tunnel mode**, in which the security of packet traffic in a local area network (as formed by the machines connected to a single router) can be provided by a single node; or the **transport mode** in which the end points of a communication link do all the packet-level security processing.

- The main thing about the tunnel mode that the IP datagram can encapsulate an inner IP datagram. The real source and the destination addresses only appear in the inner datagram. The source and the destination addresses in the outer datagram are used by the security gateways. This notion of an encapsulated IP datagram is shown in the figure on Slide 27.

- Note that IPSec-based security services can be provided between a pair of communication hosts, between a pair of communicating security gateways, or between a security gateway and a host.

- IPSec includes **filtering capability** so that only specified traffic need be subject to security processing. In other words, only those packets that are deemed to be security-sensitive need to be further processed for authentication, confidentiality, etc.

## IPv4 and IPv6 Packet Headers

- IPSec security features are implemented as extension headers that follow the main IP header in an IP packet.

- To see where the extension headers go, let's first look at the main packet headers for IPv4 and IPv6. These are shown in the figure on the Slide 23.

- The total length of an IPv4 header is fixed and equals 20 bytes. The **Total Length**, a 16-bit word, designates the total length of the packet in octets. (Therefore, the maximum size of an IPv4 packet is 65,536 bytes.) The **Identification** field holds a unique value for a given pair of source and destination addresses. The **Time to Live** field, specified by 8 bits, is subtracted by 1 for each pass through a router. The **Source Address** and the **Destination Address** are each represented by 32 bits.

- The **Protocol** field of the IPv4 header is important to us. Ordinarily it indicates the next higher level protocol that is to receive the data field at the destination. Each protocol (such as the TCP protocol) has a number assigned to it. It is this number that is stored in the **Protocol** field. [When IPSec is used with IPv4, this field contains the integer value that represents the security header to follow the main header. For example, the integer 50 represents the ESP protocol. If the next header is the ESP

header, number 50 will be stored in the **Protocol** field. The number 51 represents the AH Protocol. We will shortly talk about AH and ESP. By the way, the number 6 represents the TCP header. ]

- An IPv6 header has a fixed length of 40 bytes. What makes IPv6 headers more versatile is that the header of a packet of packet can be followed by **extension headers**. The main header and any extension headers are linked by the **Next Header** field consisting of 8 bits. The extension headers of interest to us are the **Authentication Header** and the **Encapsulating Security Payload Header**. The **Source Address** and the **Destination Address** are each represented by 128 bits.

- The IPv6 packets in the tunnel mode may have an **outer IP header** enclosing an **inner IP header**. The inner IP header carries the ultimate source and destination addresses and the outer IP header the addresses of security gateways. The authentication and encryption for confidentiality may include all of the fields of the inner packet header.

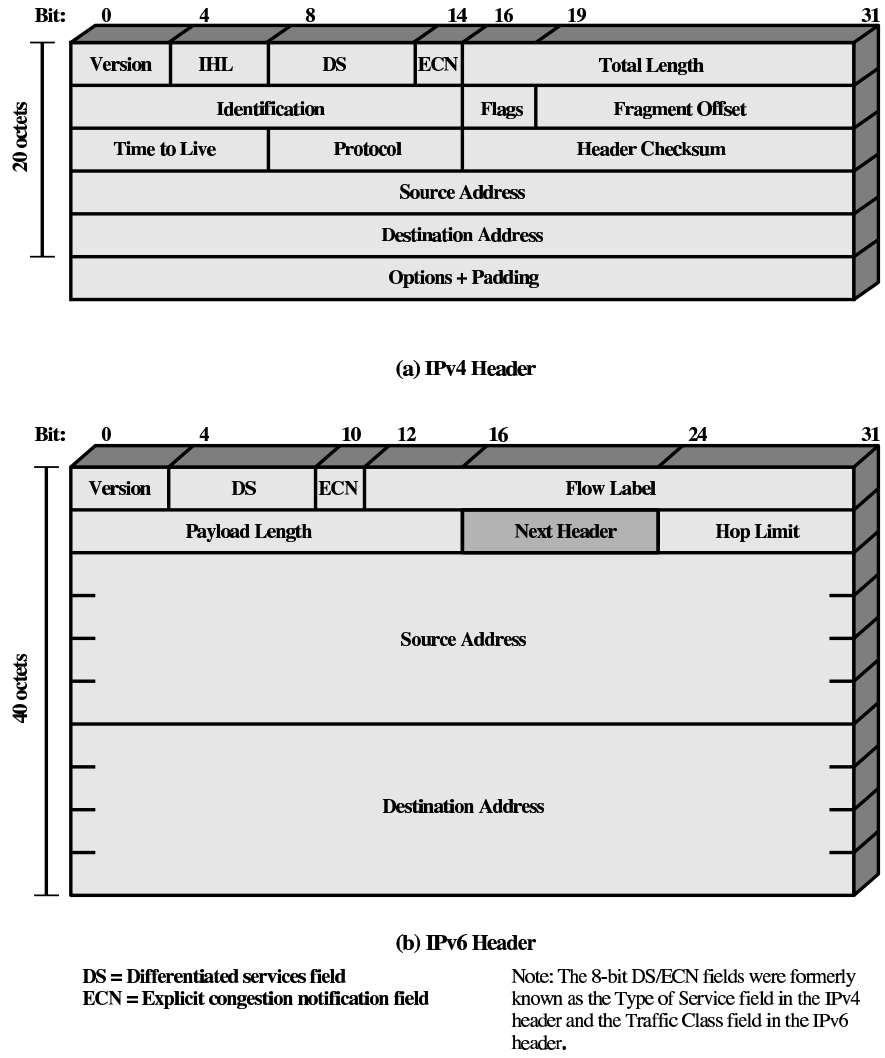



**Figure 16.14 IP Headers**

This figure is from Chapter 16 of Stallings: "Cryptography and Network Security", Fourth Edition

## IPSec: Authentication Header

- The figure on the Slide 26 shows the **Authentication Header** (AH).

- In IPv4 and in the transport mode of IPv6, the AH header is inserted after the IP header and before an upper layer protocol (such as TCP, UDP, etc.) header and before any *other* IPSec headers. For example, when no AH header is used, an IPv4 packet may look like

```
original IP  |   TCP header   |  Data
  header     |                |
```

  When the AH header is included, an IPv4 packet looks like

```
original IP  |  AH  |  TCP header   |  Data
  header     |      |               |
```

  IPv6, since it allows for various sorts of extension headers, with the AH, a packet may look like what is shown below in transport mode

```
original IP    |   extension hdrs   |  TCP header   |  Data
  header       |     if present     |               |
```

  When the AH header is included, an IPv6 packet may looks like

```
original IP    |   other extension  |  AH  |  TCP header   |  Data
  header       |       headers      |      |               |
```

- The **Payload Length** field specifies the length of the AH in 32-bit word, minus the integer 2.

- The **Security Parameter Index** field, a 32-bit value, establishes the **Security Association** for this packet.

- The **Authentication Data Field** holds the first 96 bits of the MAC (Message Authentication Code) of the packet calculated with either the SHA-1 hash function or the HMAC algorithm.

- The MAC is calculated over the IP header fields that do not change in transit, the AH header (but without the Authentication Data since it will be the output of the MAC algorithm), and the **inner IP packet**.

- The receiver calculates the MAC value over the appropriate fields of the packet and compares it with the value stored in the **Authentication Data** field. If the two values do not match, the packet is discarded.

- The Authentication Header only encrypts the **Integrity Check Value**.

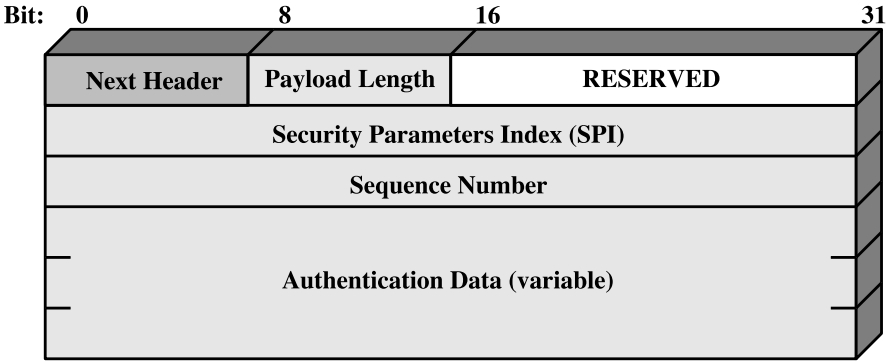
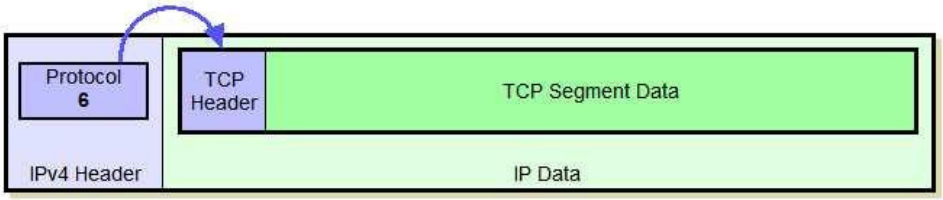
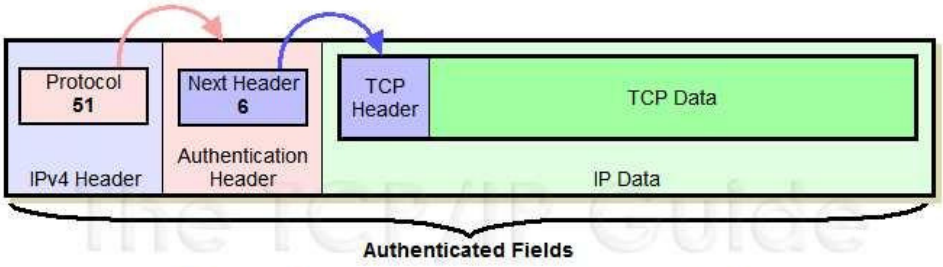



**Figure 16.3 IPSec Authentication Header**

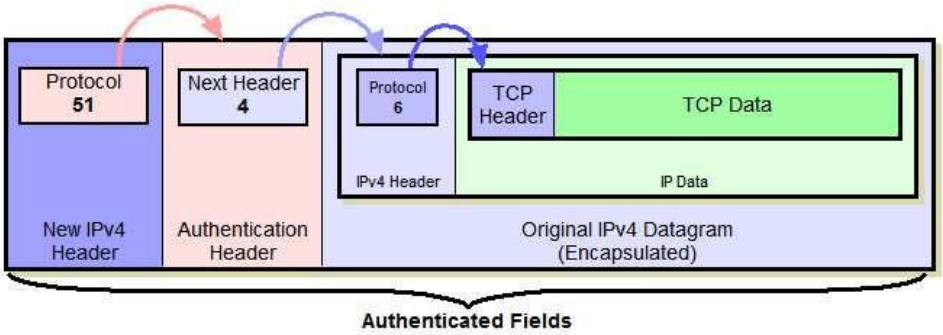
This figure is from Chapter 16 of Stallings: "Cryptography and Network Security", Fourth Edition




**Original IPv4 Datagram Format**

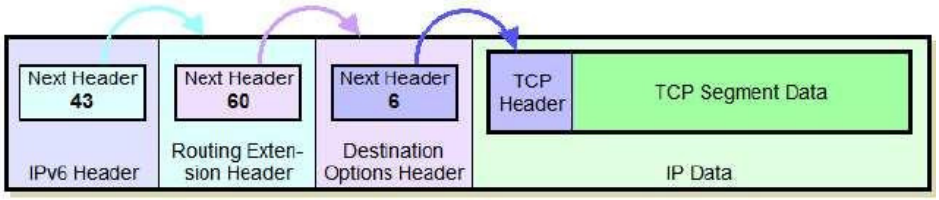



**IPv4 AH Datagram Format - IPSec Transport Mode**

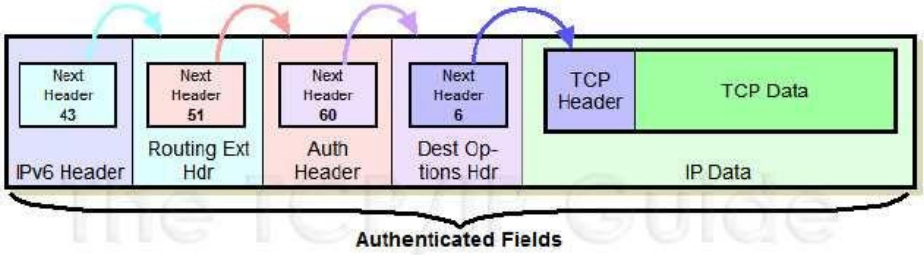



**IPv4 AH Datagram Format - IPSec Tunnel Mode**

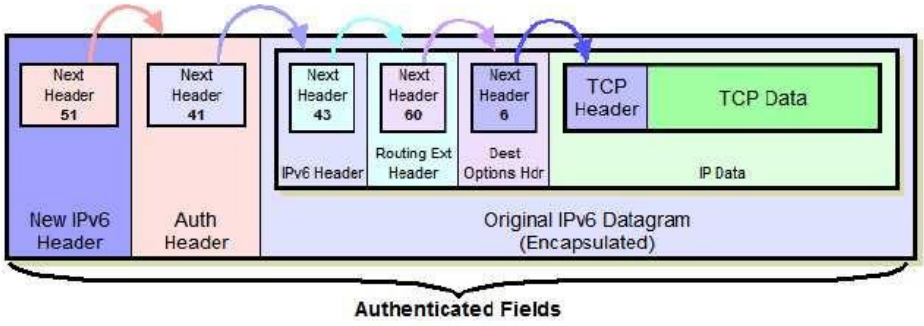
This figure is from http://www.tcpguide.com




**Original IPv6 Datagram Format (Including Routing Extension Header and Destination-Specific Destination Options Extension Header)**




**IPv6 AH Datagram Format - IPSec Transport Mode**




**IPv6 AH Datagram Format - IPSec Tunnel Mode**

This figure is from http://www.tcpguide.com

## IPSec: Encapsulating Security Payload (ESP) Header

- The figure on Slide 32 shows the **Encapsulating Security Payload** (ESP) header.

- ESP may be used to provide the same services as the AH header, **but ESP can also provide confidentiality**. ESP may be applied alone or in conjunction with the AH header. More generally, ESP can be used to provide confidentiality, data origin authentication, limited traffic flow confidential, etc., depending on the options selected through the value stored in the **Security Parameter Index** field. This value, that must be between 1 and 255,

- The **Payload Data** field, of variable length, can store the initialization vector required by the encryption algorithm.

- ESP achieves confidentiality by replacing the original contents of the data field of the enclosing IP packet with an encrypted version. Depending on the security requirements, the encryption may be applied either to the entire datagram or to, say, just its transport-layer segment (TCP).

- Since the ciphertext produced may not terminate on a 4-byte

boundary even when the plaintext does, the **Padding** field meant to ensure that this boundary constraint applies to the overall packet.

- The **Authentication Data** field includes the MAC value of the ESP packet. In the context of IPSec, this value is known as the **Integrity Check Value**.

- ESP's authentication scheme can be used either independently of the AH header or in conjunction with it.

- Before encryption, an **ESP Trailer** is appended to the data to be encrypted. The payload (meaning the TCP/UDP message in the transport mode or the encapsulated IP datagram in the tunnel mode) and the ESP Trailer are both encrypted, but the ESP Header is not.

- If the optional ESP authentication is used, the authenticator is calculated over the entire ESP datagram. This includes the ESP Header, the payload, and the trailer.

- As shown in the figure on Slides 32 through 34, the ESP Header comes just before the protected portion of the datagram. The encrypted datagram is followed by the ESP Trailer, that in turn is followed by the optional **ESP Authentication Data** field.

ESP's authentication service is similar to what is provided by AH.

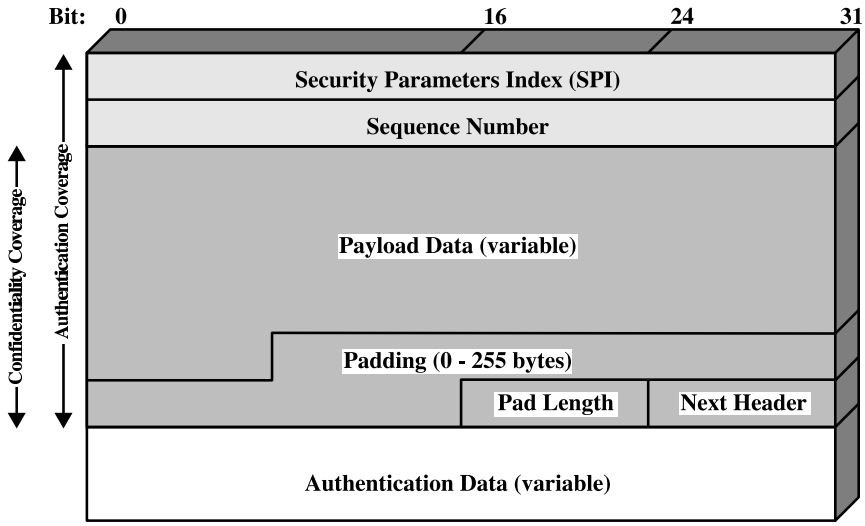
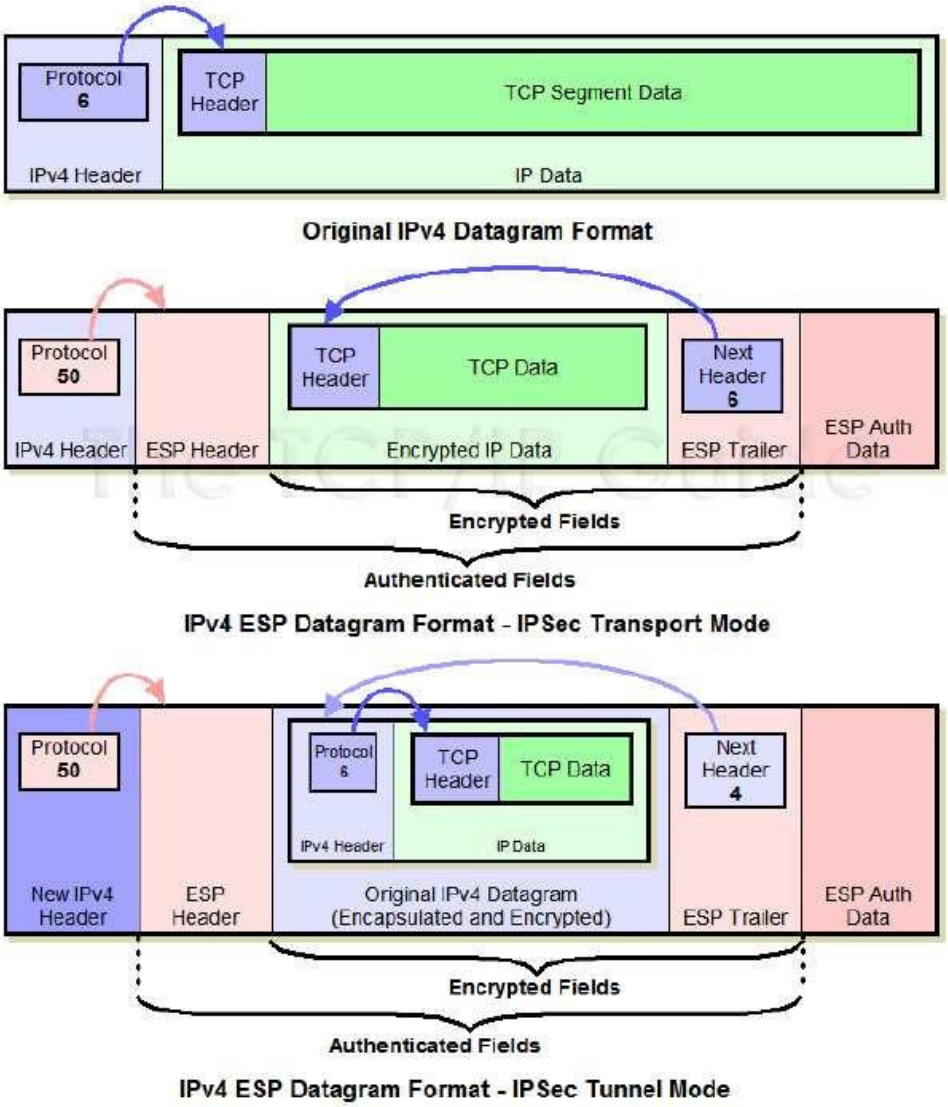
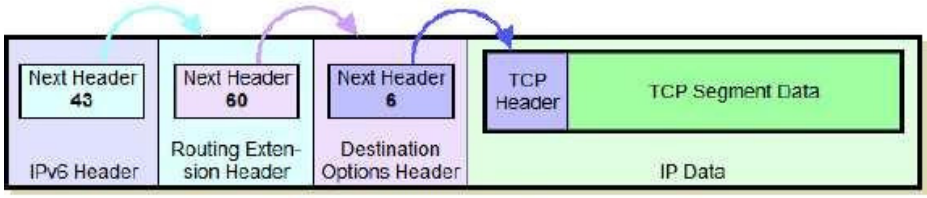



**Figure 16.7 IPSec ESP Format**

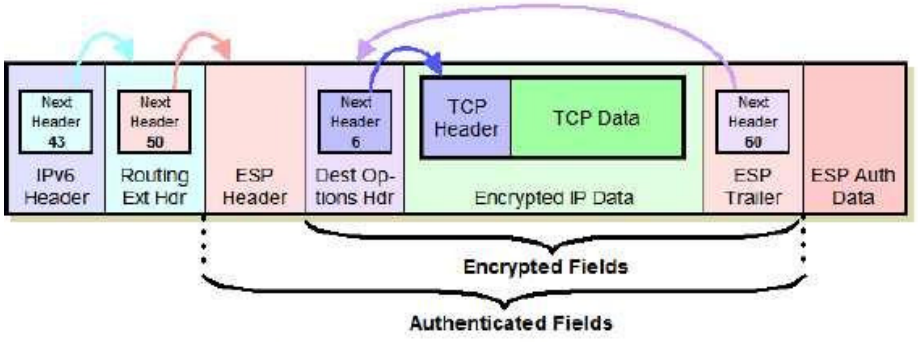
This figure is from Chapter 16 of Stallings: "Cryptography and Network Security", Fourth Edition



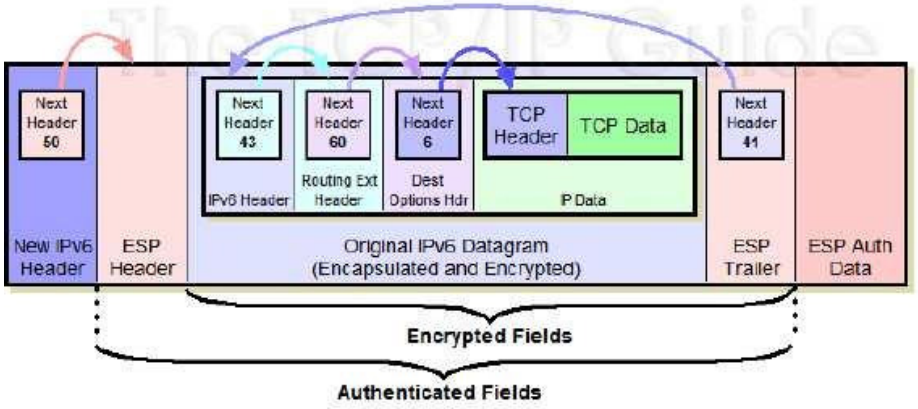

This figure is from http://www.tcpguide.com




**Original IPv6 Datagram Format (Including Routing Extension Header and Destination-Specific Destination Options Extension Header)**




**IPv6 ESP Datagram Format - IPSec Transport Mode**




**IPv6 ESP Datagram Format - IPSec Tunnel Mode**

This figure is from http://www.tcpguide.com

## IPSec Key Exchange

- Before ESP can be used, it is necessary for the two ends of a communication link to exchange the secret key that will be used for encryption. Similarly, AH needs an authentication key. Keys are exchanged with the **Internet Key Exchange** (IKE) protocol.

- IKE combines the functions of three other protocols:

  - The Internet Security Association and Key Management Protocol (ISAKMP) that provides a generic framework for exchanging encryption keys and security association information. ISAKMP supports many different key exchange methods.

  - The Oakley Key-Exchange Protocol. it is based on Diffie-Hellman algorithm but provided additional security. This is the default key-exchange method used by ISAKMP.

  - The SKEME protocol for key exchange. ISAKMP uses the re-keying feature of this protocol.

- Diffie-Hellman's computationally expensive modular exponentiation makes it vulnerable to a **clogging attack** in which a communication node spends an inordinate amount of time generating session keys if too many of them are requested all at once.

(A adversary may forge the source address of a legitimate party and send a public Diffie-Hellman key to an unsuspecting host, which then has to carry out modular exponentiation to compute the secret session key. But repeated receipts of the same request could clog up the host by causing it to spend all its time in modular exponentiation.) Diffie-Hellman is also vulnerable to the man-in-the-middle attack.

- Oakley thwarts the clogging attack by using a cookie-exchange between the two parties. A request for a secret session key must be accompanied with a cookie that is nothing but a pseudorandom number.

- Cookie exchange consists of each side sending a pseudorandom number to the other that must be acknowledged by the receiving party to the sending party. If the original requester for a secret session key was masquerading as someone else, they would never receive the cookie.

- A cookie is generated by hashing the IP source and destination addresses, the UDP source and destination ports, and a locally generated secret value.