IPsec basics

by Sorin M. SCHWARTZ

IPSec is a bundle of protocols and algorithms defining a flexible framework in which it is the user who selects the actual parameters of the algorithms and methods to be used.
As a result, one should assume that two IPSec implementation instances are not necessarily identical.
IPsec is defined in RFC 2401 to 2411 and 2451.

In a communications network, security has a few different aspects, the most important being:

Authentication	Making sure that the data received is indeed coming from the expected partner (avoiding unauthorized sources to transmit data to a station)
Integrity	Making sure that the data received is indeed what was transmitted by the source (avoiding modifications to the data by unknown parties, executed while the data traveled from source to destination)
Rejection of replayed data	Making sure that receivers identify and discard packets that have been already received (avoiding multiple executions of same command, generated by unknown parties)
Confidentiality	Making sure that nobody “listens and understands” the data on its way from source to destination

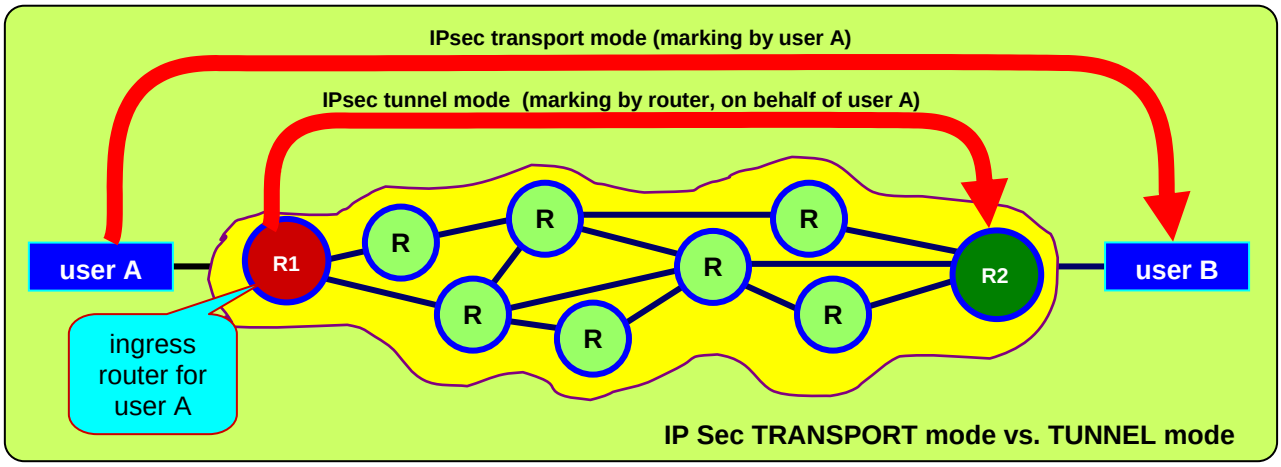
IPsec addresses all the above services and defines the necessary tools for their provision.

The basic idea of IPsec is to “mark” packets before being injected into the communications network, and use this “mark” at the receiving side in order to decide whether the packet arrived from the correct source (authentication), whether the packet content is exactly the one generated by the source, without any modifications (integrity) and whether the packet is not a replay of one of the previous packets, already received (rejection of replayed data).
In addition, IPsec also defines a framework for data encryption ensuring that potential “listeners” in the network would not be able to understand the information carried in the packet (confidentiality).

The “marking” process results in new fields being added to the packet to be protected.

The "marking" of the packets before their transmission may be executed by:
- the user computer (client), or by
- the ingress edge router (the first router met by the transmitted packet, the router connecting the user to the communications network)

When the user computer is the one marking the transmitted packets, it is said that IPsec is used in "*transport mode*".
When the ingress router is doing the job on behalf of the user (acting as a proxy IPsec entity), it is said that IPsec is used in "*tunnel mode*".

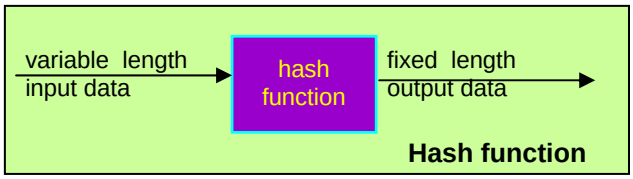


Authentication / integrity check principles

Authentication and integrity check processes are based on the addition to the packet to be protected of an Integrity Check Value (ICV) field allowing the receiving party to be sure that:
a.- The packet was indeed sent by the expected partner (authentication)
b.- The packet has not been modified (tampered with) during its trip through the network (integrity check)

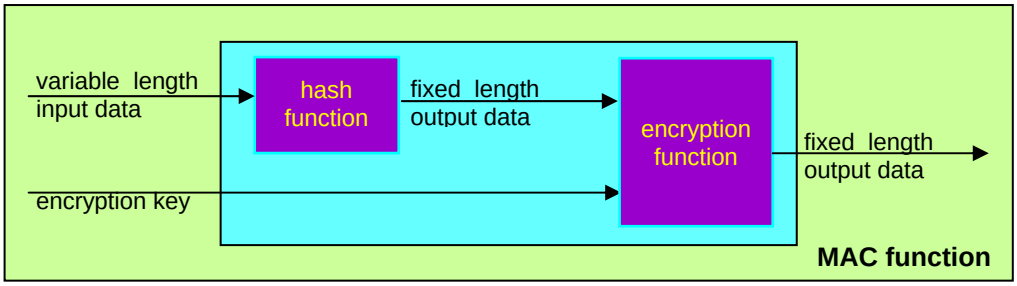
Definitions

- Hash function - a function that takes variable length input data and produces fixed length output data (= the hash value of the original input data) that can be regarded as the fingerprint of the input data.
  Hash functions should be collision resistant, i.e. it should be hard to find two different inputs generating the same hash value.



  - examples of hash functions:
    - SHA-1 - Secure Hash Algorithm 1 - generates 160 bit hash value (fixed length)
    - MD5 - Message Digest 5 - generates 128 bit hash value (fixed length)

- Message Authentication Code (MAC) - is a hash function that generates the hash value as a function of the input message AND an encryption key provided to it as input parameter.



  - examples of MAC functions:
    - HMAC-SHA-1-96 -Hashed Message Authentication Code based on Secure Hash Algorithm 1 - generates 160 bit hash value (fixed length)
    - HMAC-MD5-96 - Hashed Message Authentication Code based on Message Digest 5, generates 128 bit hash value (fixed length)

- Examples of encryption functions
  - DES - Data Encryption Standard - operates on blocks of 64 bits, has a 56 bits key used to derive 16 other keys used in the 16 phases (rounds) of the algorithm.
  - Triple DES - DES applied three times, using different keys
  - RC 5
  - IDEA - International Data Encryption Algorithm - operates on blocks of 64 bits, (128 bits key)
  - Triple IDEA
  - CAST

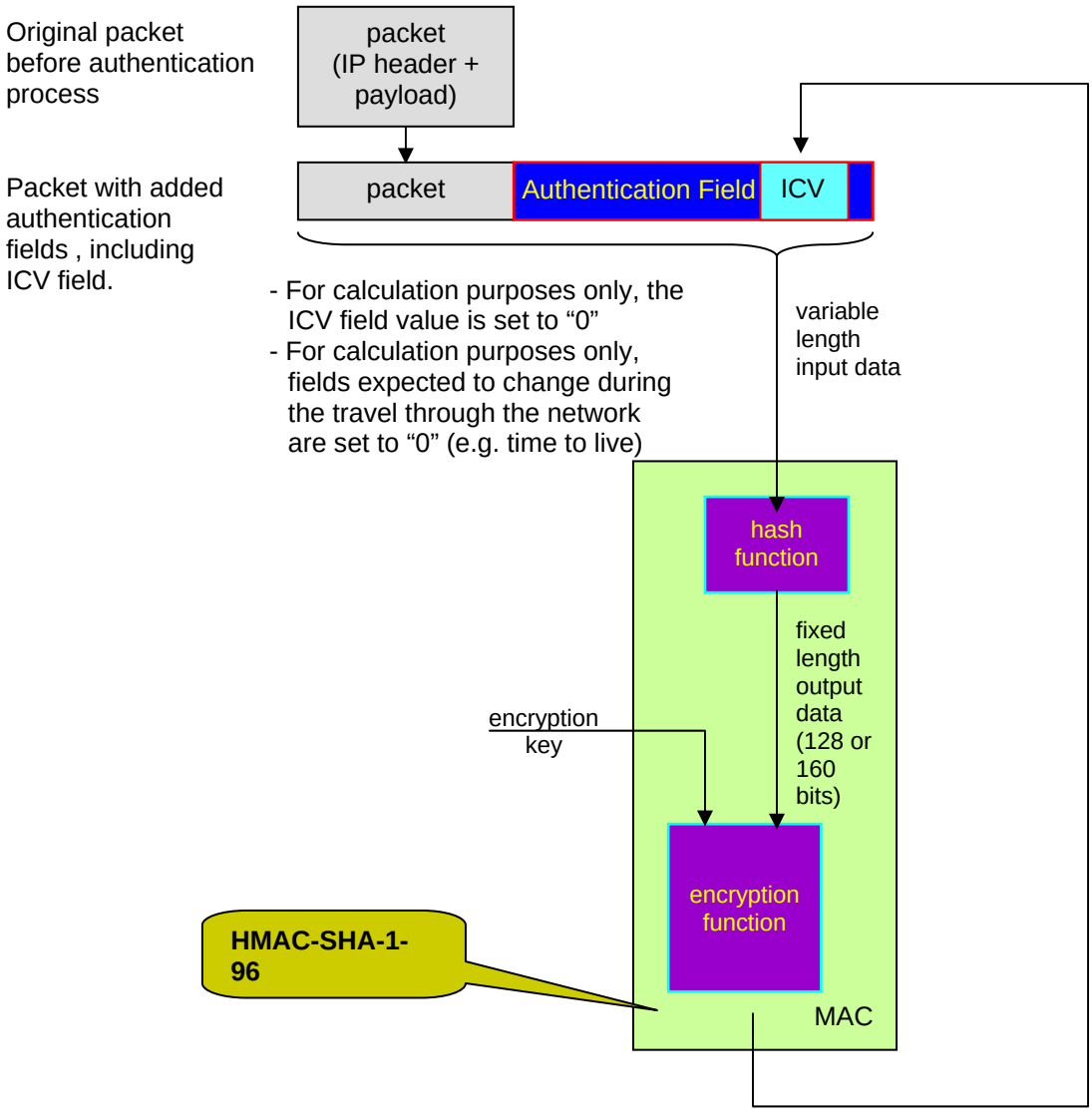
Generating and using the ICV

Observations:
1.- Messages encrypted with a specific key can be correctly decoded ONLY by using the same key.
2.- A decoding process using a specific key, applied to a message encrypted with a different key, results in a message different than the original one.
While the above observations seem trivial, they have a fundamental role in the understanding of the authentication process.

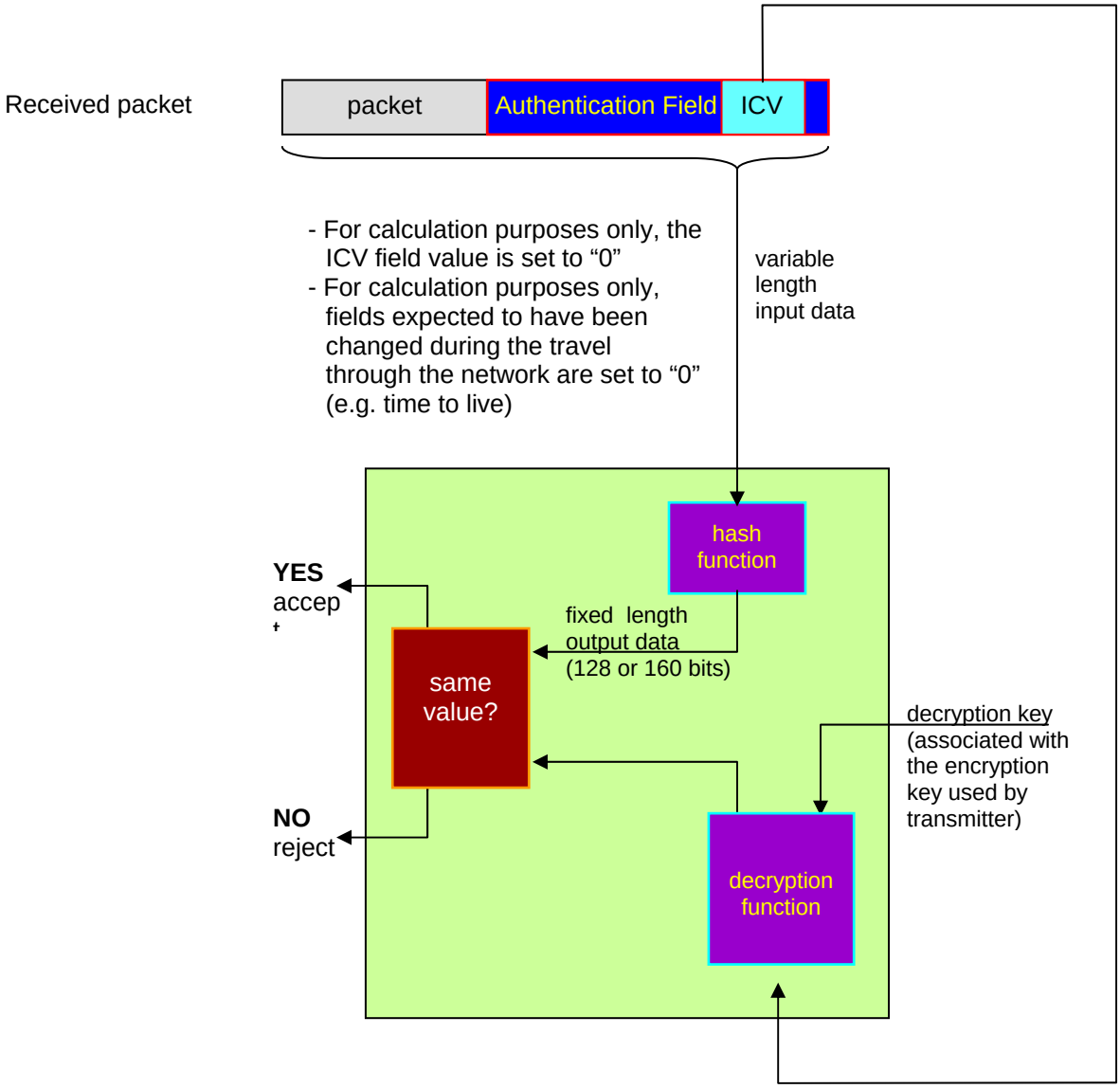
Generic Authentication process

Transmission

ICV is the result of two consecutive processes executed to the packet to be protected.
- step 1 - A hash value is calculated for the packet to be protected.
- step 2 - The hash value obtained in step 1 is encrypted using a secret key, generating the ICV value to be added to the packet.



Generic Authentication process
Reception



The receiving party calculates the hash value of the received packet. If the packet has not been tampered with, the hash value calculated at the reception should be the SAME as the hash value calculated by the transmitter.
The hash value as calculated by the transmitter is not available in the packet . . . but its encrypted value is there (the ICV).
Applying the decryption key (that has been agreed between the parties as being the one to be used for decoding messages arriving from that specific user) to the ICV, should result in the SAME hash value as the one calculated at transmitter!
If the values are indeed equal, the decoding process is considered to be successful, and the packet is accepted.
Packets for which the two values are different are considered as being corrupted, and are therefore rejected.

Generic Authentication process
Reception - Significance of correct decoding

1.- The fact that the hash function calculated at the reception is the same with the value obtained by decoding the ICV with the AGREED key, indicates that the original ICV was encrypted using the SAME AGREED key. Assuming that the agreed key is under the strict control of the expected partner, the fact that the values are equal indicate that the packet has been indeed originated by the expected partner. **(authentication)**

2.- The fact that the hash function calculated at the reception is the same with the value obtained by decoding the received ICV is also a sign that the content of the received packet is THE SAME as the content of the packet that has been transmitted. If any change would have been made in the original packet, the ICV would have been different! The only way to make an undetected modification in the transmitted packet would be to generate a new ICV, based on the modified content. But, in order to generate a new ICV, one needs the correct encryption key, which is assumed to be controlled ONLY by the known partner. The final conclusion is that indeed, the received packet is a correct replica of the transmitted packet **(integrity)**.

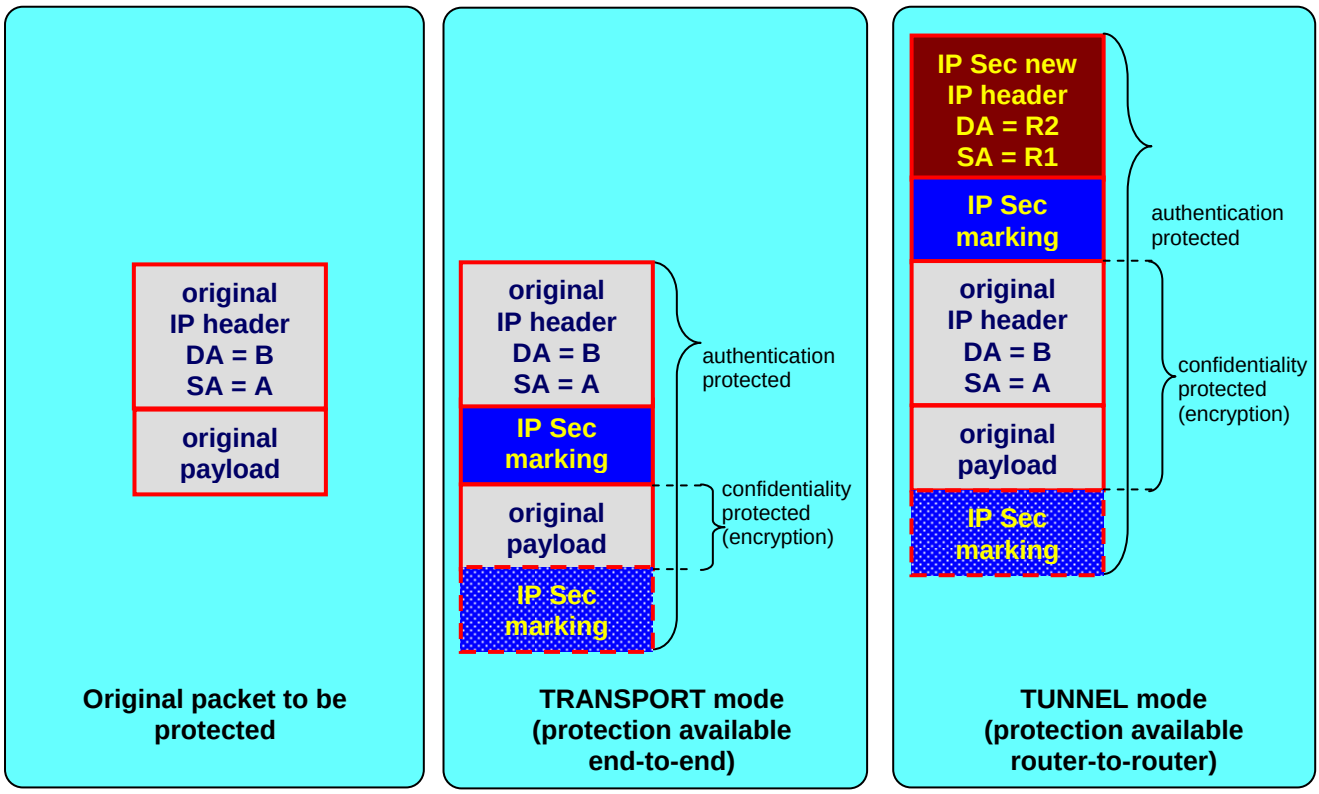
3.- Let's assume now that among the fields added in the packet before its transmission, a "sequence number" will be added. The sequence number is incremented by 1, for every transmitted frame. No two frames are generated with same sequence number.
If somebody would try to replay the transmitted packet, he could do one of the following:
- Catch the original packet and retransmit it (replay) as it is. This would generate at the receiving station two packets with SAME sequence number; one of the packets would be rejected, being detected as a replay.
- Catch the original packet, increment its sequence number by 1, and retransmit (replay) the packet. Incrementing the sequence number, means executing a modification in the packet. In order for this modification to go undetected by the receiver, the modifying entity should know the encryption key to be used, as explained in the paragraph above.
Successful decryption executed at the receiving party confirms therefore that the packet is not a replay **(rejection of replay data).**

Generic Authentication process
Comments
Basically, the authentication process is therefore based on an encryption process. While encryption processes are generally known as slow and processing power demanding, it should be noted that in the authentication methods described above, encryption is not executed over the whole packet. Instead, encryption is applied (only) to a short block (the hash value) of fixed and apriori known length, resulting in significantly faster operation.

Confidentiality principles

Confidentiality is obtained by encrypting the packet to be protected. Receiving party, and only it, should be able to decrypt the packet. Some identifiers, should however remain unencrypted, allowing the receiving party to identify the packet and to decode it with the correct algorithm, key, etc.

IP Sec marking location and protection areas

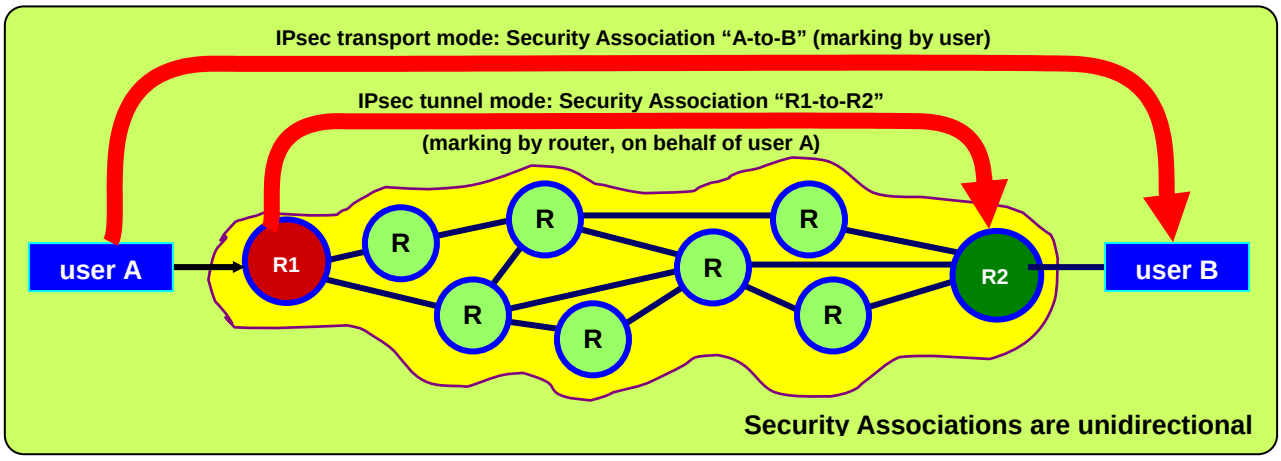


Depending on the specific IP Sec protocol used, and the selected mode (transport or tunnel), IP Sec marking appears in different locations in the protected packet.
When **authentication** is used, the marking protects the whole packet.
When **confidentiality** marking is used, the transport and tunnel mode provide different types of services.
In **transport mode** the encryption process is executed by the end station, and routers in the network are expected to route the encrypted packet toward its destination.
As a result, the encrypting station CAN NOT encrypt the IP header, as it includes the vital information needed by routers for correct operation. The only part that can be protected in this case is the payload itself (the upper layers).
In **tunnel mode** the encryption process is executed by the ingress router, and the resulting packet is sent via the network to another router (the other end of the tunnel) as indicated in a NEW IP header, added by the ingress router.
As a result, the ingress router CAN encrypt the WHOLE original packet (payload AND original IP addresses).

Both authentication and confidentiality services are based on packet marking executed at transmission site, followed by decoding executed at the receiving site.

Security Associations (SA)

Obviously, the receiving party should be aware of the marking executed by the transmitting party. The two parties, have to enter into a logical relationship, known as a “Security Association (SA)”, where the actual parameters regarding the algorithms and keys to be used are agreed.
Security Associations are unidirectional.
In a bi-directional link in which both directions have to be protected by IPsec, there will be two Security Associations: A-to-B and B-to-A.
Security Associations are uniquely identified by fields present in the protected packet, allowing the receiving party to associate a received packet to a specific Security Association and (as a result) to activate the correct algorithms for its processing.



IPSec consists of 3 major parts:

1.- AH - Authentication Header protocol (RFC 2402) - provides marking for authentication, integrity and replay protection

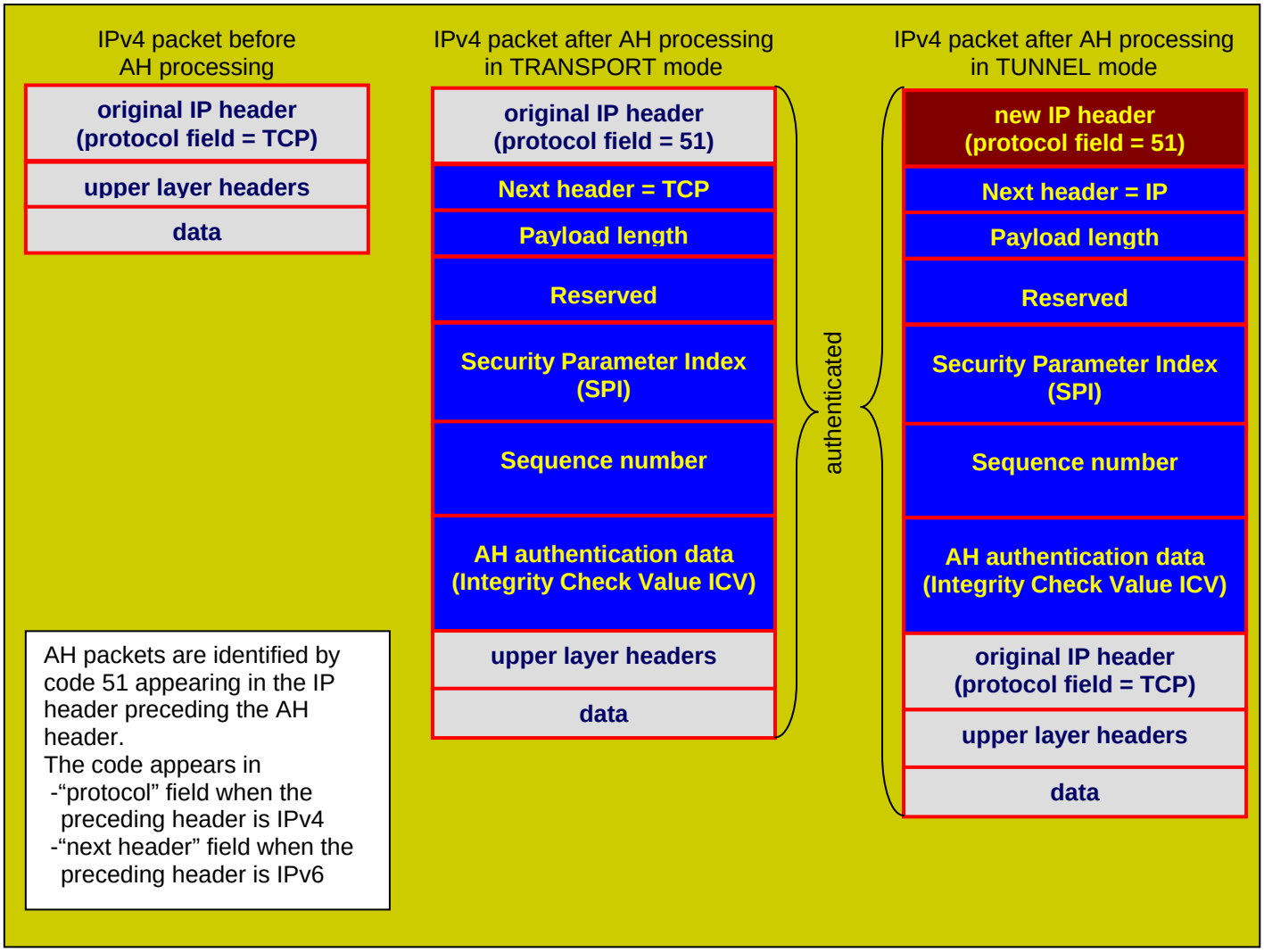
2.- ESP - Encapsulating Security Payload (RFC 2406) - provides marking for authentication, integrity and replay protection as well as for confidentiality.

As both authentication and encryption algorithms imply the use of encryption keys, one more element is needed: a protocol to handle the encryption key selection and other administrative issues related to the creation and maintenance of a Security Association between nodes.

3.- IKE - Internet Key Exchange (RFC 2409) - establishes and maintains Security Associations.

AH - Authentication Header protocol (RFC 2402)

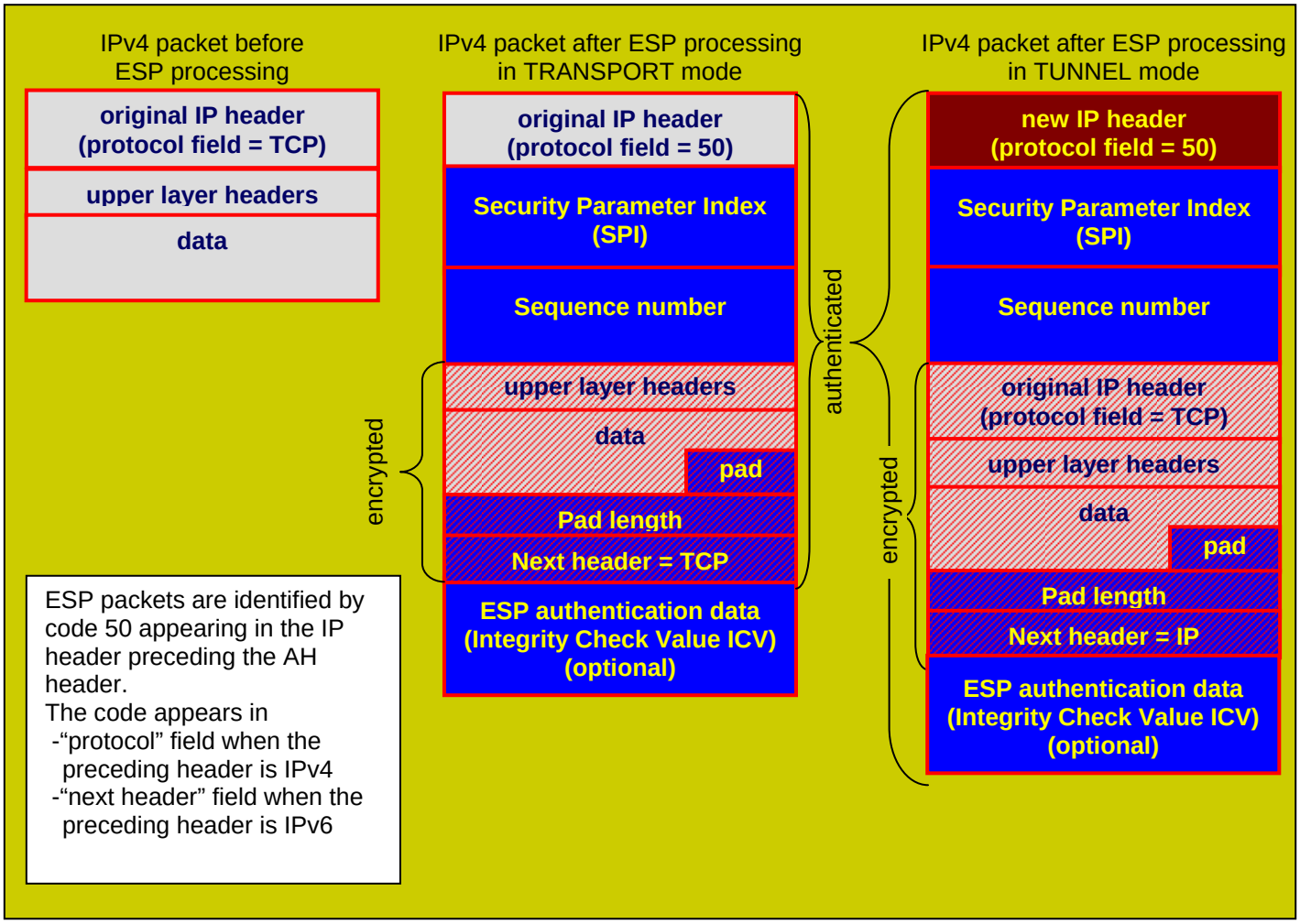
AH packet format



- Next header - type of next payload
- Payload length
- SPI - Arbitrary value that, together with the IP DA of the packet, identifies the actual Security Association (SA) to which the packet belongs.
  - Stations identify the SA to be used for the processing of an incoming packet, based on:
    - IP DA
    - SPI
    - security service (as indicated by the "protocol" value in the field preceding the security header - AH or ESP)
- Sequence number - avoids replay of the packet
- Authentication data - includes Integrity Check Value (ICV), used to authenticate the packet

ESP - Encapsulating Security Payload (RFC 2406)

ESP packet format



- SPI - Arbitrary value that, together with the IP DA of the packet, identifies the actual Security Association (SA) to which the packet belongs.
  - Stations identify the SA to be used for the processing of an incoming packet, based on:
    - IP DA
    - SPI
    - security service (as indicated by the 'protocol" value in the field preceding the security header - AH or ESP)
- Sequence number - avoids replay of the packet
- Pad length
- Next header - type of payload in the packet
- Authentication data - includes Integrity Check Value (ICV), used to authenticate the packet

Example of SA parameters for ESP based link

Protocol	ESP	
Operation mode	transport	
Encryption	3DES	
Integrity	HMAC-MD5	
Life time	7 min	
Party #1	SPI	21
	encryption key	xxxx
	integrity key	yyyy
Party #2	SPI	75
	encryption key	zzzz
	integrity key	www w

The actual hash and encryption algorithms as well as their parameters are selected by the parties during the establishment of their respective Security Association (SA).
IP Sec defines a framework for the selection of algorithms and protocols to be used, but it does not deal with the actual decisions taken by the parties.

IKE - Internet Key Exchange protocol (RFC 2409)

The purpose of the IKE protocol is to negotiate the protection parameters (protocols, keys, keys life time, etc.) to be used by the partners.

IKE is a protocol built as a combination of
- ISAKMP (Internet Security Association and Key Management Protocol) defined by NSA (National Security Agency) for the negotiation of the SA parameters, and
- OAKLEY protocol, based on Diffie-Hellman, for the selection of the encryption keys

For an effective protection of the data, the parameters negotiation should be itself protected.
IKE defines therefore two different phases:
- Phase 1 - Creates a protected environment (an SA) between the partners, to protect the negotiation of the authentication and encryption parameters to be used in the actual data transfer phase.
  Phase 1 can be executed either via the Main Mode protocol (MM) or via the Aggressive Mode protocol (AM).
- Phase 2 - Partners negotiate the protection parameters to be used in the data transfer phase. The negotiation is protected by the elements defined in phase 1.
  Phase 2 is executed through the protocol known as the Quick Mode.

Phase 1 Main Mode concept:
- Step a - One party sends an offer including all the parameters needed to define a Security Association (SA), EXCEPT the keys.
  Receiving party answers with the selected parameters.
- Step b - Parties exchange their public keys and some random sequences (nonce) to be used in step c.
- Step c - Using the keys and nonces exchanged in step b, parties authenticate each other.

At the end of these three steps, the two partners have been authenticated, and keys have been generated for the protection of the messages to be exchanged in Phase 2.

At the end of Phase 2, all the parameters and the keys necessary for operation are defined, and partners can start the actual data transfer phase, generating AH or ESP protected packets.