83953 Advanced topics in communications protocols, seminar work

IPSec

Jaakko Sundquist
139478
jaakko.sundquist@nokia.com

Contents

1 Background

With the rapid growth of the Internet in the recent years and the simultaneous emergence of e-commerce, there has been an ever increasing need for mechanisms providing communications security in the Internet. The first solutions for the provision of security were methods that were tightly integrated in the applications themselves. These were clearly proprietary methods and every application needed to implement its own security systems. More recently a number of security solutions intended for a more general usage have emerged. The most important security standards in the Internet society so far are the *Transport Layer Security* (TLS) and the *IP Security* (IPSec), both standardised by the *Internet Engineering Task Force* (IETF), although TLS was originally developed by the Netscape corporation under the name of *Secure Sockets Layer* (SSL). This document deals with the IPSec standard.

The IP Security protocol suite, or IPSec for short, is a collection of protocols that provide privacy, authentication and integrity services at the Internet Protocol (IP) layer [IPSec]. It is intended to be used both as an additional protocol scheme with the IP protocol in use today (IPv4) and as an internal part of the next version of the Internet Protocol, that is, IPv6.

The main motivation for IPSec is that by introducing security at the IP level, all applications using the Internet or any other IP network can make use of these features, because no matter what the lower or upper layer protocols are, the network level protocol in the Internet is always IP. In other words, all applications, that are connected to the Internet, make use of the IP, thus if IPSec is introduced, all applications can also make use of its security methods (even transparently to the application itself).

A good book about IPv6, is [Huitema]. It also includes a quite clear explanation of the security features of IPv6, that is IPSec.

2 Overview of IPSec

2.1 IPSec protocols

The IPSec protocol suite defines two protocols to provide privacy, authentication and integrity to the IP packets traversing on the IP network. These protocols are the *Authentication Header* (AH) [AH] and *Encapsulating Security Payload* (ESP) [ESP]. Also an integral part of the suite are the key management and exchange methods, the most important of which is the Internet Key Exchange (IKE) protocol [IKE], formerly known as the Internet Security Association Key Management Protocol (ISAKMP/Oakley).

The AH and the ESP introduce their own header fields to the IP packet to which IPSec has been applied. With IPv6 IPSec is a natural part of the protocol specifications and takes the form of just another optional IPv6 extension header. With IPv4 they are extensions to the original protocol. With both cases the protocols take the same places with respect to each other, AH is always right after the previous extension header with IPv6 (the previous extension header has AH as the value of its next header field) and right after the normal IP header with IPv4. The ESP is then located right after the AH in both cases and in the ESP is included the payload of the next higher protocol (ESP has both a header and a tailer, AH has only a header), such as TCP or UDP. Both AH and ESP are optional, in other words both of them can be used without the other. The following figures illustrate the three cases with IPSec applied to IPv4. They also indicate the scopes of authentication and confidentiality provided for the packet. Note that all these examples illustrate the use of IPSec in *transport mode* (see section 2.4).

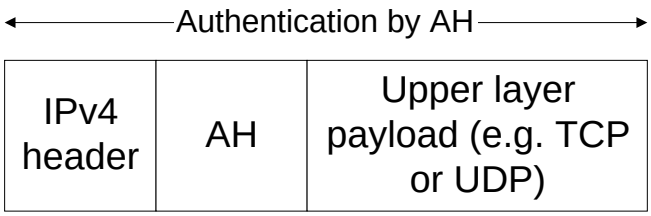

Figure 1: An IPv4 packet secured with only the Authentication Header (AH).

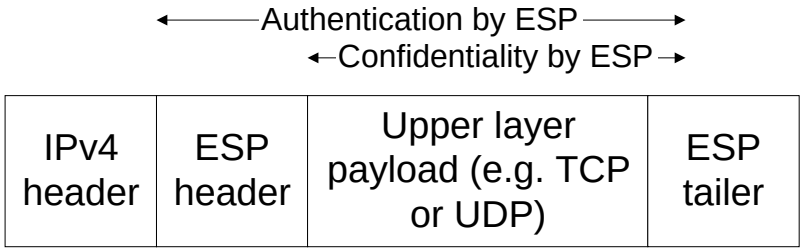

Figure 2: An IPv4 packet secured with only the Encapsulation Security Payload (ESP).

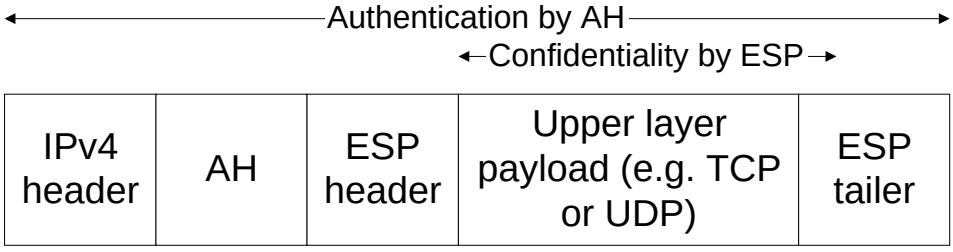

Figure 3: An IPv4 packet secured with both the Authentication Header (AH) and the Encapsulation Security Payload (ESP).

All of the IPSec protocols mentioned here will be given more detailed descriptions later in this document (in chapters 3, 4, and 5).

2.2 Cryptographic algorithms used with IPSec

The IPSec standard does not force the use of any specific algorithm for any of the different cryptographic needs of the protocols. It does, however, state one algorithm that every implementation of IPSec must have. This one algorithm is the DES algorithm in CBC mode and it is used with the user (or upper layer) payload encryption in the ESP protocol. This default algorithm is intended to guarantee at least some level of interworking between different implementations of IPSec, but as mentioned before, other algorithms can also be used for ESP encryption. Other often used algorithms for the payload encryption with ESP are for example 3DES (Triple-DES), Blowfish, IDEA, etc.

Both the AH and ESP have message authentication codes (MACs) in their headers and tailers. These MACs naturally also need to be implemented with some cryptographic algorithms. There are no mandatory algorithms for the MACs in IPSec, but almost the only ones that are used are the HMAC versions of the hash (or message digest) algorithms MD5 and SHA.

The IKE protocol uses the Diffie-Hellman key exchange method for agreeing the session keys between the participating entities. The authentication algorithms used with IKE are currently almost exclusively public-key algorithms. RSA is currently the most commonly used, but algorithms based on elliptic curves will probably also gain popularity in the future.

2.3 Security Associations (SAs)

A fundamental concept in the IPSec standard is the *Security Association* (SA). A Security Association is a relationship between two or more entities that describes how the entities will utilise security services to communicate securely. In other words, a SA is an object or a data structure in an IPSec capable endpoint which contains all the necessary information about the specific methods, how AH or ESP are used with a certain connection. This information includes for example the cryptographic algorithm, its mode, the keys to be used, how often the keys must be changed, etc.

The *Security Parameters Index* (SPI) is a number that uniquely identifies an SA. The SPI is included in both the AH and ESP headers, so the endpoints can identify the SAs used with an IP packet based on the SPI. Note that only one protocol can be included in one SA, that is, either AH or ESP. Thus, if both AH and ESP are needed for a connection, SAs for both protocols must be defined for the connection (actually total four SAs are needed, because inbound and outbound SAs are defined separately for an endpoint, even though they use the same parameters).

The SAs are negotiated with a key exchange protocol or they may even be configured manually. Several key exchange protocols are specified and can be used with IPSec, but the predominant and the only one that is defined as an IPSec related RCF is the *Internet Key Exchange* (IKE) protocol which is described in chapter 5 of this document. The SA payloads used in IKE messages when the SAs are negotiated between the endpoints are also presented in section 5.4.

2.4 IPSec modes of usage

There are two different ways to use IPSec, referred to as the modes of IPSec. These are the *tunnel mode* and the *transport mode*.

When IPSec is used in the transport mode between two computers (IPSec is always used in a point-to-point configuration, at least currently), the AH and/or ESP are used to protect the payload received from the upper layer (e.g. TCP or UDP). The header of the resulting IP packet is the same as it would be without IPSec. Figure 3 illustrates an IP packet that is secured with AH and ESP operating in transport mode.

In tunnel mode, the payload, that the AH and/or ESP are used to protect, is already a complete IP packet. The IP packet that results from tunnel mode IPSec has an outer IP header followed by the AH and/or ESP headers and after them the payload IP packet and ESP trailer. So in tunnel mode the whole IP packet is inserted into another IP packet (this inner IP packet may also be a tunnel mode IP packet carrying yet another IP packet and so

on).Thus, in this case the payload of the ESP protocol is the whole payload IP packet and the scope of the AH is the whole outer IP packet. The following figure illustrates the IPSec tunnel mode when both AH and ESP are used.

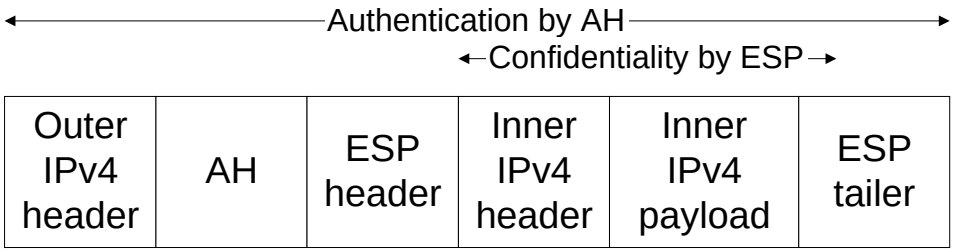

Figure 4: IPSec Tunnel Mode.

Note that with tunnel mode, the IP addresses in the inner IP header are concealed. Also, the addresses in the outer IP header need not be the same as inn the inner IP header, so the source and destination of the original (inner) IP packet can be hidden in the transmission path between the endpoints that are performing the tunnel mode transmission. Tunnel mode is used to connect the networks of different sites of an organisation. This is done with gateways on the outer edges of the networks of each site performing the tunnel mode IPSec operations on packets transmitted between the sites. This solution is called a Virtual Private Network (VPN), as it gives the same confidentiality to the organisation's network traffic as traditionally has been gained with the usage of private (or leased) lines between the sites.

As mentioned above, the tunnel mode is used in the communications between security gateways (often these perform address translation operations even without IPSec). In fact tunnel mode must be used between security gateways according to the IPSec standards. Correspondingly transport mode is more suited for communications between ordinary hosts that usually do not need confidentiality for their addresses. Transport mode also creates less overhead than tunnel mode. Tunnel mode can be used with normal hosts also, but transport mode is the more common in this case.

3 Authentication Header (AH)

The purpose of the Authentication Header is to provide authentication and integrity to the whole IP packet, including the IP header. Some fields in the IP header must, however, be left out of the examination, because they may (and will) change in transit (e.g. the Time-to-Live field of IPv4 header). The structure of the AH is illustrated in the following figure.

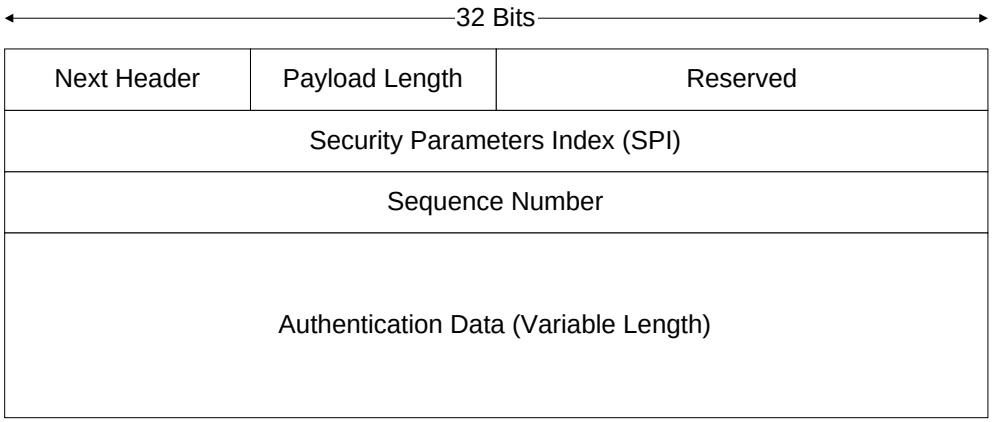

Figure 5: The structure of the Authentication Header (AH).

As mentioned earlier, the AH is located immediately after the normal IP header, when IPv4 is used and after the other extension headers (except the ESP) when IPv6 is used. The value for the Protocol field of IPv4 header and the value for the Next Header field of the IPv6 header or extension header preceding the AH header is 51, indicating AH.

The fields of the AH are explained in the following list:

- **Next Header** (8 bits) indicates the type of the next payload (protocol) in the packet after AH.
- **Payload Length** (8 bits) specifies the length of AH in 32-bit words minus 2. This peculiar encoding of the length is caused by IPv6, where *Hdr Ext Len* field is encoded by first subtracting 1 from the header length.

IPv6 uses 64-bit words, so two 32-bit words are subtracted in AH Payload Length because of that.

- **Reserved** (16 bits) is reserved for future use. In current implementations, all bits must be set to zeroes.

- **Security Parameters Index (SPI)** (32 bits) is the arbitrary value that together with the destination IP address and the security protocol in question (that is, AH in this case) identifies the Security Association used with this IP packet.

- **Sequence Number** (32 bits) is an unsigned integer field that contains a monotonically increasing counter value. This can be used to protect against replay attacks and if it this anti-replay is enabled (as is the default case), the transmitted Sequence Number is never allowed to cycle. The counters at both the transmitting end and the receiving end are always initialised to zero when an SA is established and before the 2^{32}nd packet can be sent, the counters must be reset by establishing a new SA (with a new key).

- **Authentication Data** (n * 32 bits) is the field that contains the *Integrity Check Value* (ICV) for the packet. This is the value obtained with the message authentication code algorithm calculated over the whole IP packet (in fact ICV is just another name for MAC). The length of the AH header must be a multiple of 32-bit words when IPv4 is used and a multiple of 64-bit words when IPv6 is used. Thus the Authentication Data field may contain explicit padding, if the length of the ICV is not appropriate. All IPSec implementations must support this kind of padding.

4 Encapsulating Security Payload (ESP)

The purpose of the Encapsulation Security Payload is to provide confidentiality to the payload received from the next higher protocol layer, such as TCP or UDP. This confidentiality is obtained through the usage of some encryption protocol. As mentioned, every IPSec implementation must incorporate the DES algorithm for this purpose, but other (symmetric) algorithms can also be used. ESP may also provide authentication and integrity for the payload, but it does not offer these services to the header of the IP packet (or the AH obviously). The authentication is an optional feature of ESP. The use of ESP authentication must be negotiated as a part of the ESP SA for a connection, if it is required. Usually, when AH is used, there is no need for ESP authentication, although both can be used simultaneously. The next figure describes the structure of the ESP header and tailer as well as the location of the payload with respect to these.

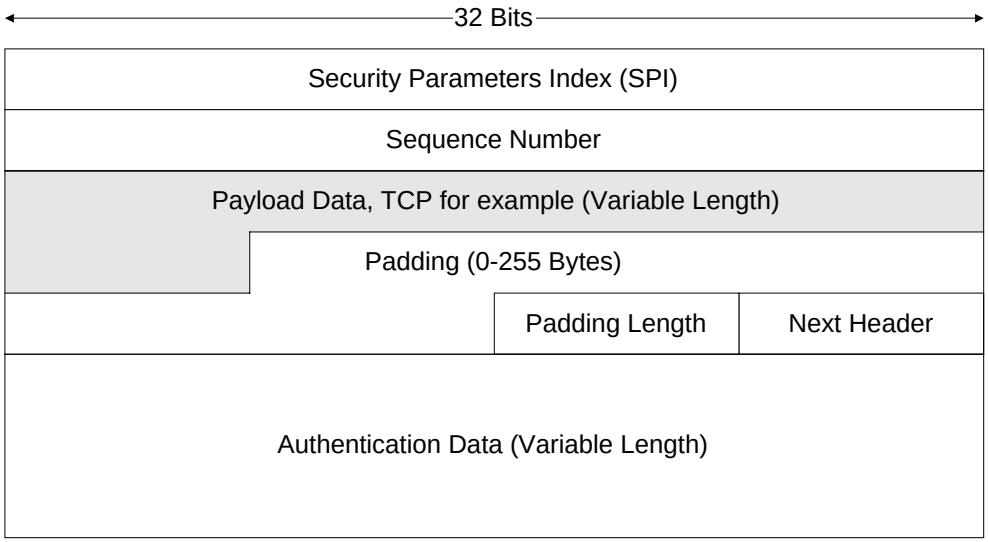

Figure 6: The structure of the Encapsulation Security Payload (ESP).

The ESP is located after the AH, if the AH is used, otherwise the ESP is located right after the IPv4 header or the last IPv6 extension header. The value for the Protocol field of IPv4 header and the value for the Next Header

field of the IPv6 header, extension header or AH preceding the ESP header is 50, indicating ESP.

The fields of the ESP are explained in the following list:

- **Security Parameters Index (SPI)** (32 bits) is the arbitrary value that together with the destination IP address and the security protocol in question (that is, ESP in this case) identifies the Security Association used with this IP packet.

- **Sequence Number** (32 bits) is an unsigned integer field that contains a monotonically increasing counter value. The field is used in the same way as with AH (see chapter 3).

- **Payload Data** is a variable length field containing the data received from the next higher protocol, described in the Next Header field. The field is mandatory and it is an integer number of bytes in length. If the encryption algorithm used with ESP requires some synchronisation data in each packet, e.g. an *Initialization Vector* (IV), this data can be carried in the Payload Data field. If this kind of algorithm is used, the ESP specification requires that the details of the usage of the algorithm with ESP are specified in a separate RFC.

- **Padding** (0 – 255 bytes) can be needed for a couple of reasons. The encryption algorithm may be such that it requires the plaintext to be a multiple of some integer of bytes (i.e. the block-size of a block cipher). Padding may also be needed in order to align the Pad Length field (next field after payload & padding) in the correct location in the ESP structure (see Figure 6). Another reason for padding may be the concealing of the length of the actual payload, although this kind of extra padding naturally increases the needed bandwidth for the data. Inclusion of padding is optional, but all implementations must support the generation and consumption of it.

- **Pad Length** (8 bits) indicates the number of pad bytes immediately preceding it. The field is mandatory and can contain the values 0 – 255, with the value zero meaning that no padding bytes are used.

- **Next Header** (8 bits) indicates the type of data (protocol) contained in the Payload Data field. It may be a higher layer protocol, such as TCP or UDP, or it may be an IPv6 extension header, if IPv6 is used.

- **Authentication Data** is a variable length field that contains an Integrity Check Value (ICV) (see chapter 3 for more about ICV) computed over the ESP packet minus the Authentication Data. This is an optional field and is included only, if the ESP authentication service is selected for the SA that the packet (or connection) is using.

The ESP provides confidentiality by encryption for the Payload Data, Padding, Pad Length and Next Header fields, the remaining IP packet is not concealed.

The Authentication Data field of ESP provides authentication and integrity for the SPI, Sequence Number, Payload Data, Padding, Pad Length and Next Header fields, thus it does not protect the AH or the IP header (IPv4 or IPv6 plus extensions). If the whole IP packet must be provided authentication and integrity, AH must be used. Both the AH and ESP authentication can be used in the same connection, but generally this is not very feasible.

5 Internet Key Exchange (IKE)

Before secured communications can occur between two peer entities with the aid of IPSec, the Security Associations used with that connection must be agreed upon by both entities, which is the function of the *Internet Key Exchange* (IKE) [IKE] protocol.

The IKE protocol used to be called ISAKMP/Oakley. In fact it consists of these two specifications. The name was quite recently simplified (for obvious reasons) to IKE.

5.1 Internet Security Association and Key Management Protocol (ISAKMP)

The *Internet Security Association and Key Management Protocol* (ISAKMP) [ISAKMP] provides a very generic framework for the key exchange protocol. It specifies an ISAKMP header that is used with every ISAKMP (or IKE) message and 12 payloads that can be inserted to a list following the ISAKMP header. The following figure illustrates the ISAKMP header structure.

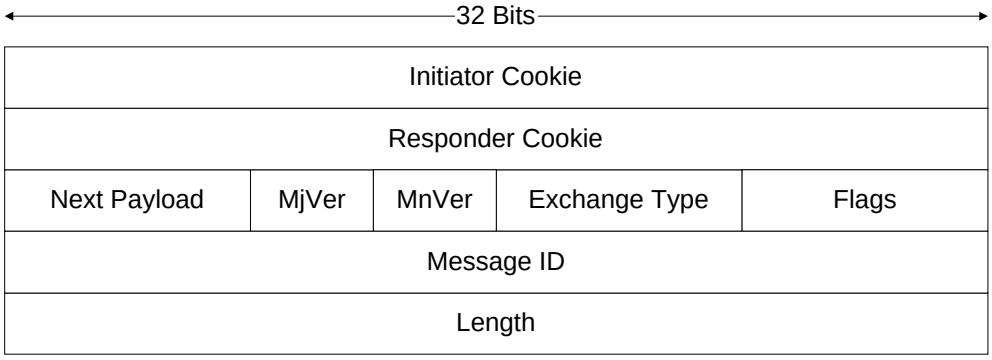

Figure 7: The ISAKMP Header.

The purposes of the header fields are described in the next list.

- **Initiator Cookie** (32 bits) is the cookie of the endpoint that is initiating the ISAKMP (or IKE) exchange.

- **Responder Cookie** (32 bits) is the cookie of the endpoint responding to the ISAKMP exchange. The cookies are exchanged between two endpoints, when the first ISAKMP exchange between them is performed. They are used to protect the key exchanges against denial-of-service attacks. The combination of these cookies is also used to identify the ISAKMP SA in the same way as the SPI identifies the ESP and AH SAs.

- **Next Payload** (8 bits) field indicates the type of the first payload after the ISAKMP header. The possible payload types and the corresponding values for the next payload field are listed in the following IETF quote.

Next Payload Type	Value
NONE	0
Security Association (SA)	1
Proposal (P)	2
Transform (T)	3
Key Exchange (KE)	4
Identification (ID)	5
Certificate (CERT)	6
Certificate Request (CR)	7
Hash (HASH)	8
Signature (SIG)	9
Nonce (NONCE)	10
Notification (N)	11
Delete (D)	12
Vendor ID (VID)	13
RESERVED	14 – 127
Private USE	128 – 255

IETF Quote 1: The ISAKMP Payload types.

- **Major Version (MjVer)** (4 bits) indicates the major version of the ISAKMP protocol in use (must be set to 1 with the currently standardised version).

- **Minor Version (MnVer)** (4 bits) indicates the minor version of the ISAKMP protocol in use (must be set to 0 with the currently standardised version and to 1 if some older version is used).

- **Exchange Type** (8 bits) indicates the type of the exchange being performed. The types and the corresponding values are listed in the following IETF quote.

Exchange Type	Value
NONE	0
Base	1
Identity Protection	2
Authentication Only	3
Aggressive	4
Informational	5
ISAKMP Future Use	6 – 31
DOI Specific Use	32 – 239
Private Use	240 - 255

IETF Quote 2: ISAKMP exchange types.

- **Flags** (8 bits) indicate specific options for the ISAKMP exchange.

- **Message ID** (32 bits) is a unique message identifier that is used during ISAKMP (or IKE) Phase 2 (see section 5.2) negotiations. It is randomly generated by the Phase 2 negotiation initiator and its purpose is to protect against collisions of simultaneous SA negotiations. In ISAKMP Phase 1 negotiations, this field must be set to 0.

- **Length** (32 bits) is the length of the total ISAKMP message expressed in octets (or bytes).

5.2 Oakley

As mentioned above, ISAKMP provides a generic framework for the key exchange protocols. The messages that it defines can be used to implement many different key exchange protocols. The purpose of *Oakley* is to define the specific procedures, called modes, for the key exchanges.

5.3 IKE Phases and Modes

The IKE protocol functions in two phases. These phases are actually the phases defined in the ISAKMP specification. The first phase (Phase 1) is for negotiating and establishing a secure channel between two peers for the usage of phase two IKE exchanges, thus establishing an SA between the peers. This first (Phase 1) SA is called the IKE SA or ISAKMP SA. In phase two (Phase 2) the peers can negotiate general purpose AH and ESP SAs, that is, SAs for the transmission of the IPSec payload, such as TCP, UDP or IP packets.

In the ISAKMP negotiations, obviously, the AH and ESP can not be used, thus the ISAKMP messages are conveyed on top of UDP without the use of ESP or AH. However, all the Phase 2 ISAKMP payloads as well as some of the Phase 1 payloads are encrypted.

There are three so called modes of exchanging keying information and setting up SAs in IKE. These modes are the modes of the Oakley protocol. Thus IKE is the combination of the ISAKMP and Oakley protocols as the older name stated. Two of the modes are for the Phase 1 exchange and one for the Phase 2 exchanges:

- *Main mode* is the normal way of doing phase one IKE exchange.
- *Aggressive mode* is an alternative way for phase one exchange. It is faster than the main mode, but less secure, because it does not provide identity protection for the negotiating entities.
- *Quick mode* establishes general purpose SAs in IKE phase two.

So, when two entities wish to utilise IPSec for secured communications, they must first negotiate the IKE SA in Phase 1 with either the main mode or aggressive mode, and then negotiate the general purpose SAs in Phase 2 with quick mode for the desired secured communications.

There are several ways to provide for the mutual authentication in the first phase exchange. The specification defines authentication schemes based on digital signatures, public key encryption (actually two schemes, original and revised) and a pre-shared key. The key exchange in the second phase is performed with the Diffie-Hellman algorithm.

5.4 The structures of the ISAKMP payloads relevant to SA negotiation

The structures of the ISAKMP payload types that are the most relevant to the negotiation of the Security Associations are described in this chapter to give the reader a clearer understanding of the quite fundamental concept of SA in IKE. Other payload types are not described in this document (because the

intention is not to produce a 100-page novel). Further information of the payloads can be found in [ISAKMP].

Note that the *Next Payload, Reserved* and *Payload Length* fields belong to the general ISAKMP payload header that is situated in the beginning of each payload.

In an ISAKMP Security Association proposal, the ordering of the payloads is as follows. The *SA payload* is the first one. Then follow the *Proposal payloads* and the *Transform payloads*. Each Proposal payload is associated with some transform payloads. These Transform payloads corresponding to the Proposal payload are located right after the Proposal payload. The next Proposal payload is located after the last Transform payload corresponding to the previous Proposal payload. These payload types constitute the Security Association proposal. The following figure illustrates the ordering of the payloads in an Security Association proposal. There are two Proposal payloads and three Transform payloads in the ISAKMP message illustrated by the figure. Two of the Transform payloads correspond to the first Proposal payload and one to the second Proposal payload.

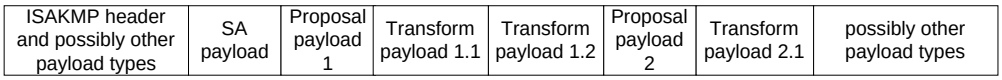

Figure 8: An ISAKMP Security Association proposal.

5.4.1 ISAKMP SA payload

The following figure illustrates the structure of the SA payload.

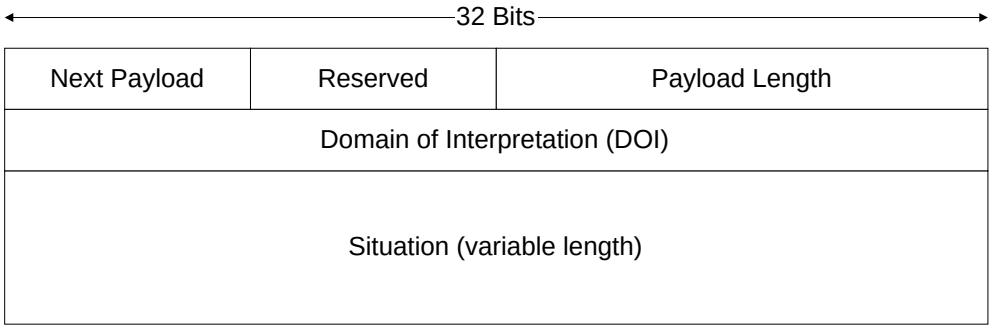

Figure 9: The ISAKMP SA payload.

The following list describes the purposes of each field in the SA payload.

- **Next Payload** (8 bits) indicates the type of the next payload of the ISAKMP message similarly to the corresponding field in the ISAKMP header. If the payload itself is the last payload in the ISAKMP message, the field must be set to 0. This field must not contain the values for the Proposal or Transform payloads as they are considered part of the security association negotiation.

- **Reserved** (8 bits) field is reserved for future use. In current implementations, it must be set to 0.

- **Payload Length** (16 bits) indicates the length in octets of the entire Security Association payload, including the SA payload itself as well as all the Proposal payloads and the Transform payloads associated with the Security Association that is being proposed.

- **Domain of Interpretation (DOI)** (32 bits) field identifies the DOI under which the negotiation is taking place. A DOI defines general information about the interpretation of ISAKMP messages.

- **Situation** (variable length) is a DOI dependent field indicating the situation under which this negotiation is taking place. The Situation is used to make policy decisions regarding the security attributes being negotiated.

5.4.2 ISAKMP Proposal payload

The Proposal Payload contains information used during Security negotiation. The proposal consists of security mechanisms, or transforms, to be used to secure the communications channel. The transforms are then described in the Transform payloads following the Proposal payload.

The following figure illustrates the structure of the Proposal payload.

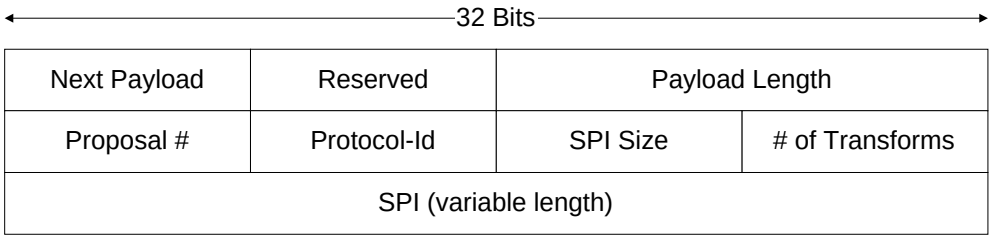

Figure 10: ISAKMP Proposal payload.

The following list describes the purposes of the fields in the Proposal payload.

- **Next Payload** (8 bits) has the same meaning as with the SA payload or the ISAKMP header. Note, however, that with Proposal payload, the only values for this field are 2, indicating another Proposal payload after the following Transform payloads and 0 indicating that this Proposal payload is the last Proposal payload within the Security Association proposal.

- **Reserved** (8 bits) is the same as with SA payload.

- **Payload Length** (16 bits) indicates the length in octets of the entire Proposal payload, including also the Transform payloads associated with this proposal.

- **Proposal #** (8 bits) identifies the proposal number for the current payload.

- **Protocol-Id** (8 bits) specifies the protocol identifier for the current negotiation. This might typically be IPSec AH or ESP, but also other, such as for example TLS.

- **SPI Size** (8 bits) indicates the length in octets of the SPI corresponding to the protocol specified by the Protocol-Id. With AH and ESP, this is thus 4.

- **# of Transforms** (8 bits) specifies the number of transforms for the Proposal. Each of these is contained in a Transform payload.

- **SPI** (variable length) indicates the sending entity's SPI for the SA being proposed.

5.4.3 ISAKMP Transform payload

The following figure illustrates the structure of the Transform payload.

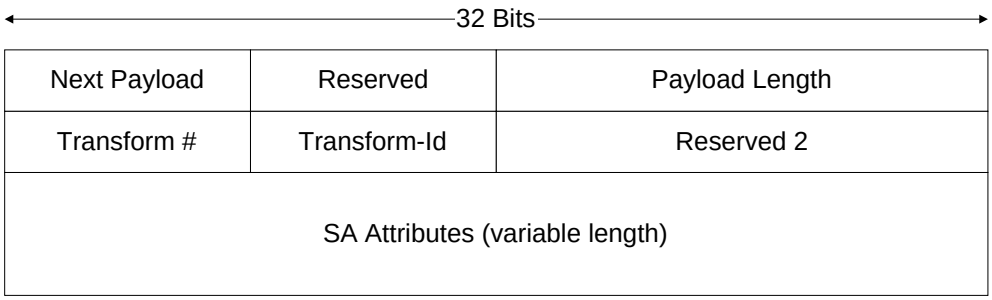

Figure 11: ISAKMP Transform payload

The following list describes the purposes of the fields in the Transform payload.

- **Next Payload** (8 bits) has the same meaning as with the SA payload or the ISAKMP header. With Transform payload, the only values for this field are 3, indicating another Transform payload after this Transform payload and 0 indicating that this Transform payload is the last within the proposal.

- **Reserved** (8 bits) is the same as with SA payload.

- **Payload Length** (16 bits) indicates the length in octets of the entire Transform payload, including only the fields of the Transform payload itself.

- **Transform #** (8 bits) identifies the Transform number for the current payload. If there is more than one transform proposed for a specific protocol within the Proposal payload, then each Transform payload has a unique Transform number.

- **Reserved 2** (16 bits) field is reserved for future use. It must be set to 0.

- **SA Attributes** (variable length) field contains the security association attributes as defined for the transform given in the Transform-Id field. The SA Attributes should be represented using the Data Attributes format specified in [ISAKMP].

The Data Attributes are either type, length, value (TLV) or type, value (TV) coded structures. The interpretation of these structures is a specified in the DOI specification for each domain. [IPDOI] is a DOI specification for Internet IP Security domain. The following IETF quote is from [IPDOI] and it specifies 9 attribute types to be used with Internet IP traffic that is secured with IPSec.

```
Attribute Types

class                    value           type
------------------------------------------------
SA Life Type                1               B
SA Life Duration            2               V
Group Description           3               B
Encapsulation Mode          4               B
Authentication Algorithm    5               B
Key Length                  6               B
Key Rounds                  7               B
Compress Dictionary Size    8               B
Compress Private Algorithm  9               V
```

IETF Quote 3: SA Attributes specified in IPDOI specification.

The type B means basic attribute and it is a constant length, or TV coded attribute. The type V means variable length attribute, or in other words TLV coded attribute.

5.5 IKE phase 1 examples

As an example, the pictures in this section illustrate the main and aggressive mode exchanges with authentication based on digital signatures. This authentication scheme is the basic Oakley authentication scheme, although Oakley does specify other schemes also, as mentioned in section 5.3. The subscripts i and r in the following figures denote the initiator and the responder of the exchanges respectively.

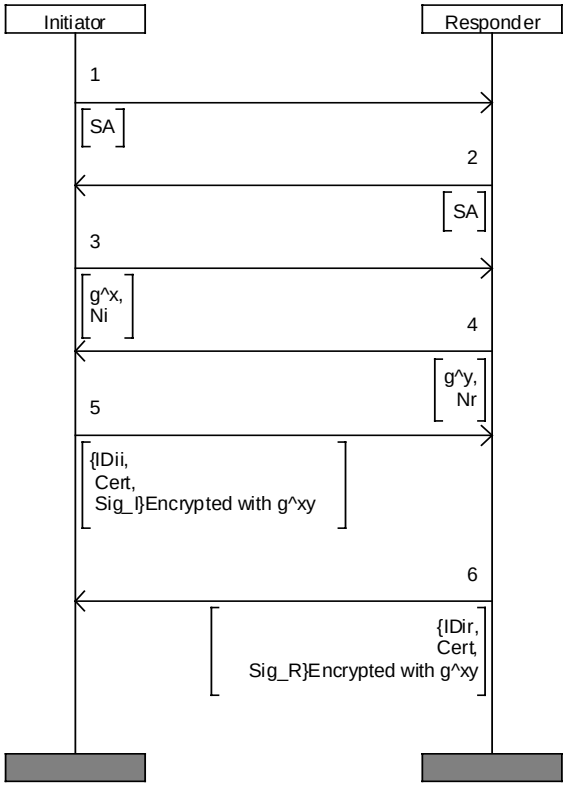

Figure 12: IKE main mode exchange with digital signatures.

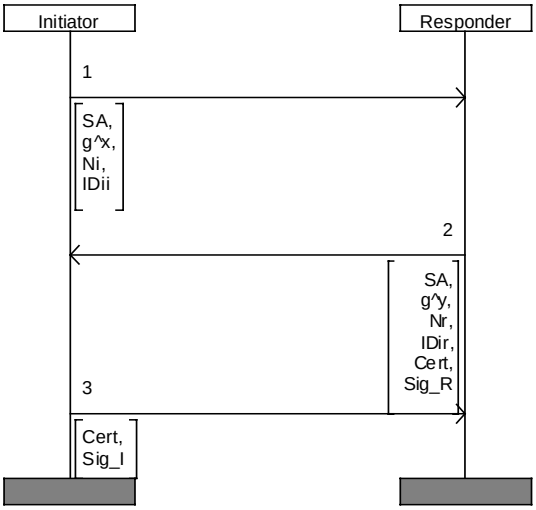

Figure 13: IKE aggressive mode exchange with digital signatures.

The SAs in message 1 of both pictures contain all the SA proposals that the initiator suggests to the responder. The SAs in message two of both pictures contain only one SA, which is the one the responder accepts. The Diffie-Hellman key exchange is performed in messages 3 and 4 in Figure 12, Ni and Nr sent in the same messages are nonce values created by both endpoints respectively. Messages 5 and 6 in the main mode perform the actual authentication. These messages are encrypted with the key g^xy obtained from the Diffie-Hellman exchange. The Certs are the certificates of each endpoint, they may be omitted, if the endpoints already have each other's certificates. IDii and IDir are the identities of the initiator and the responder (the extra i in the subscript indicates Phase 1, that is, ISAKMP SA negotiation). Sig_I and Sig_R are the signatures of the initiator and the responder respectively. They are calculated over hashes, that in turn have been calculated over most of the parameters exchanged in the process (the formula is pretty complicated and thus not presented here, it can be found in [IKE]) including the nonces exchanged in messages 3 and 4.

The aggressive mode takes the form of only three messages, but there is a price to be paid for that. Note that the identities of the endpoints are not

encrypted in messages 1 and 2 in Figure 13. Otherwise, the information exchanged with the aggressive mode is the same as with main mode.

The phase 1 exchanges with other types of authentication can also be found in [IKE].

5.6 IKE phase 2 example

The following figure illustrates the IKE quick mode exchange.

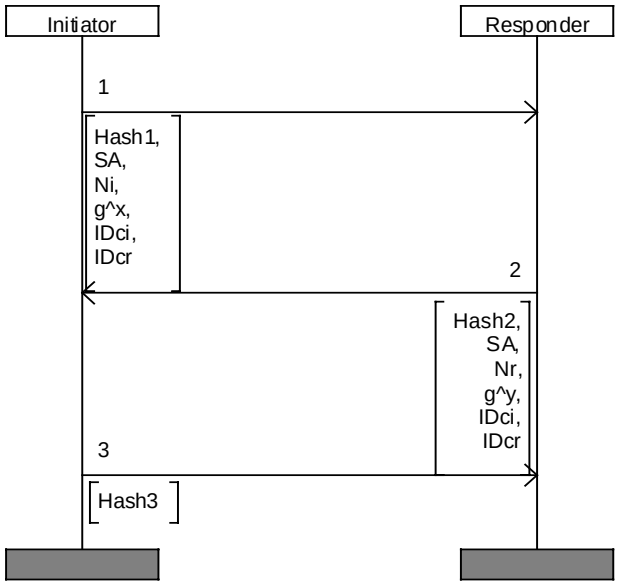

Figure 14: IKE quick mode.

All the messages are encrypted with the key that was obtained from the phase 1 exchange (so the IKE SA must be valid before quick mode can be used). Hash1 and Hash2 are calculated similarly including the message-ids (present in the header of the messages) and all the message fields. The Hash3 is calculated over the nonces Ni and Nr as well as the message id (and a zero octet). The SA in message 1 is a proposal that is accepted with the SA in message 2. Multiple SAs may also be negotiated simultaneously by including them in messages 1 and 2. The Diffie-Hellman key exchange in messages 1 and 2 is optional. If it is not used, the keying material for the SA is obtained by deriving it from material obtained in the phase 1 exchange and the nonces.

This procedure does not, however, guarantee *Perfect Forward Secrecy* (PFS) which means that new keys are not dependent on the older ones. If PFS is desired, the Diffie-Hellman exchange must be done. The identities IDci and IDcr are also optional. The subscript c in the IDs indicates that these are the identities of the endpoints whose cookies are in the ISAKMP header. If the IDs are present, they are included in the Hash1 and Hash2.

After the quick mode exchange has been done between two endpoints, they can start transmitting the IPSec payload data, such as TCP or UDP in transport mode, or IP in tunnel mode, using the rules and parameters specified in the SAs determined for that connection.

References

[IPDOI] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[ESP] Kent, S., and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[AH] Kent, S., and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[IKE] Harkins, D., and D. Carrel, D., "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[ISAKMP] Maughan, D., Schertler, M., Schneider, M., and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[Huitema] Huitema, Christian, “IPv6 The New Internet Protocol”, 2nd ed., Prentice Hall, 1998, ISBN: 0-13-850505-5