Bachelor's Thesis: Implementing Security in IoT Gateway in Legacy Devices for Telemedicine Applications. A comparison between Lightweight IPsec and CoAP

Perla Rocío Ramírez Sanabria

February 5, 2020

Contents

1 Introduction 5

2 IoT: Internet of Things 7

1 Definition 7

2 Applications 7

2.1 Application: Telemedicine 7

3 IoT Gateway 7

3 IPsec: Internet Protocol Security 9

1 Definition 9

2 Modes 9

2.1 Transport 9

2.2 Tunnel 10

3 Components of IPsec 10

3.1 Security Protocols 10

4 Key Management. Internet Key Exchange (IKE) 15

5 Lightweight IPsec 17

4 CoAP: Constrained Application Protocol 19

1 Definition 19

1.1 Messaging Model 19

1.2 Request/Response Model 19

2 Message Format 19

3 Message Transmission 19

4 Securing CoAP 19

4.1 DTLS: Datagram Transport Layer Security 19

4.2 Key Management 19

5 Test - Setup **21**

6 Test - Analysis **23**

7 Conclusion **25**

8 Discussion **27**

Chapter 1

Introduction

Chapter 2

IoT: Internet of Things

1 Definition

2 Applications

2.1 Application: Telemedicine

3 IoT Gateway

Chapter 3

IPsec: Internet Protocol Security

1 Definition

IPsec results from the abbreviation of Internet Protocol Security. This protocol provides a framework of open standards to ensure secure communications over IPv4 or IPv6, authenticating and encrypting each packet, thus ensuring communications at the network level and at all higher levels. Being optional for IPv4 and mandatory for IPv6.
IPsec can be implemented in several ways such as:

- Site-to-Site Connections (Networks Interconnection)
- Host-to-Host Connections
- Host-to-Site Connections (Remote Access)

And there exist several benefits about implementing IPsec:

- Transparent for protocols such as TPC and UPD as well for users and applications.
- It improves the security provided by other mechanisms such as SSL, TLS or SSH, operating all these at level 4 or higher.
- Possibility of integrating it into a perimeter security device.

2 Modes

IPsec operates in two possible modes, transport or tunnel which are going to be discussed in the following subsections.

2.1 Transport

When working with this mode the original IP packet is modified by inserting an additional header between the network and transport layer only protecting the information coming from the transport layer.

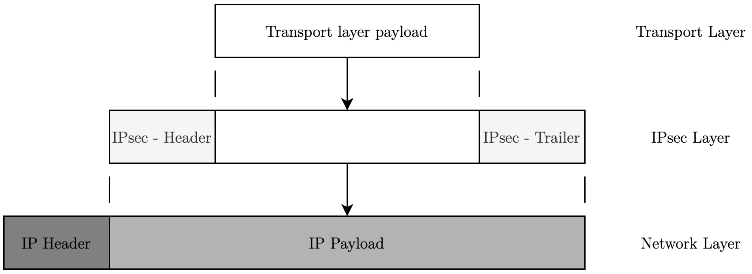

Figure 3.1: IPsec in transport mode

As it can be seen, the segment of the transport layer is encapsulated with a header and a trailer, this resulting encapsulation will be the final IP packet data field where the source and destination IP addresses are those of the two nodes that want to communicate securely.

2.2 Tunnel

When working with this mode the original IP packet is encapsulated entirely in a new IP packer, where the new IP header will have as origin and destination addresses the ends of the tunnel. Therefore, the entire packet is protected unlike when working with the transport mode.

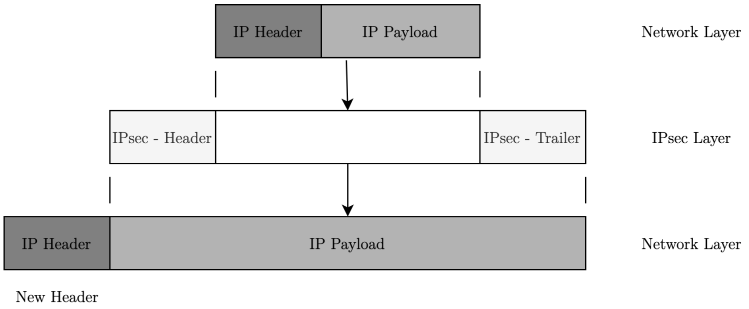

Figure 3.2: IPsec in tunnel mode

In tunnel mode it is the original IP packet the one being encapsulated with an IPsec header and a trailer. A new IP header is added to the resulting encapsulation, where, as already said before, the source and destination IP addresses are those at the ends of the tunnel.

3 Components of IPsec

3.1 Security Protocols

AH: Authentication Header

The AH security protocol provides authentication and integrity to the IP packets transmitted, but without encryption. This means that there is a guarantee that the data was sent by a legitimate sender and it has not been altered, but there is no guarantee for confidentiality. This will also

prevent IP packets' replay and at the same time it protects immutable fields in IP header.
In order to protect the IP header and data against modifications, a MAC (Message Authentication Code) function is applied to the majority of the bytes of the IP datagram, except for the fields that could change such as TOS, Flags, Checksum and TTL fragments offset. The minimum requirements for this MAC function are HMAC-MD5-96 and HMAC-SHA-1-96.

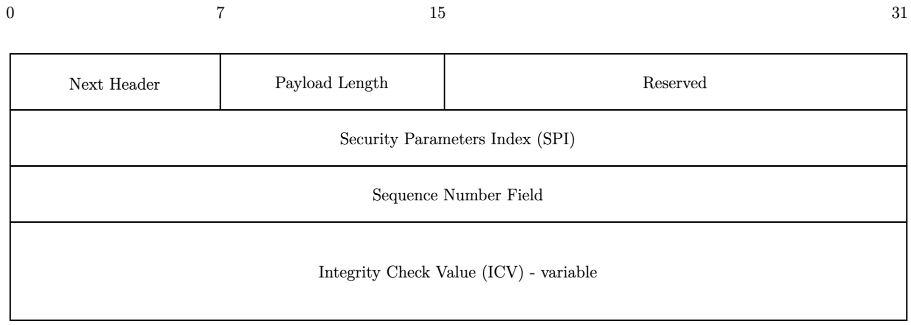
0
7
15
31
Next Header
Payload Length
Reserved
Security Parameters Index (SPI)
Sequence Number Field
Integrity Check Value (ICV) - variable

Figure 3.3: AH fields

The AH extension contains the following fields:

- **Next Header:** this field is 1 byte long and it is used in order to identify the type of the next payload after the Authentication Header. The value assigned to this field will vary depending on the protocol used and it is chosen from the set of IP Protocol Number that is defined by the IANA. For example, 1 for ICMP, 6 for TCP, 17 for UDP, etc.

- **Payload Length:** this field is 1 byte long and it is used to specify the length of AH in multiples of 4 bytes (32 bits), excluding the first 8 bits.

- **Reserved:** this field is 2 bytes long and it is reserver for future use. Since it is not used currently, the sender must set a value "0" for this field and it should be ignored by the recipient.

- **Security Parameters Index (SPI):** this field is 4 bytes long and it is used to identify the SA (Security Association), which will be discussed later in this document.

- **Sequence Number:** this field is 4 bytes long and it contains a counter value for all the packets sent, this means that with every packet sent this value will increase. The sender must include even if the recipient does not activate the anti replay protection, in which case it will be ignored by the latter.

- **Extended Sequence Number (ESN):** this field is 8 bytes long and it is used to support high-speed IPsec implementations by at the same time prolonging a SA's lifetime. This ESN must be negotiated when SA is created by the SA management protocol. This works as an extension of the already existing 4 bytes sequence number field.

- **Integrity Check Value (ICV):** this field has a variable length and it contains the ICV, which will provide authentication and integrity checking. Its length must be an integral multiple of 32 bits for both IPv4 and IPv6.

In the following figures we can see the AH encapsulation with the two possible modes:

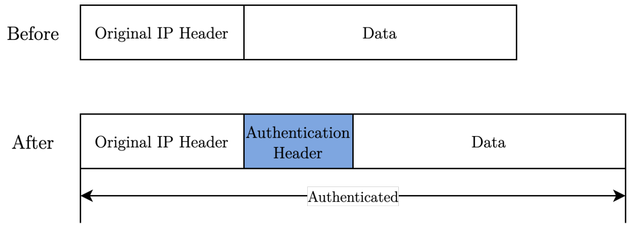

Figure 3.4: AH Encapsulation in transport mode

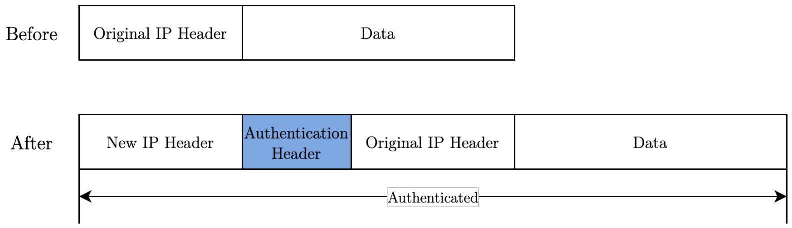

Figure 3.5: AH Encapsulation in tunnel mode

ESP: Encapsulating Security Payload

The ESP security protocol provides data integrity, source authentication and confidentiality of IP packets. However, ESP does not provide integrity and authentication for the entire IP packet, which is the reason why there exist the possibility to combine it with AH.
It guarantees that the content cannot be examined by third parties or, if possible, that it cannot be interpreted. It will be necessary to use a key in order to encrypt the data and different encryption algorithms can be used, such as DES, 3DES, RC5, etc.
It is important to note that ESP does not encrypt nor authenticate the IP header fields unless they are encapsulated by ESP using the tunnel mode.

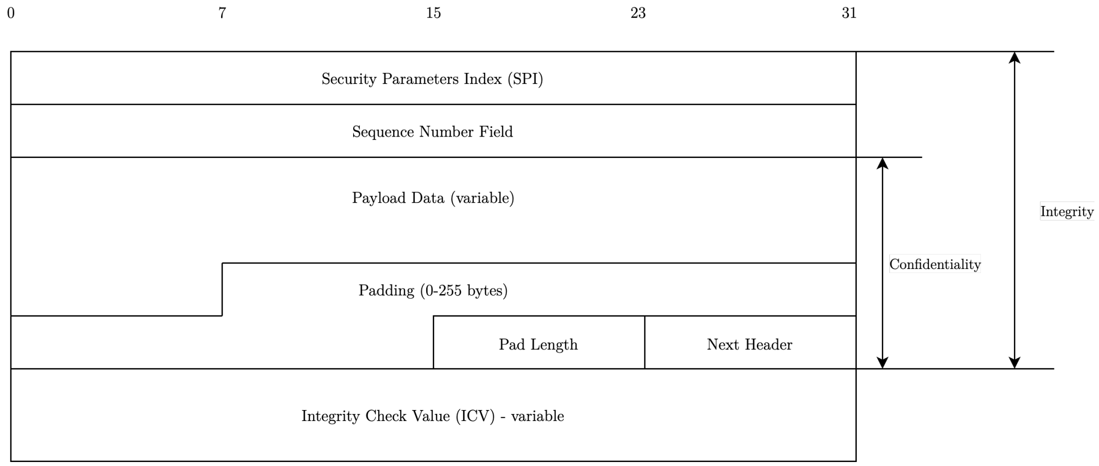

Figure 3.6: ESP fields

The ESP extension contains the following fields:

- **ESP Header:**
  - **Security Parameters Index (SPI):** this field is 4 bytes long and it is used to identify the SA (Security Association), which will be discussed later in this document.
  - **Sequence Number:** this field is 4 bytes long and it contains a counter value for all the packets sent, this means that with every packet sent this value will increase. The sender must include even if the recipient does not activate the anti replay protection, in which case it will be ignored by the latter.
  - **Extended Sequence Number (ESN):** this field is 8 bytes long and it is used to support high-speed IPsec implementations by at the same time prolonging a SA's lifetime. This ESN must be negotiated when SA is created by the SA management protocol. This works as an extension of the already existing 4 bytes sequence number field.
- **Payload Data:** this field has a variable length that will vary depending on the length of the data. Contains data described by the Next Header field as well as the initialisation vector for the encryption algorithm. It is the information that is actually protected by ESP.
- **ESP Trailer:**
  - **Padding:** there are two factors why this field is needed:
    1. The requirement for the plaintext to be a multiple of a certain number.
    2. The requirement for the resulting ciphertext to terminate on a 4-byte boundary.

    This field has a variable length that can go from 0 to 255 bytes depending on the requirements needed.
  - **Pad Length:** this field is 1 byte long and it is used to specify the length of the previously mentioned padding. It accepts values from 0 to 255, where a "0" value indicates the absence of Padding bytes.
  - **Next Header:** this field is 1 byte long and it is used in order to identify the type of the next payload after the Authentication Header. The value assigned to this field will vary depending on the protocol used and it is chosen from the set of IP Protocol Number that is defined by the IANA. For example, 1 for ICMP, 6 for TCP, 17 for UDP, etc.
- **Integrity Check Value:** this field has a variable length and it is the result of the computation over the ESP header, Payload, and ESP trailer fields.

In the following figures we can see the ESP encapsulation with the two possible modes:

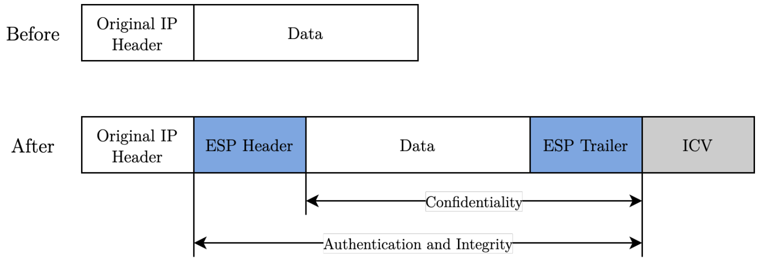

Figure 3.7: ESP Encapsulation in transport mode

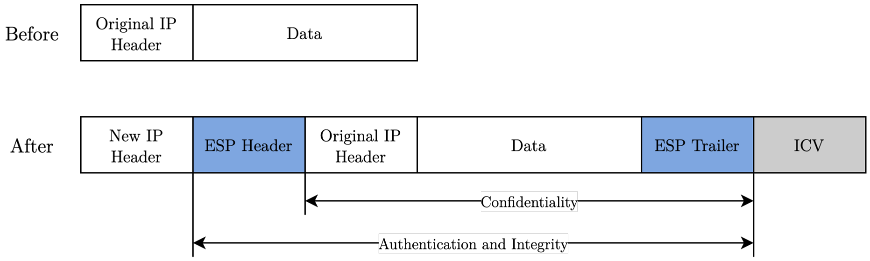

Figure 3.8: ESP Encapsulation in tunnel mode

Security Associations

An IPsec Security Association specifies the security properties for the traffic between the parties in a communication. Within these SAs the needed parameters such as encryption algorithms or keys are negotiated. These SAs are created dynamically by the Internet Key Exchange (IKE) protocol, which is going to be discussed in more detail later in the document.
An SA constitutes a simplex connection, which means that if the connection needs to be protected in both ways two SA's are required. Additionally, in every SA there is a possibility to use AH and ESP, but not simultaneously. Therefore, two SAs are required for each direction if both security protocols are used.
There are three parameters that defines an SA uniquely:

- Security Parameter Index (SPI)
- Destination IP Address (DA)
- Protocol identifier (P)

The three values previously mentioned are included in the data sent and in the Security Association Database (SAD) as we can see in the figure below:

Index	SN	OF	ARW	AH/ESP	LT	Mode	MTU
<SPI,DA,P>							
<SPI,DA,P>							

Figure 3.9: Security Association Database

A Security Association Database (SAD) a central repository that contains all the active SAs for both inbound and outbound processes. With each entry on the database the specific parameters for each SA are defined.

- **SN (Sequence Number):** 32-bit value used to generate sequence numbers for the security protocols.
- **OF (Overflow Flag)**
- **ARW (Anti-Replay Window):** detects inbound replayed packet.

- **AH/ESP:**
  - AH Information: Authentication algorithm, Keys, Key lifetime, others.
  - ESP Information:Encryption algorithm, Authentication algorithm, Keys, Key lifetime, Initialisation vectors, others.
- **LT (Lifetime):** specifies the lifetime for the SA.
- **Mode:** specifies the mode, which can be transport or tunnel.
- **MUT(Path MTU):** specifies the path MTU.

4 Key Management. Internet Key Exchange (IKE)

The IKE protocol is not a part of the IPsec specification, but it is integrated with it. As it was stated before in this document, it is the one in charge to create SAs, negotiate security policies, perform key exchange and authenticate the nodes.
It is a hybrid protocol based on the framework defined by Internet Security Association and Key Management Protocol (ISAKMP) and two other key management protocols: Oakley and SKEME. The latter protocols define a method with a view to establish an authenticated key exchange. The tasks for these protocols are:

- **ISAKMP:** provides a framework for authentication and key exchange by defining the format of the messages.
- **Oakley:** It is the protocol for determining symmetric keys, based on Diffie-Hellman, but with additional security.
- **SKEME:** “describes a versatile key exchange technique which provides anonymity, repudiability, and quick key refreshment” (RFC IKE)

Regarding the authentication, there two possible techniques:

- **Pre-Shared Key (PSK)**, from which an authentication hash is created and exchanged between the parties.
- **RSA Signature**, a hash is signed with the private key and then exchanged between the parties.

It divides its activity into two phases:

- **PHASE 1:** Negotiation of the IKE SA to protect phase 2 and it has two possible modes: main and aggresive.
- **PHASE 2:** Negotiation of SA for a data exchange protocol, in our case IPsec in each flow.

In the following figures we can se the two phases implementing both modes for PHASE 1:

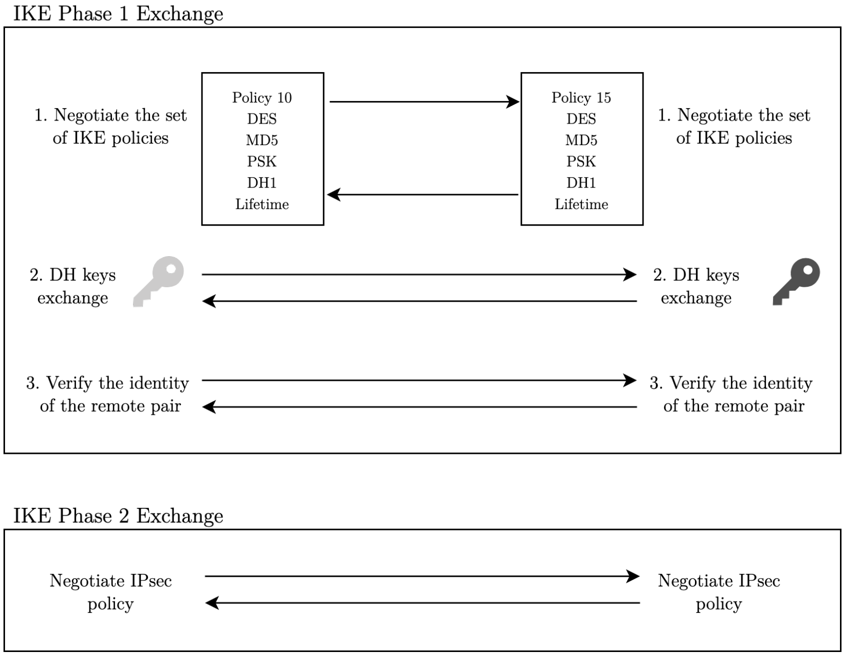

Figure 3.10: IKE: Phase 1 and Phase 2

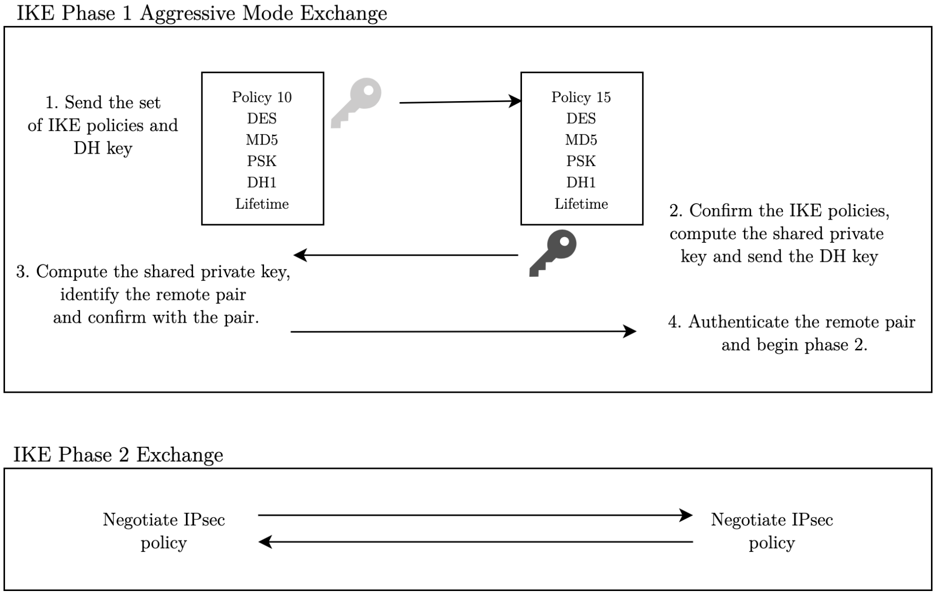
IKE Phase 1 Aggressive Mode Exchange
1. Send the set of IKE policies and DH key
Policy 10
DES
MD5
PSK
DH1
Lifetime
Policy 15
DES
MD5
PSK
DH1
Lifetime
2. Confirm the IKE policies, compute the shared private key and send the DH key
3. Compute the shared private key, identify the remote pair and confirm with the pair.
4. Authenticate the remote pair and begin phase 2.
IKE Phase 2 Exchange
Negotiate IPsec policy
Negotiate IPsec policy

Figure 3.11: IKE: Phase 1 (Aggressive Mode) and Phase 2

Finally in the following figure we can see the different algorithms used in order to establish a secure communication between two (or more) parties via IPsec implementation:

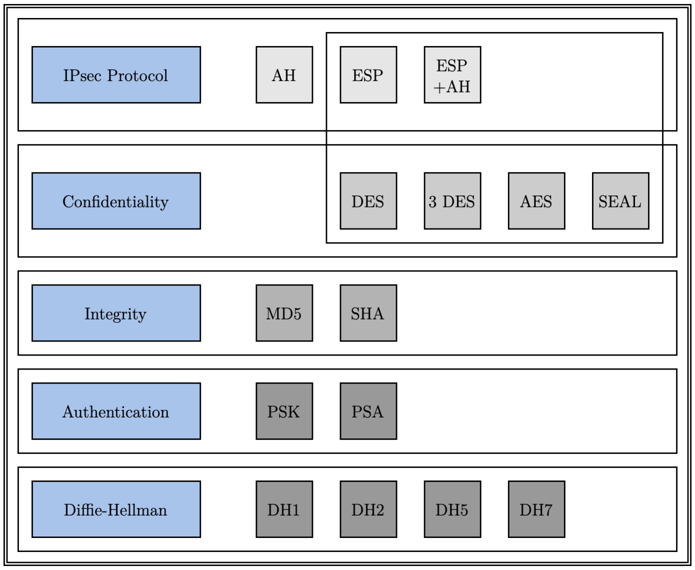

Figure 3.12: Algorithms

5 Lightweight IPsec

Chapter 4

CoAP: Constrained Application Protocol

1 Definition

1.1 Messaging Model

1.2 Request/Response Model

2 Message Format

3 Message Transmission

4 Securing CoAP

4.1 DTLS: Datagram Transport Layer Security

4.2 Key Management

Chapter 5

Test - Setup

Chapter 6

Test - Analysis

Chapter 7

Conclusion

Chapter 8

Discussion