Network Working Group
Internet Draft
draft-ietf-ipsec-auth-header-02.txt

Stephen Kent, BBN Corp
Randall Atkinson, @Home Network
2 October 1997

IP Authentication Header

Status of This Memo

Table of Contents

1. Introduction....3
2. Authentication Header Format....4
2.1 Next Header....4
2.2 Payload Length....4
2.3 Reserved....5
2.4 Security Parameters Index (SPI)....5
2.5 Sequence Number....5
2.6 Authentication Data....6
3. Authentication Header Processing....6
3.1 Authentication Header Location....6
3.2 Authentication Algorithms....8
3.3 Outbound Packet Processing....9
3.3.1 Security Association Lookup....9
3.3.2 Sequence Number Generation....9
3.3.3 Integrity Check Value Calculation....10
3.3.3.1 Handling Mutable Fields....10
3.3.3.1.1 ICV Computation for IPv4....10
3.3.3.1.1.1 Base Header Fields....10
3.3.3.1.1.2 Options....11
3.3.3.1.2 ICV Computation for IPv6....12
3.3.3.1.2.1 Base Header Fields....12
3.3.3.1.2.2 Extension Headers -- Options....12
3.3.3.1.2.3 Extension Headers -- non-Options....12
3.3.3.2 Padding....12
3.3.3.2.1 Authentication Data Padding....12
3.3.3.2.2 Implicit Packet Padding....13
3.3.4 Fragmentation....13
3.4 Inbound Packet Processing....14
3.4.1 Reassembly....14
3.4.2 Security Association Lookup....14
3.4.3 Sequence Number Verification....14
3.4.4 Integrity Check Value Verification....15
4. Auditing....16
5. Conformance Requirements....16
6. Security Considerations....17
7. Differences from RFC 1826....17
Acknowledgements....18
Appendix A -- Mutability of IP Options/Extension Headers....19
A1. IPv4 Options....19
A2. IPv6 Extension Headers....20
References....22
Disclaimer....23
Author Information....23

1. Introduction

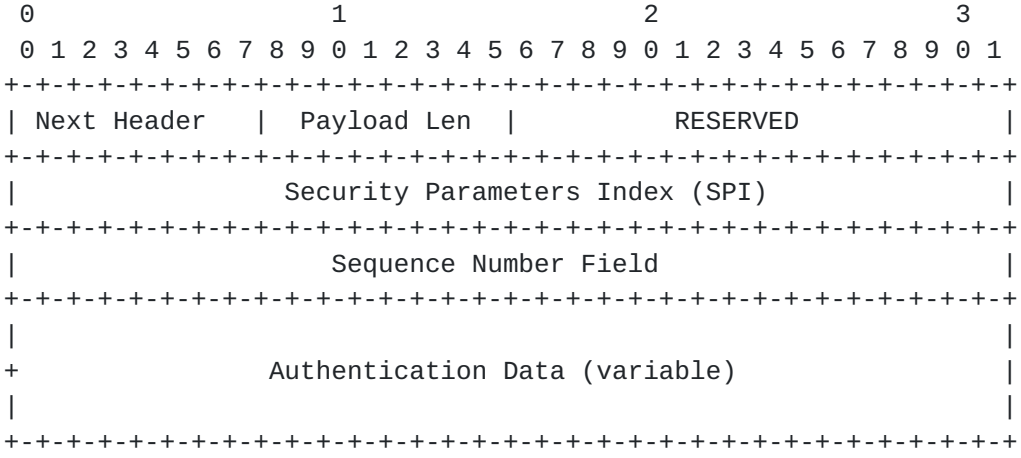
The IP Authentication Header (AH) is used to provide connectionless integrity and data origin authentication for IP datagrams (hereafter referred to as just 'authentication'), and to provide protection against replays. This latter, optional service may be selected, by the receiver, when a Security Association is established. AH provides authentication for as much of the IP header as possible, as well as for upper level protocol data. However, some IP header fields may change in transit and the value of these fields, when the packet arrives at the receiver, may not be predictable by the transmitter. The values of such fields cannot be protected by AH. Thus the protection provided to the IP header by AH is somewhat piecemeal.

AH may be applied alone, in combination with the IP Encapsulating Security Payload (ESP) [KA97b], or in a nested fashion through the use of tunnel mode (see 'Security Architecture for the Internet Protocol' [KA97a], hereafter referred to as the Security Architecture document). Security services can be provided between a pair of communicating hosts, between a pair of communicating security gateways, or between a security gateway and a host. ESP may be used to provide the same security services, and it also provides a confidentiality (encryption) service. The primary difference between the authentication provided by ESP and AH is the extent of the coverage. Specifically, ESP does not protect any IP header fields unless those fields are encapsulated by ESP (tunnel mode). For more details on how to use AH and ESP in various network environments, see the Security Architecture document [KA97a].

It is assumed that the reader is familiar with the terms and concepts described in the Security Architecture document. In particular, the reader should be familiar with the definitions of security services offered by AH and ESP, the concept of Security Associations, the ways in which AH can be used in conjunction with ESP, and the different key management options available for AH and ESP. (With regard to the last topic, the current key management options required for both AH and ESP are manual keying and automated keying via Oakley/ISAKMP.)

2. Authentication Header Format

The protocol header (IPv4, IPv6, or Extension) immediately preceding the AH header will contain the value 51 in its Protocol (IPv4) or Next Header (IPv6, Extension) field [STD-2].



The following subsections define the fields that comprise the AH format. All the fields described here are mandatory, i.e., they are always present in the AH format and are included in the ICV computation.

2.1 Next Header

The Next Header is an 8-bit field that identifies the type of the next payload after the Authentication Header. The value of this field is chosen from the set of IP Protocol Numbers defined in the most recent "Assigned Numbers" [STD-2] RFC from the Internet Assigned Numbers Authority (IANA).

2.2 Payload Length

This 8-bit field specifies the length of AH in 32-bit words (4-byte units), minus "2". (All IPv6 extension headers, as per RFC 1883, encode the "Hdr Ext Len" field by first subtracting 1 (64-bit word) from the header length (measured in 64-bit words). AH is an IPv6 extension header. However, since its length is measured in 32-bit words, the "Payload Length" is calculated by subtracting 2 (32 bit words).) In the "standard" case of a 96-bit authentication value plus the 3 32-bit word fixed portion, this length field will be "4". A "null" authentication algorithm may be used only for debugging purposes. Its use would result in a "1" value for this field, as there would be no corresponding Authentication Data field.

2.3 Reserved

This 16-bit field is reserved for future use. It MUST be set to "zero." (Note that the value is included in the Authentication Data calculation, but is otherwise ignored by the recipient.)

2.4 Security Parameters Index (SPI)

The SPI is an arbitrary 32-bit value that, in combination with the destination IP address and security protocol, uniquely identifies the Security Association for this datagram. The set of SPI values in the range 1 through 255 are reserved by the Internet Assigned Numbers Authority (IANA) for future use; a reserved SPI value will not normally be assigned by IANA unless the use of the assigned SPI value is specified in an RFC. It is ordinarily selected by the destination system upon establishment of an SA (see the Security Architecture document for more details).

The SPI value of zero (0) is reserved for local, implementation-specific use and MUST NOT be sent on the wire. For example, a key management implementation MAY use the zero SPI value to mean "No Security Association Exists" during the period when the IPsec implementation has requested that its key management entity establish a new SA, but the SA has not yet been established.

2.5 Sequence Number

This unsigned 32-bit field contains a monotonically increasing counter value (sequence number). It is mandatory and is always present even if the receiver does not elect to enable the anti-replay service for a specific SA. Processing of the Sequence Number field is at the discretion of the receiver, i.e., the sender MUST always transmit this field, but the receiver need not act upon it (see the discussion of Sequence Number Verification in the "Inbound Packet Processing" section below).

The sender's counter and the receiver's counter are initialized to 0 when an SA is established. (The first packet sent using a given SA will have a Sequence Number of 1; see Section 3.3.2 for more details on how the Sequence Number is generated.) If anti-replay has been enabled, the transmitted Sequence Number must never be allowed to cycle. Thus, the sender's counter and the receiver's counter MUST be reset (by establishing a new SA and thus a new key) prior to the transmission of the 2^32nd packet on an SA.

2.6 Authentication Data

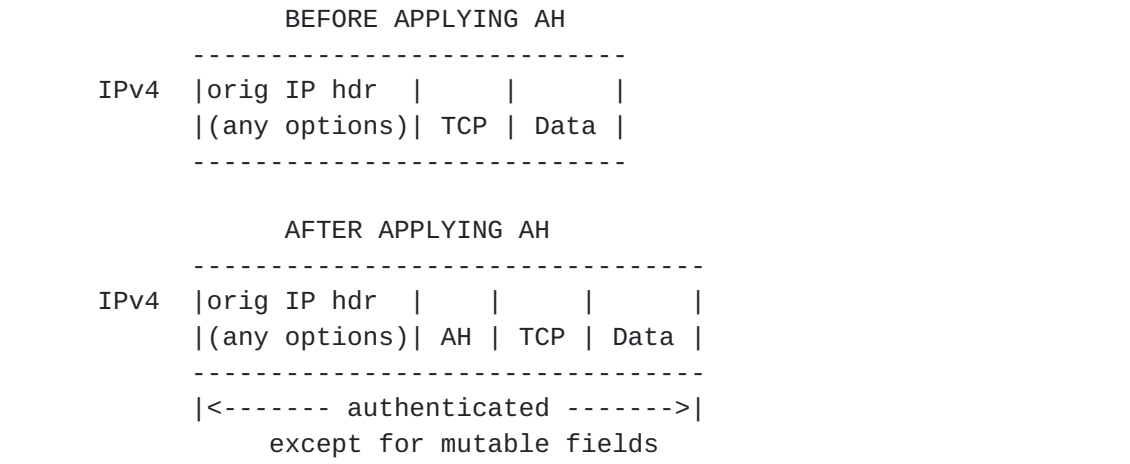
This is a variable-length field that contains the Integrity Check Value (ICV) for this packet. The field must be an integral multiple of 32 bits in length. The details of the ICV computation are described in Section 3.3.3 below. This field may include explicit padding. This padding is included to ensure that the length of the AH header is an integral multiple of 32 bits (IPv4) or 64 bits (IPv6). All implementations MUST support such padding. Details of how to compute the required padding length are provided below.

3. Authentication Header Processing

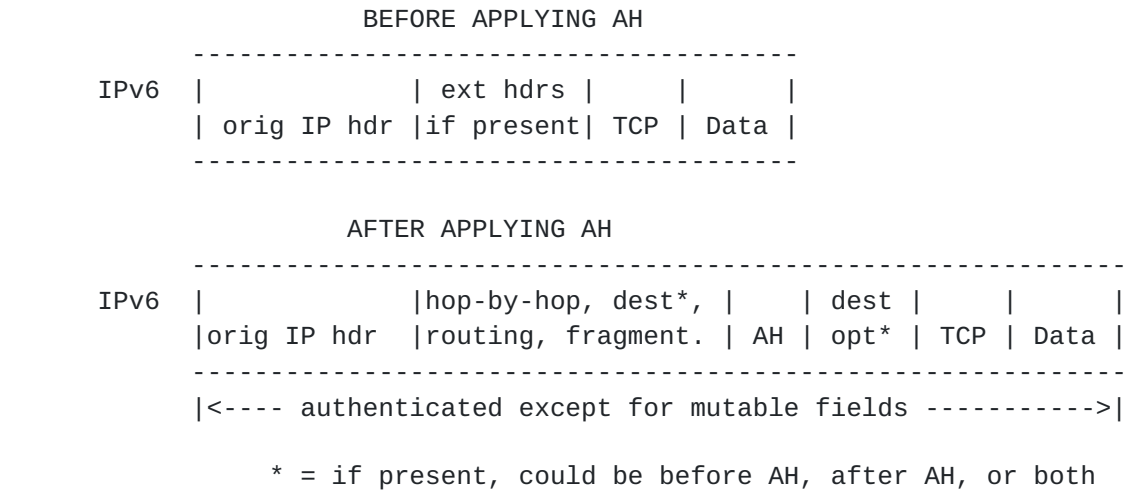
3.1 Authentication Header Location

Like ESP, AH may be employed in two ways: transport mode or tunnel mode. The former mode is applicable only to host implementations and provides protection for upper layer protocols, in addition to selected IP header fields. (In this mode, note that for "bump-in-the-stack" or "bump-in-the-wire" implementations, as defined in the Security Architecture document, inbound and outbound IP fragments may require an IPsec implementation to perform extra IP reassembly/fragmentation in order to both conform to this specification and provide transparent IPsec support. Special care is required to perform such operations within these implementations when multiple interfaces are in use.)

In transport mode, AH is inserted after the IP header and before an upper layer protocol, e.g., TCP, UDP, ICMP, etc. or before any other IPsec headers that have already been inserted. In the context of IPv4, this calls for placing AH after the IP header (and any options that it contains), but before the upper layer protocol. (Note that the term "transport" mode should not be misconstrued as restricting its use to TCP and UDP. For example, an ICMP message MAY be sent using either "transport" mode or "tunnel" mode.) The following diagram illustrates AH transport mode positioning for a typical IPv4 packet, on a "before and after" basis.



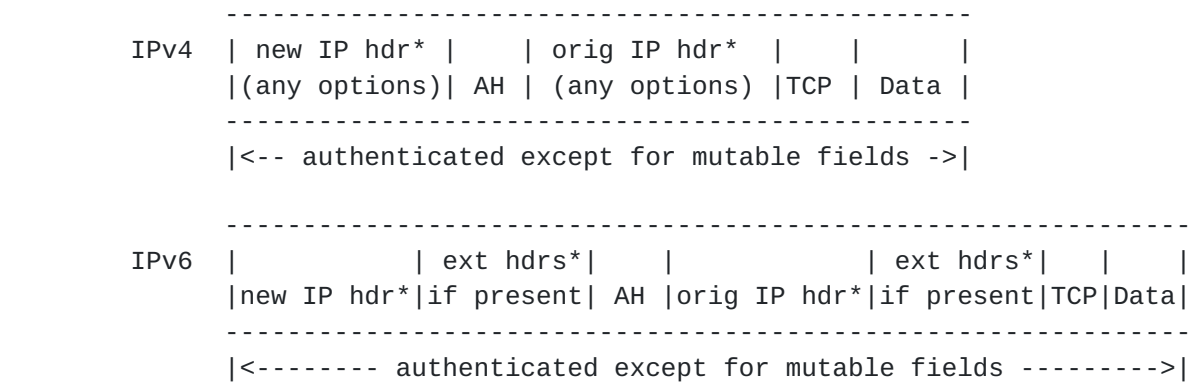
In the IPv6 context, AH is viewed as an end-to-end payload, and thus should appear after hop-by-hop, routing, and fragmentation extension headers. The destination options extension header(s) could appear either before or after the AH header depending on the semantics desired. The following diagram illustrates AH transport mode positioning for a typical IPv6 packet.



If more than one IPsec header/extension is required:

- the order of application of the security headers MUST be defined by security policy
- The following 3 cases MUST be supported:
  1. [IP][AH][upper]
  2. [IP][ESP][upper]
  3. [IP][AH][ESP][upper]
- arbitrary nesting is allowed -- Senders MAY generate arbitrary nestings of IPsec headers and Receivers SHOULD accept arbitrary nestings of IPsec headers.

Tunnel mode AH may be employed in either hosts or security gateways (or in so-called "bump-in-the-stack" or "bump-in-the-wire" implementations, as defined in the Security Architecture document). When AH is implemented in a security gateway (to protect transit traffic), tunnel mode must be used. In tunnel mode, the "inner" IP header carries the ultimate source and destination addresses, while an "outer" IP header may contain distinct IP addresses, e.g., addresses of security gateways. In tunnel mode, AH protects the entire inner IP packet, including the entire inner IP header. The position of AH in tunnel mode, relative to the outer IP header, is the same as for AH in transport mode. The following diagram illustrates AH tunnel mode positioning for typical IPv4 and IPv6 packets.



* = construction of outer IP hdr/extensions and modification of inner IP hdr/extensions is discussed below.

3.2 Authentication Algorithms

The authentication algorithm employed for the ICV computation is specified by the SA. For point-to-point communication, suitable authentication algorithms include keyed Message Authentication Codes (MACs) based on symmetric encryption algorithms (e.g., DES) or on one-way hash functions (e.g., MD5 or SHA-1). For multicast communication, one-way hash algorithms combined with asymmetric signature algorithms are appropriate, though performance and space considerations currently preclude use of such algorithms. The mandatory-to-implement authentication algorithms are described in Section 5 "Conformance Requirements". Other algorithms MAY be supported. Note: Where an algorithm yields more than 96 bits, the output of the computation is truncated to the leftmost 96 bits.

3.3 Outbound Packet Processing

In transport mode, the transmitter inserts the AH header after the IP header and before an upper layer protocol header, as described above. In tunnel mode, the outer and inner IP header/extensions can be inter-related in a variety of ways. The construction of the outer IP header/extensions during the encapsulation process is described in the Security Architecture document.

If there is more than one IPsec header/extension required, the order of the application of the security headers MUST be defined by security policy. For simplicity of processing, each IPsec header SHOULD ignore the existence (i.e., not zero the contents or try to predict the contents) of IPsec headers to be applied later. (While a native IP or bump-in-the-stack implementation could predict the contents of later IPsec headers that it applies itself, it won't be possible for it to predict any IPsec headers added by a bump-in-the-wire implementation between the host and the network.)

3.3.1 Security Association Lookup

AH is applied to an outbound packet only after an IPsec implementation determines that the packet is associated with an SA that calls for AH processing. The process of determining what, if any, IPsec processing is applied to outbound traffic is described in the Security Architecture document.

3.3.2 Sequence Number Generation

The sender's counter is initialized to 0 when an SA is established. The transmitter increments the Sequence Number for this SA and inserts the new value into the Sequence Number Field. Thus the first packet sent using a given SA will have a Sequence Number of 1.

If anti-replay has been enabled, the transmitter checks to ensure that the counter has not cycled before inserting the new value in the Sequence Number field. In other words, the transmitter MUST not send a packet on an SA if doing so would cause the Sequence Number to cycle. An attempt to transmit a packet that would result in Sequence Number overflow is an auditable event. (Note that this approach to Sequence Number management does not require use of modular arithmetic.)

If anti-replay has not been enabled, the sender does not need to monitor or reset the counter, e.g., in the case of manual key management. NOTE: If the receiver does NOT notify the sender that anti-replay is enabled, then the sender may overflow the counter and

may send packets that the receiver will reject.

3.3.3 Integrity Check Value Calculation

The AH ICV is computed over:

- o IP header fields that are either immutable in transit or that are predictable in value upon arrival at the endpoint for the AH SA
- o the AH header (Next Header, Payload Len, Reserved, SPI, Sequence Number, and the Authentication Data (which is set to zero for this computation))
- o the upper level protocol data, which is assumed to be immutable in transit

3.3.3.1 Handling Mutable Fields

If a field may be modified during transit, the value of the field is set to zero for purposes of the ICV computation. If a field is mutable, but its value at the (IPsec) receiver is predictable, then that value is inserted into the field for purposes of the ICV calculation. The Authentication Data field is also set to zero in preparation for this computation. Note that by replacing each field's value with zero, rather than omitting the field, alignment is preserved for the ICV calculation. Also, the zero-fill approach ensures that the length of the fields that are so handled cannot be changed during transit, even though their contents are not explicitly covered by the ICV.

As a new extension header or IPv4 option is created, it will be defined in its own RFC and SHOULD include (in the Security Considerations section) directions for how it should be handled when calculating the AH ICV. If the IPSEC implementation encounters an extension header that it does not recognize, it MUST zero the whole header except for the Next Header and Hdr Ext Len fields. The length of the extension header MUST be computed by 8 * Hdr Ext Len value + 8. If the IPSEC implementation encounters an IPv4 option that it does not recognize, it should zero the whole option, using the second byte of the option as the length. (IPv6 options contain a flag indicating mutability, which determines appropriate processing for such options.)

3.3.3.1.1 ICV Computation for IPv4

3.3.3.1.1.1 Base Header Fields

The IPv4 base header fields are classified as follows:

Immutable
Version
Internet Header Length
Total Length
Identification
Protocol
Source Address
Destination Address (without loose or strict source routing)

Mutable but predictable
Destination Address (with loose or strict source routing)

Mutable (zeroed prior to ICV calculation)
Type of Service (TOS)
Flags
Fragment Offset
Time to Live (TTL)
Header Checksum

TOS -- This field is excluded because some routers are known to change the value of this field, even though the IP specification does not consider TOS to be a mutable header field.

Flags -- This field is excluded since an intermediate router might set the DF bit, even if the source did not select it.

Fragment Offset -- Since AH is applied only to non-fragmented IP packets, the Offset Field must always be zero, and thus it is excluded (even though it is predictable).

TTL -- This is changed en-route as a normal course of processing by routers, and thus its value at the receiver is not predictable by the sender.

Header Checksum -- This will change if any of these other fields changes, and thus its value upon reception cannot be predicted by the sender.

3.3.3.1.1.2 Options

For IPv4 (unlike IPv6), there is no mechanism for tagging options as mutable in transit. Hence the IPv4 options are explicitly listed in Appendix A and classified as immutable, mutable but predictable, or mutable. For IPv4, the entire option is viewed as a unit; so even though the type and length fields within most options are immutable in transit, if an option is classified as mutable, the entire option is zeroed for ICV computation purposes.

3.3.3.1.2 ICV Computation for IPv6

3.3.3.1.2.1 Base Header Fields

The IPv6 base header fields are classified as follows:

Immutable
- Version
- Payload Length
- Next Header
- Source Address
- Destination Address (without Routing Extension Header)

Mutable but predictable
- Destination Address (with Routing Extension Header)

Mutable (zeroed prior to ICV calculation)
- Priority
- Flow Label
- Hop Limit

3.3.3.1.2.2 Extension Headers -- Options

The IPv6 extension headers (that are options) are explicitly listed in Appendix A and classified as immutable, mutable but predictable, or mutable.

IPv6 options in the Hop-by-Hop and Destination Extension Headers contain a bit that indicates whether the option might change (unpredictably) during transit. For any option for which contents may change en-route, the entire "Option Data" field must be treated as zero-valued octets when computing or verifying the ICV. The Option Type and Opt Data Len are included in the ICV calculation. All options for which the bit indicates immutability are included in the ICV calculation. See the IPv6 specification [DH95] for more information.

3.3.3.1.2.3 Extension Headers -- non-Options

The IPv6 extension headers (that are not options) are explicitly listed in Appendix A and classified as immutable, mutable but predictable, or mutable.

3.3.3.2 Padding

3.3.3.2.1 Authentication Data Padding

As mentioned in section 2.6, the Authentication Data field explicitly

includes padding to ensure that the AH header is a multiple of 32 bits (IPv4) or 64 bits (IPv6). If padding is required, its length is determined by two factors:

- the length of the ICV
- the IP protocol version (v4 or v6)

For example, if the output of the selected algorithm is 96-bits, no padding is required for either IPv4 or for IPv6. However, if a different length ICV is generated, due to use of a different algorithm, then padding may be required for the IPv6 environment. The content of the padding field is arbitrarily selected by the sender. (The padding is arbitrary, but need not be random to achieve security.) These padding bytes are included in the Authentication Data calculation, counted as part of the Payload Length, and transmitted at the end of the Authentication Data field to enable the receiver to perform the ICV calculation.

3.3.3.2.2 Implicit Packet Padding

For some authentication algorithms, the byte string over which the ICV computation is performed must be a multiple of a blocksize specified by the algorithm. If the IP packet length (including AH) does not match the blocksize requirements for the algorithm, implicit padding MUST be appended to the end of the packet, prior to ICV computation. The padding octets MUST have a value of zero. The blocksize (and hence the length of the padding) is specified by the algorithm specification. This padding is not transmitted with the packet.

3.3.4 Fragmentation

If required, IP fragmentation occurs after AH processing within an IPsec implementation. Thus, transport mode AH is applied only to whole IP datagrams (not to IP fragments). An IP packet to which AH has been applied may itself be fragmented by routers en route, and such fragments must be reassembled prior to AH processing at a receiver. In tunnel mode, AH is applied to an IP packet, the payload of which may be a fragmented IP packet. For example, a security gateway or a "bump-in-the-stack" or "bump-in-the-wire" IPsec implementation (see the Security Architecture document for details) may apply tunnel mode AH to such fragments.

3.4 Inbound Packet Processing

If there is more than one IPsec header/extension present, the processing for each one ignores (does not zero, does not use) any IPsec headers applied subsequent to the header being processed.

3.4.1 Reassembly

If required, reassembly is performed prior to AH processing. If a packet offered to AH for processing appears to be an IP fragment, i.e., the OFFSET field is non-zero or the MORE FRAGMENTS flag is set, the receiver MUST discard the packet; this is an auditable event. The audit log entry for this event SHOULD include the SPI value, date/time, Source Address, Destination Address, and (in IPv6) the Flow ID.

3.4.2 Security Association Lookup

Upon receipt of a packet containing an IP Authentication Header, the receiver determines the appropriate (unidirectional) SA, based on the destination IP address, security protocol (AH),and the SPI. (This process is described in more detail in the Security Architecture document.) The SA indicates whether the Sequence Number field will be checked, specifies the algorithm(s) employed for ICV computation, and indicates the key(s) required to validate the ICV.

If no valid Security Association exists for this session (e.g., the receiver has no key), the receiver MUST discard the packet; this is an auditable event. The audit log entry for this event SHOULD include the SPI value, date/time, Source Address, Destination Address, and (in IPv6) the Flow ID.

3.4.3 Sequence Number Verification

All AH implementations MUST support the anti-replay service, though its use may be enabled or disabled on a per-SA basis. (Note that there are no provisions for managing transmitted Sequence Number values among multiple senders directing traffic to a single, multicast SA. Thus the anti-replay service SHOULD NOT be used in a multi-sender multicast environment that employs a single, multicast SA.)

If the receiver does not enable anti-replay for an SA, no checks are performed on the inbound Sequence Number. If an SA establishment protocol such as Oakley/ISAKMP is employed, then the receiver SHOULD notify the transmitter, during SA establishment, if the receiver will provide anti-replay protection.

If the receiver has enabled the anti-replay service for this SA, the receiver packet counter for the SA MUST be initialized to zero when the SA is established. For each received packet, the receiver MUST verify that the packet contains a Sequence Number that does not duplicate the Sequence Number of any other packets received during the life of this SA. This SHOULD be the first AH check applied to a packet after it has been matched to an SA, to speed rejection of duplicate packets.

Duplicates are rejected through the use of a sliding receive window. (How the window is implemented is a local matter, but the following text describes the functionality that the implementation must exhibit.) A MINIMUM window size of 32 MUST be supported; but a window size of 64 is preferred and SHOULD be employed as the default. Another window size (larger than the MINIMUM) MAY be chosen by the receiver. (The receiver does NOT notify the sender of the window size.)

The "right" edge of the window represents the highest, validated Sequence Number value received on this SA. Packets that contain Sequence Numbers lower than the "left" edge of the window are rejected. Packets falling within the window are checked against a list of received packets within the window. An efficient means for performing this check, based on the use of a bit mask, is described in the Security Architecture document.

If the received packet falls within the window and is new, or if the packet is to the right of the window, then the receiver proceeds to ICV verification. If the ICV validation fails, the receiver MUST discard the received IP datagram as invalid; this is an auditable event. The audit log entry for this event SHOULD include the SPI value, date/time, Source Address, Destination Address, the Sequence Number, and (in IPv6) the Flow ID. The receive window is updated only if the ICV verification succeeds.

DISCUSSION:

> Note that if the packet is either inside the window and new, or is outside the window on the "right" side, the receiver MUST authenticate the packet before updating the Sequence Number window data.

3.4.4 Integrity Check Value Verification

The receiver computes the ICV over the appropriate fields of the packet, using the specified authentication algorithm, and verifies that it is the same as the ICV included in the Authentication Data field of the packet. Details of the computation are provided below.

If the computed and received ICV's match, then the datagram is valid, and it is accepted. If the test fails, then the receiver MUST discard the received IP datagram as invalid; this is an auditable event. The audit log entry SHOULD include the SPI value, date/time received, Source Address, Destination Address, and (in IPv6) the Flow ID.

DISCUSSION:

Begin by saving the ICV value and replacing it (but not any Authentication Data padding) with zero. Zero all other fields that may have been modified during transit. (See section 3.3.3.1 for a discussion of which fields are zeroed before performing the ICV calculation.) Check the overall length of the packet, and if it requires implicit padding based on the requirements of the authentication algorithm, append zero-filled bytes to the end of the packet as required. Now perform the ICV computation and compare the result with the saved value. Note that if the output of the authentication algorithm is greater than 96 bits, the output should be truncated to the leftmost 96 bits. (If a digital signature and one-way hash are used for the ICV computation, the matching process is more complex and will be described in the algorithm specification.)

4. Auditing

Not all systems that implement AH will implement auditing. However, if AH is incorporated into a system that supports auditing, then the AH implementation MUST also support auditing and MUST allow a system administrator to enable or disable auditing for AH. For the most part, the granularity of auditing is a local matter. However, several auditable events are identified in this specification and for each of these events a minimum set of information that SHOULD be included in an audit log is defined. Additional information also MAY be included in the audit log for each of these events, and additional events, not explicitly called out in this specification, also MAY result in audit log entries. There is no requirement for the receiver to transmit any message to the purported transmitter in response to the detection of an auditable event, because of the potential to induce denial of service via such action.

5. Conformance Requirements

Implementations that claim conformance or compliance with this specification MUST fully implement the AH syntax and processing described here and MUST comply with all requirements of the Security Architecture document. If the key used to compute an ICV is manually

distributed, correct provision of the anti-replay service would require correct maintenance of the counter state at the transmitter, until the key is replaced, and there likely would be no automated recovery provision if counter overflow were imminent. Thus a compliant implementation SHOULD NOT provide this service in conjunction with SAs that are manually keyed. A compliant AH implementation MUST support the following mandatory-to-implement algorithms:

- HMAC with MD5 [MG97a]
- HMAC with SHA-1 [MG97b]

6. Security Considerations

Security is central to the design of this protocol, and these security considerations permeate the specification. Additional security-relevant aspects of using the IPsec protocol are discussed in the Security Architecture document.

7. Differences from RFC 1826

This specification of AH differs from RFC 1826 [ATK95] in several important respects, but the fundamental features of AH remain intact. One goal of the revision of RFC 1826 was to provide a complete framework for AH, with ancillary RFCs required only for algorithm specification. For example, the anti-replay service is now an integral, mandatory part of AH, not a feature of a transform defined in another RFC. Carriage of a sequence number to support this service is now required at all times. The default algorithms required for interoperability have been changed to HMAC with MD5 or SHA-1 (vs. keyed MD5), for security reasons. The list of IPv4 header fields excluded from the ICV computation has been expanded to include the OFFSET and FLAGS fields.

Another motivation for revision was to provide additional detail and clarification of subtle points. This specification provides rationale for exclusion of selected IPv4 header fields from AH coverage and provides examples on positioning of AH in both the IPv4 and v6 contexts. Auditing requirements have been clarified in this version of the specification. Tunnel mode AH was mentioned only in passing in RFC 1826, but now is a mandatory feature of AH. Discussion of interactions with key management and with security labels have been moved to the Security Architecture document.

Acknowledgements

For over 2 years, this document has evolved through multiple versions and iterations. During this time, many people have contributed significant ideas and energy to the process and the documents themselves. The authors would like to thank Karen Seo for providing extensive help in the review, editing, background research, and coordination for this version of the specification. The authors would also like to thank the members of the IPsec and IPng working groups, with special mention of the efforts of (in alphabetic order): Steve Bellovin, Steve Deering, Francis Dupont, Phil Karn, Frank Kastenholz, Perry Metzger, David Mihelcic, Hilarie Orman, Norman Shulman, William Simpson, and Nina Yuan.

Appendix A -- Mutability of IP Options/Extension Headers

A1. IPv4 Options

This table shows how the IPv4 options are classified with regard to "mutability". Where two references are provided, the second one supercedes the first. This table is based in part on information provided in RFC1700, "ASSIGNED NUMBERS", (October 1994).

Copy	Class	Opt. #	Name	Reference
IMMUTABLE -- included in ICV calculation				
0	0	0	End of Options List	[RFC791]
0	0	1	No Operation	[RFC791]
1	0	2	Security	[RFC1108(historic but in use)]
1	0	5	Extended Security	[RFC1108(historic but in use)]
1	0	6	Commercial Security	[expired I-D, now US MIL STD]
1	0	20	Router Alert	[RFC2113]
1	0	21	Sender Directed Multi-Destination Delivery	[RFC1770]
MUTABLE -- zeroed				
1	0	3	Loose Source Route	[RFC791]
0	2	4	Time Stamp	[RFC791]
0	0	7	Record Route	[RFC791]
1	0	9	Strict Source Route	[RFC791]
0	2	18	Traceroute	[RFC1393]
EXPERIMENTAL, SUPERCEDED -- zeroed				
1	0	8	Stream ID	[RFC791, RFC1122 (Host Req)]
0	0	11	MTU Probe	[RFC1063, RFC1191 (PMTU)]
0	0	12	MTU Reply	[RFC1063, RFC1191 (PMTU)]
1	0	17	Extended Internet Protocol	[RFC1385, RFC1883 (IPv6)]
0	0	10	Experimental Measurement	[ZSu]
1	2	13	Experimental Flow Control	[Finn]
1	0	14	Experimental Access Ctl	[Estrin]
0	0	15	???	[VerSteeg]
1	0	16	IMI Traffic Descriptor	[Lee]
1	0	19	Address Extension	[Ullmann IPv7]

NOTE: Use of the Router Alert option is potentially incompatible with use of IPSEC. Although the option is immutable, its use implies that each router along a packet's path will "process" the packet and consequently might change the packet. This would happen on a hop by hop basis as the packet goes from router to router. Prior to being processed by the application to which the option contents are directed, e.g., RSVP/IGMP, the packet should encounter AH processing.

However, AH processing would require that each router along the path is a member of a multicast-SA defined by the SPI. This might pose problems for packets that are not strictly source routed, and it requires multicast support techniques not currently available.

NOTE: Addition or removal of any security labels (BSO, ESO, CIPSO) by systems along a packet's path conflicts with the classification of these IP Options as immutable and is incompatible with the use of IPSEC.

A2. IPv6 Extension Headers

This table shows how the IPv6 Extension Headers are classified with regard to "mutability".

Option/Extension Name	Reference
MUTABLE BUT PREDICTABLE -- included in ICV calculation	
Routing (Type 0)	[RFC1883]
BIT INDICATES IF OPTION IS MUTABLE (CHANGES UNPREDICTABLY DURING TRANSIT)	
Hop by Hop options	[RFC1883]
Destination options	[RFC1883]
NOT APPLICABLE	
Fragmentation	[RFC1883]

Options -- IPv6 options in the Hop-by-Hop and Destination Extension Headers contain a bit that indicates whether the option might change (unpredictably) during transit. For any option for which contents may change en-route, the entire "Option Data" field must be treated as zero-valued octets when computing or verifying the ICV. The Option Type and Opt Data Len are included in the ICV calculation. All options for which the bit indicates immutability are included in the ICV calculation. See the IPv6 specification [DH95] for more information.

Routing (Type 0) -- The IPv6 Routing Header "Type 0" will rearrange the address fields within the packet during transit from source to destination. However, the contents of the packet as it will appear at the receiver are known to the sender and to all intermediate hops. Hence, the IPv6 Routing Header "Type 0" is included in the Authentication Data calculation as mutable but predictable. The transmitter must order the field so that it appears as it will at the receiver, prior to performing the ICV computation.

Fragmentation -- Fragmentation occurs after outbound IPSEC processing

(section 3.3) and reassembly occurs before inbound IPSEC processing (section 3.4). So the Fragmentation Extension Header, if it exists, is not seen by IPSEC.

Note that on the receive side, the IP implementation could leave a Fragmentation Extension Header in place when it does re-assembly. If this happens, then when AH receives the packet, before doing ICV processing, AH MUST "remove" (or skip over) this header and change the previous header's "Next Header" field to be the "Next Header" field in the Fragmentation Extension Header.

Note that on the send side, the IP implementation could give the IPSEC code a packet with a Fragmentation Extension Header with Offset of 0 (first fragment) and a More Fragments Flag of 0 (last fragment). If this happens, then before doing ICV processing, AH MUST first "remove" (or skip over) this header and change the previous header's "Next Header" field to be the "Next Header" field in the Fragmentation Extension Header.

References

[ATK95] R. Atkinson, "The IP Authentication Header," RFC 1826, August 1995.

[BCCH94] R. Braden, D. Clark, S. Crocker, & C.Huitema, "Report of IAB Workshop on Security in the Internet Architecture", RFC-1636, 9 June 1994, pp. 21-34.

[Bel89] Steven M. Bellovin, "Security Problems in the TCP/IP Protocol Suite", ACM Computer Communications Review, Vol. 19, No. 2, March 1989.

[CER95] Computer Emergency Response Team (CERT), "IP Spoofing Attacks and Hijacked Terminal Connections", CA-95:01, January 1995. Available via anonymous ftp from info.cert.org in /pub/cert_advisories.

[DH95] Steve Deering & Bob Hinden, "Internet Protocol version 6 (IPv6) Specification", RFC-1883, December 1995.

[GM93] James Galvin & Keith McCloghrie, Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2), RFC-1446, April 1993.

[KA97a] Steve Kent, Randall Atkinson, "Security Architecture for the Internet Protocol", Internet Draft, ?? 1997.

[KA97b] Steve Kent, Randall Atkinson, "IP Encapsulating Security Payload (ESP)", Internet Draft, ?? 1997.

[KA97c] Steve Kent, Randall Atkinson, "IP Authentication Header", Internet Draft, ?? 1997.

[Ken91] Steve Kent, "US DoD Security Options for the Internet Protocol", RFC-1108, November 1991.

[MG97a] C. Madson & R. Glenn, "The Use of HMAC-MD5-96 within ESP and AH", Internet Draft, 7/2/97.

[MG97b] C. Madson & R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", Internet Draft, 7/2/97.

[Riv92] Ronald Rivest, "The MD5 Message Digest Algorithm," RFC-1321, April 1992.

[SHA] NIST, FIPS PUB 180-1: Secure Hash Standard, April 1995

[STD-1] J. Postel, "Internet Official Protocol Standards", STD-1, March 1996.

[STD-2] J. Reynolds & J. Postel, "Assigned Numbers", STD-2, 20 October 1994.

Disclaimer

The views and specification here are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this specification.

Author Information

Stephen Kent
BBN Corporation
70 Fawcett Street
Cambridge, MA 02140
USA
E-mail: kent@bbn.com
Telephone: +1 (617) 873-3988

Randall Atkinson
@Home Network
425 Broadway,
Redwood City, CA 94063
USA
E-mail: rja@inet.org