Inside the IPsec Headers: AH (Authentication Header) and ESP (Encapsulating Security Payload)

Bogdan CRAINICU
"Petru Maior" University of Târgu Mureş

Abstract: *IPsec protocol provides a robust security mechanism at the network layer: secure office connectivity and secure remote access over the Internet. IPsec provides data origin authentication, data integrity, data confidentiality (encryption), and replay protection, through the proper choising of the security service: Authentication Header (AH), which allows authentication of the sender of data, and Encapsulating Security Payload (ESP), which supports both authentication of the sender and encryption of data.*

I. IP Security overview

In Internet, there are various protocols and mechanisms to secure data and traffic at different levels in the network. So, network security, at what level ?

Application-level security is implemented in end hosts. Transport-level security is protocol specific – Transport Layer Security (TLS) protocol provides authentication, confidentiality an integrity on top of Transmission Control Protocol (TCP) [1]. Network-level security reduces the implementation of security protocols at the higher layers and allows to create intranets and Virtual Private Networks (VPNs). Data Link-level security requires hardware devices for encryption, but this solution is not scalable and works well on dedicated links[2].

IPsec protocol provides a robust security mechanism at the network layer: secure office connectivity and secure remote access over the Internet, establishing extranet and intranet connectivity. The principal feature of IPsec is that it can authenticate and encrypt all traffic at the IP level. IPsec is below the transport layer and is transparent to applications – there is no need to change software on a user system [8].

The architecture specifications for IPsec consist of more RFCs, the most important of these are 2401, 2402, 2406, 2408. RFC 2401 defines the security services provided by IPsec (access control, data origin authentication, connectionless integrity, rejection of replayed packets, confidentiality-encryption, limited traffic flow confidentiality), how and where they can be used, how packets are constructed and processed [2].

The security mechanisms are implemented as extension headers at network layer: Authentication Header (AH) for authentication and Encapsulating Security Payload (ESP) header for encryption and authentication. IPsec protocols consist of AH, ESP, IKE (Internet Key Exchange Protocol) and ISAKMP/Oakley (Internet Security Association and Key Management Protocol). The RFC 2401 establishes the relationship between these protocols and components and how these protocols deliver together the capabilities described by the IPsec architecture (Figure 1.1):

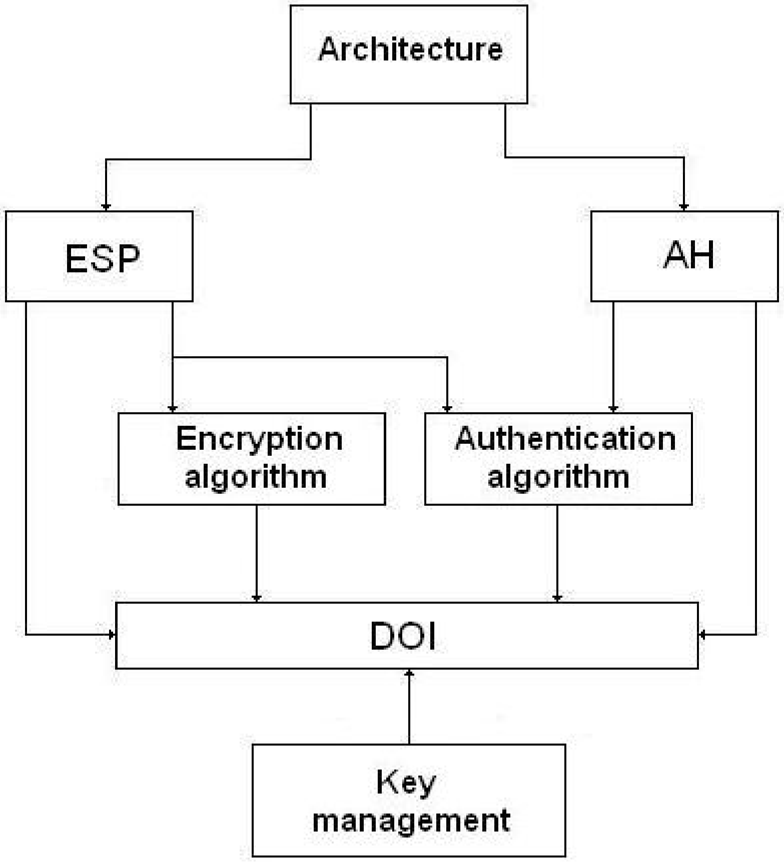

Figure 1.1 - IPSec architecture

- **Architecture**: covers definitions, general concepts , capabilities the devices should provide; the architecture document explains the sem antics of IPsec protocols, interactions between IPsec protocols and the rest of the TCP/IP protocols, and it does not specify the header format for AH and ESP [2].
- **Encapsulating Security Payload (ESP)**: defines the ESP packet format, the services it provides (encryption and, optionally, authentication) and the packet processing rules.
- **Authentication Header (AH)**: defines the AH packet format, the services it provides (authentication) and the packet processing rules.
- **Encryption Algorithm**: covers encryption algorithms and key size used for ESP and applied to the data to secure it; in addition, there are included any algorithmic-specific information.
- **Authentication Algorithm**: covers authentication algorithms used for AH and for the authentication option of ESP; in addition, there are included any algorithmic-specific information.
- **Key Management**: defines key management mechanism; IKE generates keys for the IPsec and for any protocol that need keys.
- **Domain of Interpretation (DOI)**: considered the master database of all IPsec negotiated parameters, ties together the IPsec documents by specifying all algorithms, attributes, operational parameters (as for instance key lifetime), and identifiers for approved encryption and authentication algorithms.

II. IPsec Transport Mode and Tunnel Mode

IPsec protocols, AH and ESP, support two modes of use: transport and tunnel mode. The operation of these two modes depends on what it is they are protecting, an IP payload or IP packet.

II.1. Transport Mode

In transport mode, AH and ESP protect the upper-layer protocols – TCP or UDP segment, ICMP packet. The transport mode can be used for end-to-end communication. For IPv4, the payload is the data that follow the IP header; for IPv6, the payload is the data that follow IP header and any IP v6 extension header that are present, with the possible exception of the destination options header, which may be included in the protection [8].

As a rule, when AH and ESP are used together in transport mode, ESP should be set up first, because the data integrity has to be calculated over as musch data as possible (Figure 2.1).

IP header	AH header	ESP header	TCP header	Data

Figure 2.1 – Packet with AH and ESP in transport mode

II.2. Tunnel Mode

Tunnel mode is normally used to protect the entire IP packet. IPsec encapsulates packet with AH or ESP and add a new outer IP header (Figure 2.2).

New IP header	ESP or AH	Original IP header	TCP header	Data

Figure 2.2 – IPsec packet in tunnel mode

An IPsec packet in tunnel mode has two IP headers. The inner (original) IP header is constructed by the node and no intermediate routers are able to examine it. The outer (new) IP header is added by the device responsible for providing the IPsec services and may have totally different source and destination address. Tunnel mode is used to implement VPNs, when both ends of an Security Association (SA – a one-way relationship between a sender and a receiver that affords security services to the traffic carried on it [8]) is a system that implements IPsec.

ESP in tunnel mode encrypts the entire inner IP packet and just optionally authenticates the entire inner IP packet. AH in tunnel mode authenticates the entire inner IP packet and some parts of the outer (new) IP header.

IPsec also allows nested tunnels, where it is possibly to create a tunnel for a tunneled packet (Figure 3.3).

IP header	ESP	IP header	AH	IP header	Data

Figure 3.3 – IPsec nested packet

III. **The Authentication Header (AH) format**

AH can be used to protect the upper-layer or the entire IP packet, and is based on the use of a message authentication code (MAC – a public function of the message and a secret key that produces a fixed-length value that serves as the authenticator [8]).

AH has assigned the number 51 as protocol number, indicating that following the IPv4 header is an AH header (for IPv6, the value of the next header field depends on the extension headers).

The AH header has the fields (Figure 3.1):

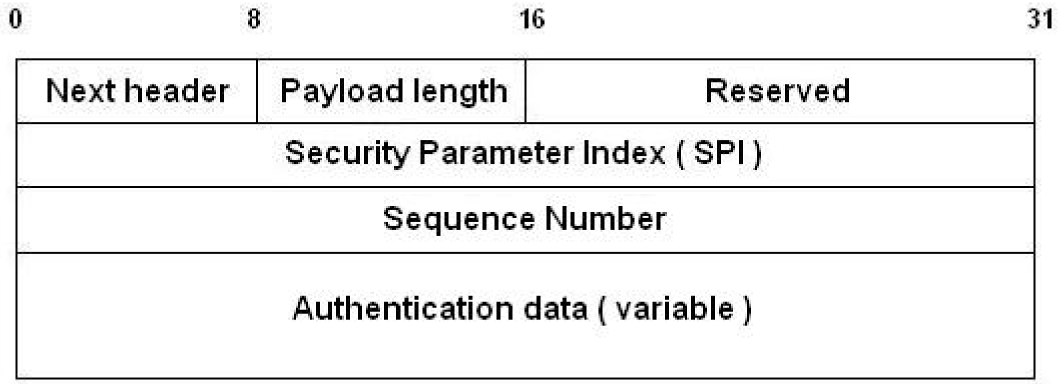

Figure 3.1 – AH header

- **Next header** (8 bits): indicates the type of header that follows the AH header (in transport mode – the value of transport layer protocol (TCP, UDP); in tunnel mode – the value is 4 for IPv4 and 41 for IPv6).
- **Payload length** (8 bits): indicates the length of AH in 32-bit words minus two.
- **Reserved** (16 bits): this field must be set to zero (for future use).
- **Security Parameter Index** (32 bits): an arbitrary value that, along with the destination address of the outer IP header, uniquely identifies a security association; the SPI value of zero is reserved for local, implementation-specific use and must not be sent on the wire [4].
- **Sequence Number** (32 bits): a monotonically increasing counter value that is always present even if the receiver does not enable the anti-replay service for a specific security association (the sender must transmit this field). The sender initializes the counter to 0 when a new SA is established; each time a packet is sent on this SA, the sender incremets the counter. If anti-replay is enabled, the transmitted Sequence Number must never be allowed to cycle past $2^{32} - 1$ back to 0, avoiding multiple valid packets with the same Sequence Number. Thus the sender's counter and the receiver's counter must be reset if the limit of $2^{32} - 1$ is reached (the sender and the receiver will negociate a new SA with a new key).
- **Authentication Data** (variable, integral multiple of 32 bits): contains the Integrity Check Value (ICV) or MAC for this packet. AH does not define an authenticator, but the curreant specification dictates that a normal implementation must support two authenticators: HMAC-SHA-96 and HMAC-MD5-96. In both cases, the HMAC value is computed, but the output is truncated to 96 bits (96 bits is the default length of the Authentication Data field). The authentication algorithm specification must define the length of the ICV and the comparison rules and processing steps for validation [4].
  Public key algorithms are too slow for data authentication, that is the reason no public key authentication algorithms have been defined for use with AH [2].
  The Authentication Data field may include explicit padding for ensuring that the length of the AH header is an integral multiple of 32 bits (for IPv4) or 64 bits (for IPv6).

III.1. **AH – Transport Mode**

In transport mode, AH is used for end-to-end authentication.

For IPv4, AH is placed after the original IP header and before the transport segment or before any other IPsec headers that have been inserted. The authentication process covers the entire packet, except for mutable fields in the IPv4 header that are set to 0 for MAC calculation [8].

For IPv6, AH is viewed as an end-to-end payload and is placed after the original IPv6 header and hop-by-hop, routing and fagmentation extension headers. The destination options extension header(s) could appear either before or after the AH header depending on the semantics desired. The authentication process covers the entire packet, except for mutable fields that are set to 0 for MAC calculation (Figure 4.2).

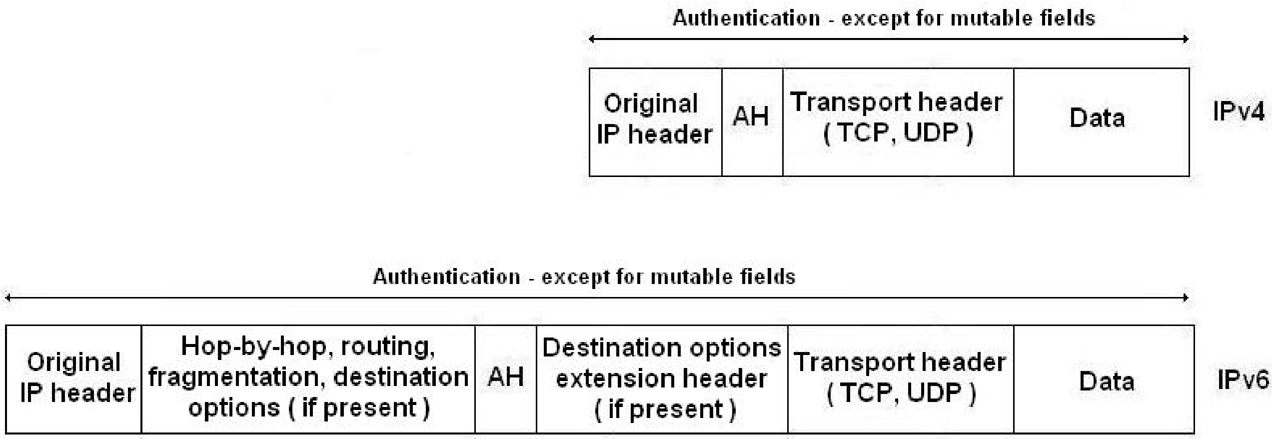

Figure 3.2 – AH transport mode

ESP and AH headers can be combined in a variety of modes. The IPsec Architecture Document describes the combinations of security associations that must be supported [3].

III.2. **AH – Tunnel Mode**

When AH is implemented in a security gateway, tunnel mode must be used. In this case, the AH encapsulates the protected datagram and it is inserted between the original IP header (inner header) and a new IP header (outer header). The inner header maintains the original source and destination address, while the outer header may contain different IP addresses (addresses of the IPsec endpoints).

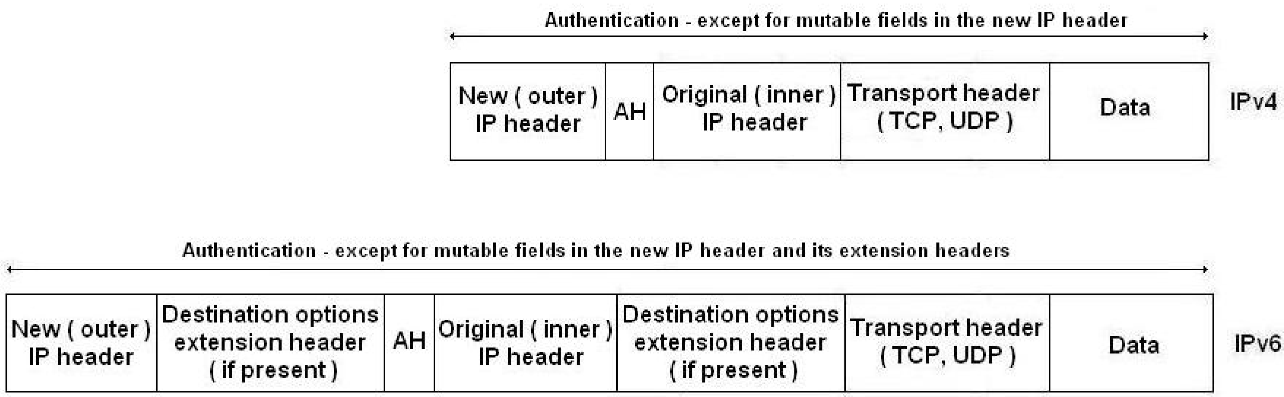

Figure 3.3 – AH tunnel mode

In tunnel mode, the entire inner IP packet is protected. Th e outer IP header (and for IPv6, the outer IP extension headers) is autheticated except for mutable and unpredictable fields (Figure 3.3).

IV. The Encapsulating Security Payload (ESP) format

ESP may be applied alone, in com bination with the Authentication Header (AH) or in a nested fashion through the use of tunnel mode. ESP has assigned the number 50 as protocol number indicating that following the IPv4 he ader is an ESP header (for IPv6, ESP is inserted after the hop-by-hop, routing and fragm entation extension headers, and before the destination options header; if extension headers are present, the next header fi eld of the extension head er immediately preceding the ESP header is set to 50, and in the absence of any extension head er, the next header field in the IPv6 header is set to 50 [2]). The format of ESP packets for a given SA is fixed, for the duration of the SA.

The ESP packet contains the following fields (Figure 4.1):

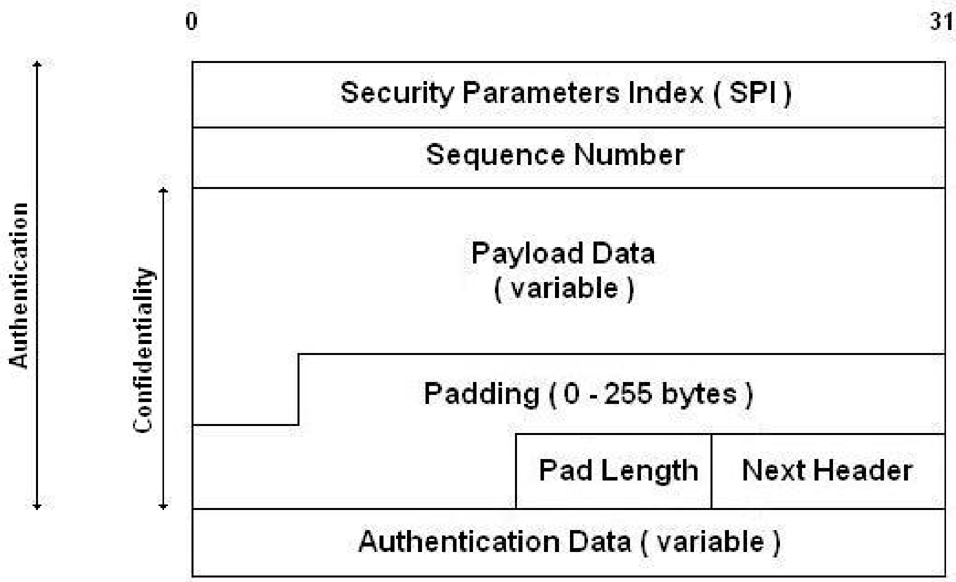

Figure 4.1 – ESP packet

- **Security Parameters Index** (32 bits): an arbitrary value that, along with the destination IP address and the ESP security protocol (the pr otocol in the preceding IP header), uniquely identifies a security association; the SPI value of zero is reserved for local, implementation-specific use and must not be sent on the wire [5].
- **Sequence Number** (32 bits): a monotonically increas ing counter value inserted into th e header by the source, that provides anti-replay servics. This field is not encrypted, but it is authenticated (thus, the anti-replay check can be performed prior to decryption). When an SA is established, both sender's counter and destination's counter are initialized to 0 and if anti-replay is enabled, the sender's Sequence Number must never be allowed to cycle.
  The source must always transmit this field, but the receiver need not act upon it [5].
- **Payload Data** (va riable): the a ctual data desc ribed by the Next Head er field and being protected by ESP (transport-level segm ent in transport mode and IP packet in tunnel mode). Any initialization vector (IV) used by the algorithm to encrypt the payload may be included explicitly in this f iled; accordingly, the encryp tion algorithm must define, as specifications that shows how the algorithm is used with ESP, the length and the location of

the IV. The IV is the first 8 octets of data in the protected data field (in some cases (depends on the operation mode), the destination treats the IV as the start of the ciphertext, in other cases, the destination reads the IV in separately from the ciphertext – the algorithm specification must address any alignament issues of the ciphertext).

- **Padding** (0 – 255 bytes): this field is motivated by several factors:
    - some encryption algorithms require that the input to the cipher be a multiple of its block size – Padding will expand the plaintext (Payload Data, Pad Length and Next Header fields) to the required size;
    - the ESP format requires that the resulting ciphertext must be an integer multiple of 32 bits (Pad Length and Next Header fields must be right aligned within a 32-bit word);
    - the ESP format may not require padding if the payload data already provides the necessary alignment, but up to 255 bytes of padding can still be included; this mechanism can be used to conceal the actual length of the payload data, in support of partial traffic flow confidentiality.

  If Padding are needed, but the encryption algorithm does not specify the padding contents, then ESP dictates that the first byte of the pad be the value 1 with subsequent padding bytes making up a monotonically increasing sequence: 1, 2, 3, ... When this padding scheme is employed, the value of the Padding should be inspected by the destination as an additional check for faultless decryption.
- **Pad Length** (8 bits): indicates the number of pad bytes added; this field is mandatory.
- **Next Header** (8 bits): identifies the type of data contained in the Payload Data field (an upper-layer protocol – TCP, UDP, or an IPv6 extension header).
- **Authentication Data** (variable – integral number of 32-bit words): contains an ICV (usually a keyed hash function) done on the ESP packet minus the Authentication Data. The length of the field depends on the authetication algorithm.
  The Authentication Data field is optional and is included only if an autheticator is specified in the SA.

The current IPsec specifications specifies that a standard implementation must support DES in cipher block chaining mode, but in the DOI document are specified in addition other algorithms: 3DES, RC5, IDEA, Three-key IDEA, Blowfish, CAST.

IV.1. ESP – Transport Mode

The ESP transport mode is applicable only to host implementations and provides protection for upper layer protocols, but not the IP header (Figure 4.2). The transport-

In transport mode, the ESP header is inserted after the IP header and prior to the transport-layer header (TCP, UDP, ICMP) or prior to the IPv6 destination options header. The ESP trailer contains the Padding, Pad Length and Nexter Header fields; if authentication is provided, the ESP Authentication Data field is added after the ESP trailer.

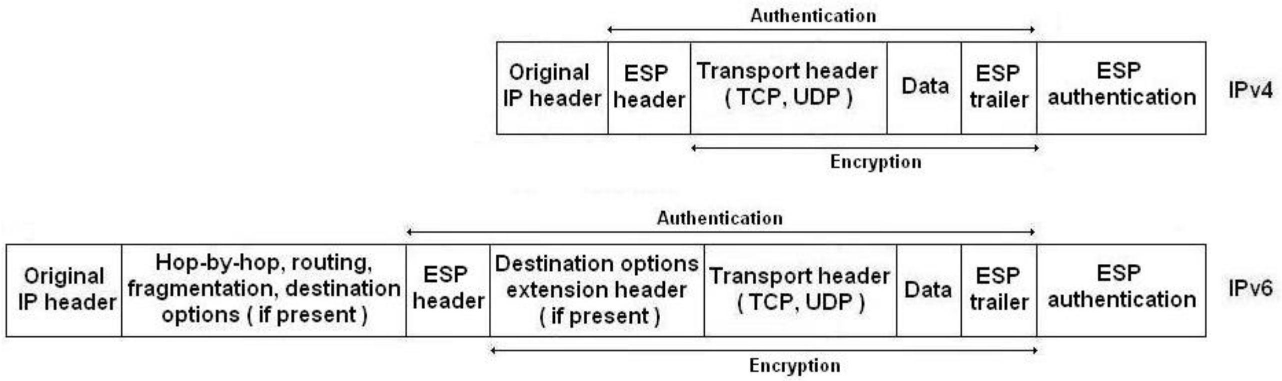

Figure 4.2 – ESP transport mode

In the IPv6 context, ESP is viewed as an end-to-end payload, and should appear after hop-by-hop, routing, and fragmentation extension headers. The destination options extension header(s) could appear either before or after the ESP header depending on the semantics desired [5].

IV.2. **ESP – Tunnel Mode**

Tunnel mode may be implemented in either hosts or security gateways and is used to encrypt an entire IP packet (Figure 4.3).

The ESP header is prefixed to the original IP packet; the original IP header contains the final source and destination addresses, and the new IP header may contains different IP addresses (addresses of firewall, router or security gateway).

The original IP packet plus ESP trailer is encrypted. The original IP packet plus ESP header and ESP trailer may be authenticated.

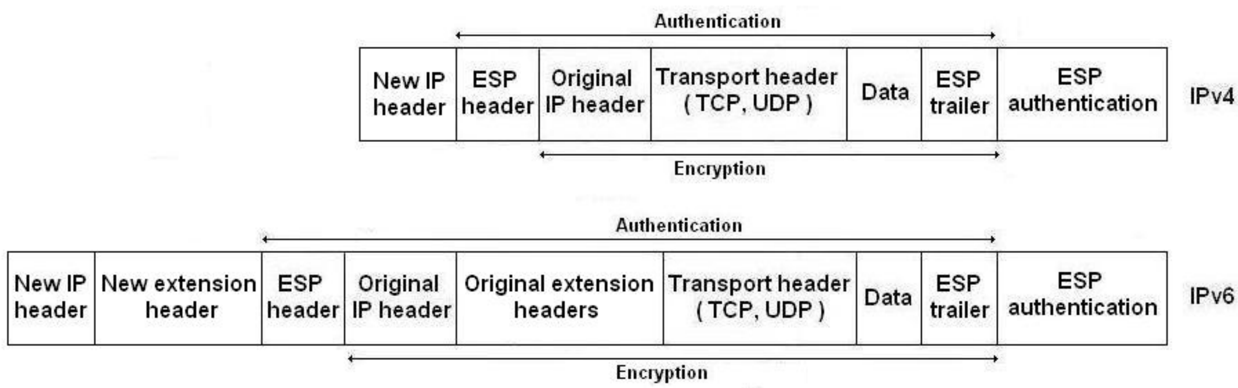

Figure 4.3 – ESP tunnel mode

V. Conclusions

The IPsec objectives - cryptographically-based securities for IPv4 and IPv6 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 [3]. The major benefits of IPsec remain the strong security capabilities that can applied to all traffic (encryption and authentication), no need to change the software for these services and no need to train users on security implementations (IPsec is transparent to end users).

References

1. Crainicu, Bogdan, An overview of the IPsec extension headers – AH (Authentication Header) and ESP (Encapsulating Security Payload), The 4th International Conference RoEduNet Romania, Sovata, May 2005

2. Doraswamy, Naganand, and Harkins, Dan, IPSec: the new security standard for the Internet, intranets, and virtual private networks, Second Edition, Prentice-Hall, 2003

3. Kent, S., Atkinson, R., Security Architecture for the Internet Protocol, Request for Comments: 2401, Network Working Group, 1998

4. Kent, S., Atkinson, R., IP Authentication Header, Request for Comments: 2402, Network Working Group, 1998

5. Kent, S., Atkinson, R., IP Encapsulating Security Payload (ESP), Request for Comments: 2406, Network Working Group, 1998

6. Madson, C., Glenn, R., The Use of HMAC-MD5-96 within ESP and AH, Request for Comments: 2403, Network Working Group, 1998

7. Madson, C., Glenn, R., The Use of HMAC-SHA-1-96 within ESP and AH, Request for Comments: 2404, Network Working Group, 1998

8. Stallings, William, Cryptography and Network Security: Principles and Practice (3rd Edition), Prentice Hall, 2002