IP Authentication Header

Status of this Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

ABSTRACT

   This document describes a mechanism for providing cryptographic
   authentication for IPv4 and IPv6 datagrams. An Authentication Header
   (AH) is normally inserted after an IP header and before the other
   information being authenticated.

1. INTRODUCTION

   The Authentication Header is a mechanism for providing strong
   integrity and authentication for IP datagrams. It might also provide
   non-repudiation, depending on which cryptographic algorithm is used
   and how keying is performed. For example, use of an asymmetric
   digital signature algorithm, such as RSA, could provide non-
   repudiation.

   Confidentiality, and protection from traffic analysis are not
   provided by the Authentication Header. Users desiring
   confidentiality should consider using the IP Encapsulating Security
   Protocol (ESP) either in lieu of or in conjunction with the
   Authentication Header [Atk95b]. This document assumes the reader has
   previously read the related IP Security Architecture document which
   defines the overall security architecture for IP and provides
   important background information for this specification [Atk95a].

1.1 Overview

   The IP Authentication Header seeks to provide security by adding
   authentication information to an IP datagram. This authentication
   information is calculated using all of the fields in the IP datagram
   (including not only the IP Header but also other headers and the user
   data) which do not change in transit. Fields or options which need
   to change in transit (e.g., "hop count", "time to live", "ident",

"fragment offset", or "routing pointer") are considered to be zero
for the calculation of the authentication data. This provides
significantly more security than is currently present in IPv4 and
might be sufficient for the needs of many users.

Use of this specification will increase the IP protocol processing
costs in participating end systems and will also increase the
communications latency. The increased latency is primarily due to
the calculation of the authentication data by the sender and the
calculation and comparison of the authentication data by the receiver
for each IP datagram containing an Authentication Header. The impact
will vary with authentication algorithm used and other factors.

In order for the Authentication Header to work properly without
changing the entire Internet infrastructure, the authentication data
is carried in its own payload. Systems that aren't participating in
the authentication MAY ignore the Authentication Data. When used
with IPv6, the Authentication Header is normally placed after the
Fragmentation and End-to-End headers and before the ESP and
transport-layer headers. The information in the other IP headers is
used to route the datagram from origin to destination. When used
with IPv4, the Authentication Header immediately follows an IPv4
header.

If a symmetric authentication algorithm is used and intermediate
authentication is desired, then the nodes performing such
intermediate authentication would need to be provided with the
appropriate keys. Possession of those keys would permit any one of
those systems to forge traffic claiming to be from the legitimate
sender to the legitimate receiver or to modify the contents of
otherwise legitimate traffic. In some environments such intermediate
authentication might be desirable [BCCH94]. If an asymmetric
authentication algorithm is used and the routers are aware of the
appropriate public keys and authentication algorithm, then the
routers possessing the authentication public key could authenticate
the traffic being handled without being able to forge or modify
otherwise legitimate traffic. Also, Path MTU Discovery MUST be used
when intermediate authentication of the Authentication Header is
desired and IPv4 is in use because with this method it is not
possible to authenticate a fragment of a packet [MD90] [Kno93].

1.2 Requirements Terminology

    In this document, the words that are used to define the significance
    of each particular requirement are usually capitalised. These words
    are:

    - MUST

      This word or the adjective "REQUIRED" means that the item is an
      absolute requirement of the specification.

    - SHOULD

      This word or the adjective "RECOMMENDED" means that there might
      exist valid reasons in particular circumstances to ignore this
      item, but the full implications should be understood and the case
      carefully weighed before taking a different course.

    - MAY

      This word or the adjective "OPTIONAL" means that this item is
      truly optional. One vendor might choose to include the item
      because a particular marketplace requires it or because it
      enhances the product, for example; another vendor may omit the
      same item.

2. KEY MANAGEMENT

    Key management is an important part of the IP security architecture.
    However, it is not integrated with this specification because of a
    long history in the public literature of subtle flaws in key
    management algorithms and protocols. The IP Authentication Header
    tries to decouple the key management mechanisms from the security
    protocol mechanisms. The only coupling between the key management
    protocol and the security protocol is with the Security Parameters
    Index (SPI), which is described in more detail below. This
    decoupling permits several different key management mechanisms to be
    used. More importantly, it permits the key management protocol to be
    changed or corrected without unduly impacting the security protocol
    implementations.

    The key management mechanism is used to negotiate a number of
    parameters for each "Security Association", including not only the
    keys but also other information (e.g., the authentication algorithm
    and mode) used by the communicating parties. The key management
    mechanism creates and maintains a logical table containing the
    several parameters for each current security association. An
    implementation of the IP Authentication Header will need to read that

logical table of security parameters to determine how to process each
datagram containing an Authentication Header (e.g., to determine
which algorithm/mode and key to use in authentication).

Security Associations are unidirectional. A bidirectional
communications session will normally have one Security Association in
each direction. For example, when a TCP session exists between two
systems A and B, there will normally be one Security Association from
A to B and a separate second Security Assocation from B to A. The
receiver assigns the SPI value to the the Security Association with
that sender. The other parameters of the Security Association are
determined in a manner specified by the key management mechanism.
Section 4 of this document describes in detail the process of
selecting a Security Association for an outgoing packet and
identifying the Security Assocation for an incoming packet.

The IP Security Architecture document describes key management in
detail. It includes specification of the key management requirements
for this protocol, and is incorporated here by reference [Atk95a].

3. AUTHENTICATION HEADER SYNTAX

The Authentication Header (AH) may appear after any other headers
which are examined at each hop, and before any other headers which
are not examined at an intermediate hop. The IPv4 or IPv6 header
immediately preceding the Authentication Header will contain the
value 51 in its Next Header (or Protocol) field [STD-2].

Example high-level diagrams of IP datagrams with the Authentication
Header follow.

```
+------------+------------------+------------+-------+---------------+
| IPv6 Header| Hop-by-Hop/Routing| Auth Header| Others| Upper Protocol|
+------------+------------------+------------+-------+---------------+
```
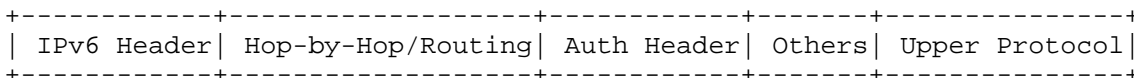
Figure 1: IPv6 Example

When used with IPv6, the Authentication Header normally appears after
the IPv6 Hop-by-Hop Header and before the IPv6 Destination Options.

```
+-------------+-------------+-----------------------------+
| IPv4 Header | Auth Header | Upper Protocol (e.g. TCP, UDP)|
+-------------+-------------+-----------------------------+
```
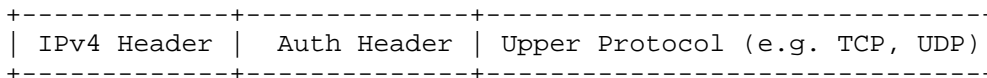
              Figure 2: IPv4 Example

When used with IPv4, the Authentication Header normally follows the
main IPv4 header.

3.1 Authentication Header Syntax

The authentication data is the output of the authentication algorithm
calculated over the the entire IP datagram as described in more
detail later in this document. The authentication calculation must
treat the Authentication Data field itself and all fields that are
normally modified in transit (e.g., TTL or Hop Limit) as if those
fields contained all zeros. All other Authentication Header fields
are included in the authentication calculation normally.

The IP Authentication Header has the following syntax:

```
+--------------+--------------+--------------+--------------+
| Next Header  | Length       |          RESERVED           |
+--------------+--------------+--------------+--------------+
|                 Security Parameters Index                 |
+--------------+--------------+--------------+--------------+
|                                                          |
+     Authentication Data (variable number of 32-bit words) |
|                                                          |
+--------------+--------------+--------------+--------------+
 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
```
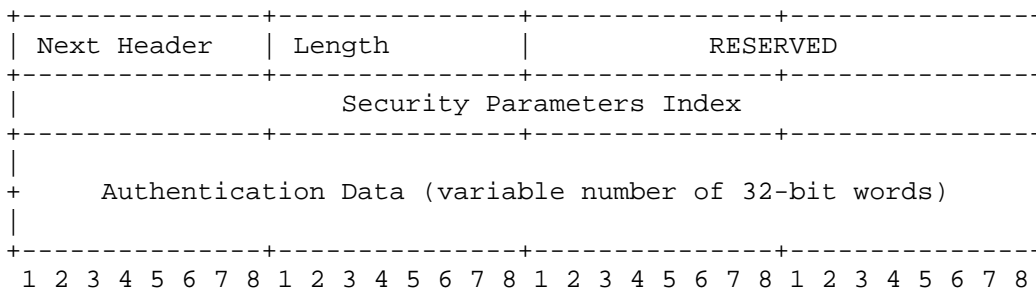

            Figure 3: Authentication Header syntax

3.2 Fields of the Authentication Header

    NEXT HEADER
        8 bits wide. Identifies the next payload after the Authentication
        Payload. This values in this field are the set of IP Protocol
        Numbers as defined in the most recent RFC from the Internet
        Assigned Numbers Authority (IANA) describing "Assigned Numbers"
        [STD-2].

    PAYLOAD LENGTH
        8 bits wide. The length of the Authentication Data field in 32-
        bit words. Minimum value is 0 words, which is only used in the
        degenerate case of a "null" authentication algorithm.

    RESERVED
        16 bits wide. Reserved for future use. MUST be set to all zeros
        when sent. The value is included in the Authentication Data
        calculation, but is otherwise ignored by the recipient.

    SECURITY PARAMETERS INDEX (SPI)
        A 32-bit pseudo-random value identifying the security association
        for this datagram. The Security Parameters Index value 0 is
        reserved to indicate that "no security association exists".

        The set of Security Parameters Index values in the range 1 through
        255 are reserved to the Internet Assigned Numbers Authority (IANA)
        for future use. A reserved SPI value will not normally be
        assigned by IANA unless the use of that particular assigned SPI
        value is openly specified in an RFC.

    AUTHENTICATION DATA
        This length of this field is variable, but is always an integral
        number of 32-bit words.

        Many implementations require padding to other alignments, such as
        64-bits, in order to improve performance. All implementations
        MUST support such padding, which is specified by the Destination
        on a per SPI basis. The value of the padding field is arbitrarily
        selected by the sender and is included in the Authentication Data
        calculation.

        An implementation will normally use the combination of Destination
        Address and SPI to locate the Security Association which specifies
        the field's size and use. The field retains the same format for
        all datagrams of any given SPI and Destination Address pair.

The Authentication Data fills the field beginning immediately
after the SPI field. If the field is longer than necessary to
store the actual authentication data, then the unused bit
positions are filled with unspecified, implementation-dependent
values.

Refer to each Authentication Transform specification for more
information regarding the contents of this field.

3.3 Sensitivity Labeling

As is discussed in greater detail in the IP Security Architecture
document, IPv6 will normally use implicit Security Labels rather than
the explicit labels that are currently used with IPv4 [Ken91]
[Atk95a]. In some situations, users MAY choose to carry explicit
labels (for example, IPSO labels as defined by RFC-1108 might be used
with IPv4) in addition to using the implicit labels provided by the
Authentication Header. Explicit label options could be defined for
use with IPv6 (e.g., using the IPv6 end-to-end options header or the
IPv6 hop-by-hop options header). Implementations MAY support
explicit labels in addition to implicit labels, but implementations
are not required to support explicit labels. If explicit labels are
in use, then the explicit label MUST be included in the
authentication calculation.

4. CALCULATION OF THE AUTHENTICATION DATA

The authentication data carried by the IP Authentication Header is
usually calculated using a message digest algorithm (for example,
MD5) either encrypting that message digest or keying the message
digest directly [Riv92]. Only algorithms that are believed to be
cryptographically strong one-way functions should be used with the IP
Authentication Header.

Because conventional checksums (e.g., CRC-16) are not
cryptographically strong, they MUST NOT be used with the
Authentication Header.

When processing an outgoing IP packet for Authentication, the first
step is for the sending system to locate the appropriate Security
Association. All Security Associations are unidirectional. The
selection of the appropriate Security Association for an outgoing IP
packet is based at least upon the sending userid and the Destination
Address. When host-oriented keying is in use, all sending userids
will share the same Security Association to a given destination.
When user-oriented keying is in use, then different users or possibly
even different applications of the same user might use different
Security Associations. The Security Association selected will

indicate which algorithm, algorithm mode, key, and other security
properties apply to the outgoing packet.

Fields which NECESSARILY are modified during transit from the sender
to the receiver (e.g., TTL and HEADER CHECKSUM for IPv4 or Hop Limit
for IPv6) and whose value at the receiver are not known with
certainty by the sender are included in the authentication data
calculation but are processed specially. For these fields which are
modified during transit, the value carried in the IP packet is
replaced by the value zero for the purpose of the authentication
calculation. By replacing the field's value with zero rather than
omitting these fields, alignment is preserved for the authentication
calculation.

The sender MUST compute the authentication over the packet as that
packet will appear at the receiver. This requirement is placed in
order to allow for future IP optional headers which the receiver
might not know about but the sender necessarily knows about if it is
including such options in the packet. This also permits the
authentication of data that will vary in transit but whose value at
the final receiver is known with certainty by the sender in advance.

The sender places the calculated message digest algorithm output into
the Authentication Data field within the Authentication Header. For
purposes of Authentication Data computation, the Authentication Data
field is considered to be filled with zeros.

The IPv4 "TIME TO LIVE" and "HEADER CHECKSUM" fields are the only
fields in the IPv4 base header that are handled specially for the
Authentication Data calculation. Reassembly of fragmented packets
occurs PRIOR to processing by the local IP Authentication Header
implementation. The "more" bit is of course cleared upon reassembly.
 Hence, no other fields in the IPv4 header will vary in transit from
the perspective of the IP Authentication Header implementation. The
"TIME TO LIVE" and "HEADER CHECKSUM" fields of the IPv4 base header
MUST be set to all zeros for the Authentication Data calculation.
All other IPv4 base header fields are processed normally with their
actual contents. Because IPv4 packets are subject to intermediate
fragmentation in routers, it is important that the reassembly of IPv4
packets be performed prior to the Authentication Header processing.
IPv4 Implementations SHOULD use Path MTU Discovery when the IP
Authentication Header is being used [MD90]. For IPv4, not all
options are openly specified in a RFC, so it is not possible to
enumerate in this document all of the options that might normally be
modified during transit. The IP Security Option (IPSO) MUST be
included in the Authentication Data calculation whenever that option
is present in an IP datagram [Ken91]. If a receiving system does not
recognise an IPv4 option that is present in the packet, that option

is included in the Authentication Data calculation. This means that
any IPv4 packet containing an IPv4 option that changes during transit
in a manner not predictable by the sender and which IPv4 option is
unrecognised by the receiver will fail the authentication check and
consequently be dropped by the receiver.

The IPv6 "HOP LIMIT" field is the only field in the IPv6 base header
that is handled specially for Authentication Data calculation. The
value of the HOP LIMIT field is zero for the purpose of
Authentication Data calculation. All other fields in the base IPv6
header MUST be included in the Authentication Data calculation using
the normal procedures for calculating the Authentication Data. All
IPv6 "OPTION TYPE" values contain a bit which MUST be used to
determine whether that option data will be included in the
Authentication Data calculation. This bit is the third-highest-order
bit of the IPv6 OPTION TYPE field. If this bit is set to zero, then
the corresponding option is included in the Authentication Data
calculation. If this bit is set to one, then the corresponding
option is replaced by all zero bits of the same length as the option
for the purpose of the Authentication Data calculation. The IPv6
Routing Header "Type 0" will rearrange the address fields within the
packet during transit from source to destination. However, this is
not a problem because 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 using the normal procedure.

Upon receipt of a packet containing an IP Authentication Header, the
receiver first uses the Destination Address and SPI value to locate
the correct Security Association. The receiver then independently
verifies that the Authentication Data field and the received data
packet are consistent. Again, the Authentication Data field is
assumed to be zero for the sole purpose of making the authentication
computation. Exactly how this is accomplished is algorithm
dependent. If the processing of the authentication algorithm
indicates the datagram is valid, then it is accepted. If the
algorithm determines that the data and the Authentication Header do
not match, then the receiver SHOULD discard the received IP datagram
as invalid and MUST record the authentication failure in the system
log or audit log. If such a failure occurs, the recorded log data
MUST include the SPI value, date/time received, clear-text Sending
Address, clear-text Destination Address, and (if it exists) the
clear-text Flow ID. The log data MAY also include other information
about the failed packet.

5. CONFORMANCE REQUIREMENTS

   Implementations that claim conformance or compliance with this
   specification MUST fully implement the header described here, MUST
   support manual key distribution for use with this option, MUST comply
   with all requirements of the "Security Architecture for the Internet
   Protocol" [Atk95a], and MUST support the use of keyed MD5 as
   described in the companion document entitled "IP Authentication using
   Keyed MD5" [MS95]. Implementations MAY also implement other
   authentication algorithms. Implementors should consult the most
   recent version of the "IAB Official Standards" RFC for further
   guidance on the status of this document.

6. SECURITY CONSIDERATIONS

   This entire RFC discusses an authentication mechanism for IP. This
   mechanism is not a panacea to the several security issues in any
   internetwork, however it does provide a component useful in building
   a secure internetwork.

   Users need to understand that the quality of the security provided by
   this specification depends completely on the strength of whichever
   cryptographic algorithm has been implemented, the strength of the key
   being used, the correctness of that algorithm's implementation, upon
   the security of the key management mechanism and its implementation,
   and upon the correctness of the IP Authentication Header and IP
   implementations in all of the participating systems. If any of these
   assumptions do not hold, then little or no real security will be
   provided to the user. Implementors are encouraged to use high
   assurance methods to develop all of the security relevant parts of
   their products.

   Users interested in confidentiality should consider using the IP
   Encapsulating Security Payload (ESP) instead of or in conjunction
   with this specification [Atk95b]. Users seeking protection from
   traffic analysis might consider the use of appropriate link
   encryption. Description and specification of link encryption is
   outside the scope of this note [VK83]. Users interested in combining
   the IP Authentication Header with the IP Encapsulating Security
   Payload should consult the IP Encapsulating Security Payload
   specification for details.

   One particular issue is that in some cases a packet which causes an
   error to be reported back via ICMP might be so large as not to
   entirely fit within the ICMP message returned. In such cases, it
   might not be possible for the receiver of the ICMP message to
   independently authenticate the portion of the returned message. This
   could mean that the host receiving such an ICMP message would either

trust an unauthenticated ICMP message, which might in turn create
some security problem, or not trust and hence not react appropriately
to some legitimate ICMP message that should have been reacted to. It
is not clear that this issue can be fully resolved in the presence of
packets that are the same size as or larger than the minimum IP MTU.
Similar complications arise if an encrypted packet causes an ICMP
error message to be sent and that packet is truncated.

Active attacks are now widely known to exist in the Internet [CER95].
The presence of active attacks means that unauthenticated source
routing, either unidirectional (receive-only) or with replies
following the original received source route represents a significant
security risk unless all received source routed packets are
authenticated using the IP Authentication Header or some other
cryptologic mechanism. It is noteworthy that the attacks described
in [CER95] include a subset of those described in [Bel89].

The use of IP tunneling with AH creates multiple pairs of endpoints
that might perform AH processing. Implementers and administrators
should carefully consider the impacts of tunneling on authenticity of
the received tunneled packets.

ACKNOWLEDGEMENTS

This document benefited greatly from work done by Bill Simpson, Perry
Metzger, and Phil Karn to make general the approach originally
defined by the author for SIP, SIPP, and finally IPv6.

The basic concept here is derived in large part from the SNMPv2
Security Protocol work described in [GM93]. Steve Bellovin, Steve
Deering, Frank Kastenholz, Dave Mihelcic, and Hilarie Orman provided
thoughtful critiques of early versions of this note. Francis Dupont
discovered and pointed out the security issue with ICMP in low IP MTU
links that is noted just above.

REFERENCES

    [Atk95a] Atkinson, R., "Security Architecture for the Internet
             Protocol", RFC 1825, NRL, August 1995.

    [Atk95b] Atkinson, R., "IP Encapsulating Security Payload", RFC 1827,
             NRL, August 1995.

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

   [BCCH94] Braden, R., Clark, D., Crocker, S., and C. Huitema, "Report
            of IAB Workshop on Security in the Internet Architecture",
            RFC 1636, USC/Information Sciences Institute, MIT, Trusted
            Information Systems, INRIA, June 1994, pp. 21-34.

   [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.

   [GM93]   Galvin J., and K. McCloghrie, "Security Protocols for
            version 2 of the Simple Network Management Protocol
            (SNMPv2)", RFC 1446, Trusted Information Systems, Hughes LAN
            Systems, April 1993.

   [Hin94] Bob Hinden (Editor), Internet Protocol version 6 (IPv6)
            Specification, Work in Progress, October 1994.

   [Ken91] Kent, S., "US DoD Security Options for the Internet Protocol",
            RFC 1108, BBN Communications, November 1991.

   [Kno93] Knowles, Stev, "IESG Advice from Experience with Path MTU
            Discovery", RFC 1435, FTP Software, March 1993.

   [MS95]   Metzger, P., and W. Simpson, "IP Authentication with Keyed
            MD5", RFC 1828, Piermont, Daydreamer, August 1995.

   [MD90]   Mogul, J., and S. Deering, "Path MTU Discovery", RFC 1191,
            DECWRL, Stanford University, November 1990.

   [STD-2] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2,
            RFC 1700, USC/Information Sciences Institute, October 1994.

   [Riv92] Rivest, R., "MD5 Digest Algorithm", RFC 1321, MIT and RSA Data
            Security, Inc., April 1992.

   [VK83]   V.L. Voydock & S.T. Kent, "Security Mechanisms in High-level
            Networks", ACM Computing Surveys, Vol. 15, No. 2, June 1983.

DISCLAIMER

   The views and specification here are those of the author and are not
   necessarily those of his employer. The Naval Research Laboratory has
   not passed judgement on the merits, if any, of this work. The author
   and his employer specifically disclaim responsibility for any
   problems arising from correct or incorrect implementation or use of
   this specification.

AUTHOR INFORMATION

   Randall Atkinson
   Information Technology Division
   Naval Research Laboratory
   Washington, DC 20375-5320
   USA

   Phone: (202) 767-2389
   Fax: (202) 404-8590
   EMail: atkinson@itd.nrl.navy.mil