Network Working Group
Request for Comments: 2407
Category: Standards Track

D. Piper
Network Alchemy

November 1998

The Internet IP Security Domain of Interpretation for ISAKMP

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.

Copyright Notice

IESG Note

Section 4.4.4.2 states, "All implememtations within the IPSEC DOI MUST support ESP_DES...". Recent work in the area of cryptanalysis suggests that DES may not be sufficiently strong for many applications. Therefore, it is very likely that the IETF will deprecate the use of ESP_DES as a mandatory cipher suite in the near future. It will remain as an optional use protocol. Although the IPsec working group and the IETF in general have not settled on an alternative algorithm (taking into account concerns of security and performance), implementers may want to heed the recommendations of section 4.4.4.3 on the use of ESP_3DES.

1. Abstract

The Internet Security Association and Key Management Protocol (ISAKMP) defines a framework for security association management and cryptographic key establishment for the Internet. This framework consists of defined exchanges, payloads, and processing guidelines that occur within a given Domain of Interpretation (DOI). This document defines the Internet IP Security DOI (IPSEC DOI), which instantiates ISAKMP for use with IP when IP uses ISAKMP to negotiate security associations.

For a list of changes since the previous version of the IPSEC DOI, please see Section 7.

2. Introduction

Within ISAKMP, a Domain of Interpretation is used to group related protocols using ISAKMP to negotiate security associations. Security protocols sharing a DOI choose security protocol and cryptographic transforms from a common namespace and share key exchange protocol identifiers. They also share a common interpretation of DOI-specific payload data content, including the Security Association and Identification payloads.

Overall, ISAKMP places the following requirements on a DOI definition:

- o define the naming scheme for DOI-specific protocol identifiers
- o define the interpretation for the Situation field
- o define the set of applicable security policies
- o define the syntax for DOI-specific SA Attributes (Phase II)
- o define the syntax for DOI-specific payload contents
- o define additional Key Exchange types, if needed
- o define additional Notification Message types, if needed

The remainder of this document details the instantiation of these requirements for using the IP Security (IPSEC) protocols to provide authentication, integrity, and/or confidentiality for IP packets sent between cooperating host systems and/or firewalls.

For a description of the overall IPSEC architecture, see [ARCH], [AH], and [ESP].

3. Terms and Definitions

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC 2119].

4.1 IPSEC Naming Scheme

Within ISAKMP, all DOI’s must be registered with the IANA in the "Assigned Numbers" RFC [STD-2]. The IANA Assigned Number for the Internet IP Security DOI (IPSEC DOI) is one (1). Within the IPSEC DOI, all well-known identifiers MUST be registered with the IANA under the IPSEC DOI. Unless otherwise noted, all tables within this document refer to IANA Assigned Numbers for the IPSEC DOI. See Section 6 for further information relating to the IANA registry for the IPSEC DOI.

All multi-octet binary values are stored in network byte order.

4.2 IPSEC Situation Definition

Within ISAKMP, the Situation provides information that can be used by the responder to make a policy determination about how to process the incoming Security Association request. For the IPSEC DOI, the Situation field is a four (4) octet bitmask with the following values.

Situation	Value
SIT_IDENTITY_ONLY	0x01
SIT_SECRECY	0x02
SIT_INTEGRITY	0x04

4.2.1 SIT_IDENTITY_ONLY

The SIT_IDENTITY_ONLY type specifies that the security association will be identified by source identity information present in an associated Identification Payload. See Section 4.6.2 for a complete description of the various Identification types. All IPSEC DOI implementations MUST support SIT_IDENTITY_ONLY by including an Identification Payload in at least one of the Phase I Oakley exchanges ([IKE], Section 5) and MUST abort any association setup that does not include an Identification Payload.

If an initiator supports neither SIT_SECRECY nor SIT_INTEGRITY, the situation consists only of the 4 octet situation bitmap and does not include the Labeled Domain Identifier field (Figure 1, Section 4.6.1) or any subsequent label information. Conversely, if the initiator supports either SIT_SECRECY or SIT_INTEGRITY, the Labeled Domain Identifier MUST be included in the situation payload.

4.2.2 SIT_SECRECY

The SIT_SECRECY type specifies that the security association is being negotiated in an environment that requires labeled secrecy. If SIT_SECRECY is present in the Situation bitmap, the Situation field will be followed by variable-length data that includes a sensitivity level and compartment bitmask. See Section 4.6.1 for a complete description of the Security Association Payload format.

If an initiator does not support SIT_SECRECY, SIT_SECRECY MUST NOT be set in the Situation bitmap and no secrecy level or category bitmaps shall be included.

If a responder does not support SIT_SECRECY, a SITUATION-NOT-SUPPORTED Notification Payload SHOULD be returned and the security association setup MUST be aborted.

4.2.3 SIT_INTEGRITY

The SIT_INTEGRITY type specifies that the security association is being negotiated in an environment that requires labeled integrity. If SIT_INTEGRITY is present in the Situation bitmap, the Situation field will be followed by variable-length data that includes an integrity level and compartment bitmask. If SIT_SECRECY is also in use for the association, the integrity information immediately follows the variable-length secrecy level and categories. See section 4.6.1 for a complete description of the Security Association Payload format.

If an initiator does not support SIT_INTEGRITY, SIT_INTEGRITY MUST NOT be set in the Situation bitmap and no integrity level or category bitmaps shall be included.

If a responder does not support SIT_INTEGRITY, a SITUATION-NOT-SUPPORTED Notification Payload SHOULD be returned and the security association setup MUST be aborted.

4.3 IPSEC Security Policy Requirements

The IPSEC DOI does not impose specific security policy requirements on any implementation. Host system policy issues are outside of the scope of this document.

However, the following sections touch on some of the issues that must be considered when designing an IPSEC DOI host implementation. This section should be considered only informational in nature.

4.3.1 Key Management Issues

It is expected that many systems choosing to implement ISAKMP will strive to provide a protected domain of execution for a combined IKE key management daemon. On protected-mode multiuser operating systems, this key management daemon will likely exist as a separate privileged process.

In such an environment, a formalized API to introduce keying material into the TCP/IP kernel may be desirable. The IP Security architecture does not place any requirements for structure or flow between a host TCP/IP kernel and its key management provider.

4.3.2 Static Keying Issues

Host systems that implement static keys, either for use directly by IPSEC, or for authentication purposes (see [IKE] Section 5.4), should take steps to protect the static keying material when it is not residing in a protected memory domain or actively in use by the TCP/IP kernel.

For example, on a laptop, one might choose to store the static keys in a configuration store that is, itself, encrypted under a private password.

Depending on the operating system and utility software installed, it may not be possible to protect the static keys once they've been loaded into the TCP/IP kernel, however they should not be trivially recoverable on initial system startup without having to satisfy some additional form of authentication.

4.3.3 Host Policy Issues

It is not realistic to assume that the transition to IPSEC will occur overnight. Host systems must be prepared to implement flexible policy lists that describe which systems they desire to speak securely with and which systems they require speak securely to them. Some notion of proxy firewall addresses may also be required.

A minimal approach is probably a static list of IP addresses, network masks, and a security required flag or flags.

A more flexible implementation might consist of a list of wildcard DNS names (e.g. '*.foo.bar'), an in/out bitmask, and an optional firewall address. The wildcard DNS name would be used to match incoming or outgoing IP addresses, the in/out bitmask would be used to determine whether or not security was to be applied and in which direction, and the optional firewall address would be used to indicate whether or not tunnel mode would be needed to talk to the target system though an intermediate firewall.

4.3.4 Certificate Management

Host systems implementing a certificate-based authentication scheme will need a mechanism for obtaining and managing a database of certificates.

Secure DNS is to be one certificate distribution mechanism, however the pervasive availability of secure DNS zones, in the short term, is doubtful for many reasons. What's far more likely is that hosts will need an ability to import certificates that they acquire through secure, out-of-band mechanisms, as well as an ability to export their own certificates for use by other systems.

However, manual certificate management should not be done so as to preclude the ability to introduce dynamic certificate discovery mechanisms and/or protocols as they become available.

4.4 IPSEC Assigned Numbers

The following sections list the Assigned Numbers for the IPSEC DOI: Situation Identifiers, Protocol Identifiers, Transform Identifiers, AH, ESP, and IPCOMP Transform Identifiers, Security Association Attribute Type Values, Labeled Domain Identifiers, ID Payload Type Values, and Notify Message Type Values.

4.4.1 IPSEC Security Protocol Identifier

The ISAKMP proposal syntax was specifically designed to allow for the simultaneous negotiation of multiple Phase II security protocol suites within a single negotiation. As a result, the protocol suites listed below form the set of protocols that can be negotiated at the same time. It is a host policy decision as to what protocol suites might be negotiated together.

The following table lists the values for the Security Protocol Identifiers referenced in an ISAKMP Proposal Payload for the IPSEC DOI.

Protocol ID	Value
RESERVED	0
PROTO_ISAKMP	1
PROTO_IPSEC_AH	2
PROTO_IPSEC_ESP	3
PROTO_IPCOMP	4

4.4.1.1 PROTO_ISAKMP

The PROTO_ISAKMP type specifies message protection required during Phase I of the ISAKMP protocol. The specific protection mechanism used for the IPSEC DOI is described in [IKE]. All implementations within the IPSEC DOI MUST support PROTO_ISAKMP.

NB: ISAKMP reserves the value one (1) across all DOI definitions.

4.4.1.2 PROTO_IPSEC_AH

The PROTO_IPSEC_AH type specifies IP packet authentication. The default AH transform provides data origin authentication, integrity protection, and replay detection. For export control considerations, confidentiality MUST NOT be provided by any PROTO_IPSEC_AH transform.

4.4.1.3 PROTO_IPSEC_ESP

The PROTO_IPSEC_ESP type specifies IP packet confidentiality. Authentication, if required, must be provided as part of the ESP transform. The default ESP transform includes data origin authentication, integrity protection, replay detection, and confidentiality.

4.4.1.4 PROTO_IPCOMP

The PROTO_IPCOMP type specifies IP payload compression as defined in [IPCOMP].

4.4.2 IPSEC ISAKMP Transform Identifiers

As part of an ISAKMP Phase I negotiation, the initiator's choice of Key Exchange offerings is made using some host system policy description. The actual selection of Key Exchange mechanism is made using the standard ISAKMP Proposal Payload. The following table lists the defined ISAKMP Phase I Transform Identifiers for the Proposal Payload for the IPSEC DOI.

Transform	Value
RESERVED	0
KEY_IKE	1

Within the ISAKMP and IPSEC DOI framework it is possible to define key establishment protocols other than IKE (Oakley). Previous versions of this document defined types both for manual keying and for schemes based on use of a generic Key Distribution Center (KDC). These identifiers have been removed from the current document.

The IPSEC DOI can still be extended later to include values for additional non-Oakley key establishment protocols for ISAKMP and IPSEC, such as Kerberos [RFC-1510] or the Group Key Management Protocol (GKMP) [RFC-2093].

4.4.2.1 KEY_IKE

The KEY_IKE type specifies the hybrid ISAKMP/Oakley Diffie-Hellman key exchange (IKE) as defined in the [IKE] document. All implementations within the IPSEC DOI MUST support KEY_IKE.

4.4.3 IPSEC AH Transform Identifiers

The Authentication Header Protocol (AH) defines one mandatory and several optional transforms used to provide authentication, integrity, and replay detection. The following table lists the defined AH Transform Identifiers for the ISAKMP Proposal Payload for the IPSEC DOI.

Note: the Authentication Algorithm attribute MUST be specified to identify the appropriate AH protection suite. For example, AH_MD5 can best be thought of as a generic AH transform using MD5. To request the HMAC construction with AH, one specifies the AH_MD5 transform ID along with the Authentication Algorithm attribute set to HMAC-MD5. This is shown using the "Auth(HMAC-MD5)" notation in the following sections.

Transform ID	Value
RESERVED	0-1
AH_MD5	2
AH_SHA	3
AH_DES	4

Note: all mandatory-to-implement algorithms are listed as "MUST" implement (e.g. AH_MD5) in the following sections. All other algorithms are optional and MAY be implemented in any particular implementation.

4.4.3.1 AH_MD5

The AH_MD5 type specifies a generic AH transform using MD5. The actual protection suite is determined in concert with an associated SA attribute list. A generic MD5 transform is currently undefined.

All implementations within the IPSEC DOI MUST support AH_MD5 along with the Auth(HMAC-MD5) attribute. This suite is defined as the HMAC-MD5-96 transform described in [HMACMD5].

The AH_MD5 type along with the Auth(KPDK) attribute specifies the AH transform (Key/Pad/Data/Key) described in RFC-1826.

Use of AH_MD5 with any other Authentication Algorithm attribute value is currently undefined.

4.4.3.2 AH_SHA

The AH_SHA type specifies a generic AH transform using SHA-1. The actual protection suite is determined in concert with an associated SA attribute list. A generic SHA transform is currently undefined.

All implementations within the IPSEC DOI MUST support AH_SHA along with the Auth(HMAC-SHA) attribute. This suite is defined as the HMAC-SHA-1-96 transform described in [HMACSHA].

Use of AH_SHA with any other Authentication Algorithm attribute value is currently undefined.

4.4.3.3 AH_DES

The AH_DES type specifies a generic AH transform using DES. The actual protection suite is determined in concert with an associated SA attribute list. A generic DES transform is currently undefined.

The IPSEC DOI defines AH_DES along with the Auth(DES-MAC) attribute to be a DES-MAC transform. Implementations are not required to support this mode.

Use of AH_DES with any other Authentication Algorithm attribute value is currently undefined.

4.4.4 IPSEC ESP Transform Identifiers

The Encapsulating Security Payload (ESP) defines one mandatory and many optional transforms used to provide data confidentiality. The following table lists the defined ESP Transform Identifiers for the ISAKMP Proposal Payload for the IPSEC DOI.

Note: when authentication, integrity protection, and replay detection are required, the Authentication Algorithm attribute MUST be specified to identify the appropriate ESP protection suite. For example, to request HMAC-MD5 authentication with 3DES, one specifies the ESP_3DES transform ID with the Authentication Algorithm attribute set to HMAC-MD5. For additional processing requirements, see Section 4.5 (Authentication Algorithm).

Transform ID	Value
RESERVED	0
ESP_DES_IV64	1
ESP_DES	2
ESP_3DES	3
ESP_RC5	4
ESP_IDEA	5
ESP_CAST	6
ESP_BLOWFISH	7
ESP_3IDEA	8
ESP_DES_IV32	9
ESP_RC4	10
ESP_NULL	11

Note: all mandatory-to-implement algorithms are listed as "MUST" implement (e.g. ESP_DES) in the following sections. All other algorithms are optional and MAY be implemented in any particular implementation.

4.4.4.1 ESP_DES_IV64

The ESP_DES_IV64 type specifies the DES-CBC transform defined in RFC-1827 and RFC-1829 using a 64-bit IV.

4.4.4.2 ESP_DES

The ESP_DES type specifies a generic DES transform using DES-CBC. The actual protection suite is determined in concert with an associated SA attribute list. A generic transform is currently undefined.

All implementations within the IPSEC DOI MUST support ESP_DES along with the Auth(HMAC-MD5) attribute. This suite is defined as the [DES] transform, with authentication and integrity provided by HMAC MD5 [HMACMD5].

4.4.4.3 ESP_3DES

The ESP_3DES type specifies a generic triple-DES transform. The actual protection suite is determined in concert with an associated SA attribute list. The generic transform is currently undefined.

All implementations within the IPSEC DOI are strongly encouraged to support ESP_3DES along with the Auth(HMAC-MD5) attribute. This suite is defined as the [ESPCBC] transform, with authentication and integrity provided by HMAC MD5 [HMACMD5].

4.4.4.4 ESP_RC5

The ESP_RC5 type specifies the RC5 transform defined in [ESPCBC].

4.4.4.5 ESP_IDEA

The ESP_IDEA type specifies the IDEA transform defined in [ESPCBC].

4.4.4.6 ESP_CAST

The ESP_CAST type specifies the CAST transform defined in [ESPCBC].

4.4.4.7 ESP_BLOWFISH

The ESP_BLOWFISH type specifies the BLOWFISH transform defined in [ESPCBC].

4.4.4.8 ESP_3IDEA

The ESP_3IDEA type is reserved for triple-IDEA.

4.4.4.9 ESP_DES_IV32

The ESP_DES_IV32 type specifies the DES-CBC transform defined in RFC-1827 and RFC-1829 using a 32-bit IV.

4.4.4.10 ESP_RC4

The ESP_RC4 type is reserved for RC4.

4.4.4.11 ESP_NULL

The ESP_NULL type specifies no confidentiality is to be provided by ESP. ESP_NULL is used when ESP is being used to tunnel packets which require only authentication, integrity protection, and replay detection.

All implementations within the IPSEC DOI MUST support ESP_NULL. The ESP NULL transform is defined in [ESPNULL]. See the Authentication Algorithm attribute description in Section 4.5 for additional requirements relating to the use of ESP_NULL.

4.4.5 IPSEC IPCOMP Transform Identifiers

The IP Compression (IPCOMP) transforms define optional compression algorithms that can be negotiated to provide for IP payload compression ([IPCOMP]). The following table lists the defined IPCOMP Transform Identifiers for the ISAKMP Proposal Payload within the

IPSEC DOI.

Transform ID	Value
RESERVED	0
IPCOMP_OUI	1
IPCOMP_DEFLATE	2
IPCOMP_LZS	3

4.4.5.1 IPCOMP_OUI

The IPCOMP_OUI type specifies a proprietary compression transform. The IPCOMP_OUI type must be accompanied by an attribute which further identifies the specific vendor algorithm.

4.4.5.2 IPCOMP_DEFLATE

The IPCOMP_DEFLATE type specifies the use of the "zlib" deflate algorithm as specified in [DEFLATE].

4.4.5.3 IPCOMP_LZS

The IPCOMP_LZS type specifies the use of the Stac Electronics LZS algorithm as specified in [LZS].

4.5 IPSEC Security Association Attributes

The following SA attribute definitions are used in Phase II of an IKE negotiation. Attribute types can be either Basic (B) or Variable-Length (V). Encoding of these attributes is defined in the base ISAKMP specification.

Attributes described as basic MUST NOT be encoded as variable. Variable length attributes MAY be encoded as basic attributes if their value can fit into two octets. See [IKE] for further information on attribute encoding in the IPSEC DOI. All restrictions listed in [IKE] also apply to the IPSEC DOI.

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

Class Values

SA Life Type
SA Duration

Specifies the time-to-live for the overall security association. When the SA expires, all keys negotiated under the association (AH or ESP) must be renegotiated. The life type values are:

RESERVED	0
seconds	1
kilobytes	2

Values 3-61439 are reserved to IANA. Values 61440-65535 are for private use. For a given Life Type, the value of the Life Duration attribute defines the actual length of the component lifetime -- either a number of seconds, or a number of Kbytes that can be protected.

If unspecified, the default value shall be assumed to be 28800 seconds (8 hours).

An SA Life Duration attribute MUST always follow an SA Life Type which describes the units of duration.

See Section 4.5.4 for additional information relating to lifetime notification.

Group Description

Specifies the Oakley Group to be used in a PFS QM negotiation. For a list of supported values, see Appendix A of [IKE].

Encapsulation Mode

RESERVED	0
Tunnel	1
Transport	2

Values 3-61439 are reserved to IANA. Values 61440-65535 are for private use.

If unspecified, the default value shall be assumed to be unspecified (host-dependent).

Authentication Algorithm

RESERVED	0
HMAC-MD5	1
HMAC-SHA	2
DES-MAC	3
KPDK	4

Values 5-61439 are reserved to IANA. Values 61440-65535 are for private use.

There is no default value for Auth Algorithm, as it must be specified to correctly identify the applicable AH or ESP transform, except in the following case.

When negotiating ESP without authentication, the Auth Algorithm attribute MUST NOT be included in the proposal.

When negotiating ESP without confidentiality, the Auth Algorithm attribute MUST be included in the proposal and the ESP transform ID must be ESP_NULL.

Key Length

RESERVED	0

There is no default value for Key Length, as it must be specified for transforms using ciphers with variable key lengths. For fixed length ciphers, the Key Length attribute MUST NOT be sent.

Key Rounds

RESERVED	0

There is no default value for Key Rounds, as it must be specified for transforms using ciphers with varying numbers of rounds.

Compression Dictionary Size

```
RESERVED                0
```

Specifies the log2 maximum size of the dictionary.

There is no default value for dictionary size.

Compression Private Algorithm

Specifies a private vendor compression algorithm. The first three (3) octets must be an IEEE assigned company_id (OUI). The next octet may be a vendor specific compression subtype, followed by zero or more octets of vendor data.

4.5.1 Required Attribute Support

To ensure basic interoperability, all implementations MUST be prepared to negotiate all of the following attributes.

SA Life Type
SA Duration
Auth Algorithm

4.5.2 Attribute Parsing Requirement (Lifetime)

To allow for flexible semantics, the IPSEC DOI requires that a conforming ISAKMP implementation MUST correctly parse an attribute list that contains multiple instances of the same attribute class, so long as the different attribute entries do not conflict with one another. Currently, the only attributes which requires this treatment are Life Type and Duration.

To see why this is important, the following example shows the binary encoding of a four entry attribute list that specifies an SA Lifetime of either 100MB or 24 hours. (See Section 3.3 of [ISAKMP] for a complete description of the attribute encoding format.)

```
   Attribute #1:
     0x80010001  (AF = 1, type = SA Life Type, value = seconds)

   Attribute #2:
     0x00020004  (AF = 0, type = SA Duration, length = 4 bytes)
     0x00015180  (value = 0x15180 = 86400 seconds = 24 hours)

   Attribute #3:
     0x80010002  (AF = 1, type = SA Life Type, value = KB)
   Attribute #4:
     0x00020004  (AF = 0, type = SA Duration, length = 4 bytes)
     0x000186A0  (value = 0x186A0 = 100000KB = 100MB)
```

If conflicting attributes are detected, an ATTRIBUTES-NOT-SUPPORTED Notification Payload SHOULD be returned and the security association setup MUST be aborted.

4.5.3 Attribute Negotiation

If an implementation receives a defined IPSEC DOI attribute (or attribute value) which it does not support, an ATTRIBUTES-NOT-SUPPORT SHOULD be sent and the security association setup MUST be aborted, unless the attribute value is in the reserved range.

If an implementation receives an attribute value in the reserved range, an implementation MAY chose to continue based on local policy.

4.5.4 Lifetime Notification

When an initiator offers an SA lifetime greater than what the responder desires based on their local policy, the responder has three choices: 1) fail the negotiation entirely; 2) complete the negotiation but use a shorter lifetime than what was offered; 3) complete the negotiation and send an advisory notification to the initiator indicating the responder’s true lifetime. The choice of what the responder actually does is implementation specific and/or based on local policy.

To ensure interoperability in the latter case, the IPSEC DOI requires the following only when the responder wishes to notify the initiator: if the initiator offers an SA lifetime longer than the responder is willing to accept, the responder SHOULD include an ISAKMP Notification Payload in the exchange that includes the responder’s IPSEC SA payload. Section 4.6.3.1 defines the payload layout for the RESPONDER-LIFETIME Notification Message type which MUST be used for this purpose.

4.6 IPSEC Payload Content

The following sections describe those ISAKMP payloads whose data representations are dependent on the applicable DOI.

4.6.1 Security Association Payload

The following diagram illustrates the content of the Security Association Payload for the IPSEC DOI. See Section 4.2 for a description of the Situation bitmap.

```
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !  Next Payload !   RESERVED    !         Payload Length        !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !                Domain of Interpretation (IPSEC)               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !                       Situation (bitmap)                      !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !                    Labeled Domain Identifier                  !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !  Secrecy Length (in octets)   !           RESERVED            !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                        Secrecy Level                          ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ! Secrecy Cat. Length (in bits) !           RESERVED            !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                    Secrecy Category Bitmap                    ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ! Integrity Length (in octets)  !           RESERVED            !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                       Integrity Level                         ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ! Integ. Cat. Length (in bits)  !           RESERVED            !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                  Integrity Category Bitmap                    ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Security Association Payload Format

The Security Association Payload is defined as follows:

- o Next Payload (1 octet) - Identifier for the payload type of the next payload in the message. If the current payload is the last in the message, this field will be zero (0).

- o RESERVED (1 octet) - Unused, must be zero (0).

- o Payload Length (2 octets) - Length, in octets, of the current payload, including the generic header.

- o Domain of Interpretation (4 octets) - Specifies the IPSEC DOI, which has been assigned the value one (1).

- o Situation (4 octets) - Bitmask used to interpret the remainder of the Security Association Payload. See Section 4.2 for a complete list of values.

- Labeled Domain Identifier (4 octets) - IANA Assigned Number used to interpret the Secrecy and Integrity information.

- Secrecy Length (2 octets) - Specifies the length, in octets, of the secrecy level identifier, excluding pad bits.

- RESERVED (2 octets) - Unused, must be zero (0).

- Secrecy Level (variable length) - Specifies the mandatory secrecy level required. The secrecy level MUST be padded with zero (0) to align on the next 32-bit boundary.

- Secrecy Category Length (2 octets) - Specifies the length, in bits, of the secrecy category (compartment) bitmap, excluding pad bits.

- RESERVED (2 octets) - Unused, must be zero (0).

- Secrecy Category Bitmap (variable length) - A bitmap used to designate secrecy categories (compartments) that are required. The bitmap MUST be padded with zero (0) to align on the next 32-bit boundary.

- Integrity Length (2 octets) - Specifies the length, in octets, of the integrity level identifier, excluding pad bits.

- RESERVED (2 octets) - Unused, must be zero (0).

- Integrity Level (variable length) - Specifies the mandatory integrity level required. The integrity level MUST be padded with zero (0) to align on the next 32-bit boundary.

- Integrity Category Length (2 octets) - Specifies the length, in bits, of the integrity category (compartment) bitmap, excluding pad bits.

- RESERVED (2 octets) - Unused, must be zero (0).

- Integrity Category Bitmap (variable length) - A bitmap used to designate integrity categories (compartments) that are required. The bitmap MUST be padded with zero (0) to align on the next 32-bit boundary.

4.6.1.1 IPSEC Labeled Domain Identifiers

The following table lists the assigned values for the Labeled Domain Identifier field contained in the Situation field of the Security Association Payload.

Domain	Value
RESERVED	0

4.6.2 Identification Payload Content

The Identification Payload is used to identify the initiator of the Security Association. The identity of the initiator SHOULD be used by the responder to determine the correct host system security policy requirement for the association. For example, a host might choose to require authentication and integrity without confidentiality (AH) from a certain set of IP addresses and full authentication with confidentiality (ESP) from another range of IP addresses. The Identification Payload provides information that can be used by the responder to make this decision.

During Phase I negotiations, the ID port and protocol fields MUST be set to zero or to UDP port 500. If an implementation receives any other values, this MUST be treated as an error and the security association setup MUST be aborted. This event SHOULD be auditable.

The following diagram illustrates the content of the Identification Payload.

```
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !  Next Payload !   RESERVED    !        Payload Length         !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !   ID Type     !  Protocol ID  !             Port              !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                     Identification Data                       ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Identification Payload Format

The Identification Payload fields are defined as follows:

- o Next Payload (1 octet) - Identifier for the payload type of the next payload in the message. If the current payload is the last in the message, this field will be zero (0).

- o RESERVED (1 octet) - Unused, must be zero (0).

- o Payload Length (2 octets) - Length, in octets, of the identification data, including the generic header.

- o Identification Type (1 octet) - Value describing the identity information found in the Identification Data field.

- Protocol ID (1 octet) - Value specifying an associated IP protocol ID (e.g. UDP/TCP). A value of zero means that the Protocol ID field should be ignored.

- Port (2 octets) - Value specifying an associated port. A value of zero means that the Port field should be ignored.

- Identification Data (variable length) - Value, as indicated by the Identification Type.

4.6.2.1 Identification Type Values

The following table lists the assigned values for the Identification Type field found in the Identification Payload.

ID Type	Value
RESERVED	0
ID_IPV4_ADDR	1
ID_FQDN	2
ID_USER_FQDN	3
ID_IPV4_ADDR_SUBNET	4
ID_IPV6_ADDR	5
ID_IPV6_ADDR_SUBNET	6
ID_IPV4_ADDR_RANGE	7
ID_IPV6_ADDR_RANGE	8
ID_DER_ASN1_DN	9
ID_DER_ASN1_GN	10
ID_KEY_ID	11

For types where the ID entity is variable length, the size of the ID entity is computed from size in the ID payload header.

When an IKE exchange is authenticated using certificates (of any format), any ID’s used for input to local policy decisions SHOULD be contained in the certificate used in the authentication of the exchange.

4.6.2.2 ID_IPV4_ADDR

The ID_IPV4_ADDR type specifies a single four (4) octet IPv4 address.

4.6.2.3 ID_FQDN

The ID_FQDN type specifies a fully-qualified domain name string. An example of a ID_FQDN is, "foo.bar.com". The string should not contain any terminators.

4.6.2.4 ID_USER_FQDN

The ID_USER_FQDN type specifies a fully-qualified username string, An example of a ID_USER_FQDN is, "piper@foo.bar.com". The string should not contain any terminators.

4.6.2.5 ID_IPV4_ADDR_SUBNET

The ID_IPV4_ADDR_SUBNET type specifies a range of IPv4 addresses, represented by two four (4) octet values. The first value is an IPv4 address. The second is an IPv4 network mask. Note that ones (1s) in the network mask indicate that the corresponding bit in the address is fixed, while zeros (0s) indicate a "wildcard" bit.

4.6.2.6 ID_IPV6_ADDR

The ID_IPV6_ADDR type specifies a single sixteen (16) octet IPv6 address.

4.6.2.7 ID_IPV6_ADDR_SUBNET

The ID_IPV6_ADDR_SUBNET type specifies a range of IPv6 addresses, represented by two sixteen (16) octet values. The first value is an IPv6 address. The second is an IPv6 network mask. Note that ones (1s) in the network mask indicate that the corresponding bit in the address is fixed, while zeros (0s) indicate a "wildcard" bit.

4.6.2.8 ID_IPV4_ADDR_RANGE

The ID_IPV4_ADDR_RANGE type specifies a range of IPv4 addresses, represented by two four (4) octet values. The first value is the beginning IPv4 address (inclusive) and the second value is the ending IPv4 address (inclusive). All addresses falling between the two specified addresses are considered to be within the list.

4.6.2.9 ID_IPV6_ADDR_RANGE

The ID_IPV6_ADDR_RANGE type specifies a range of IPv6 addresses, represented by two sixteen (16) octet values. The first value is the beginning IPv6 address (inclusive) and the second value is the ending IPv6 address (inclusive). All addresses falling between the two specified addresses are considered to be within the list.

4.6.2.10 ID_DER_ASN1_DN

The ID_DER_ASN1_DN type specifies the binary DER encoding of an ASN.1 X.500 Distinguished Name [X.501] of the principal whose certificates are being exchanged to establish the SA.

4.6.2.11 ID_DER_ASN1_GN

The ID_DER_ASN1_GN type specifies the binary DER encoding of an ASN.1 X.500 GeneralName [X.509] of the principal whose certificates are being exchanged to establish the SA.

4.6.2.12 ID_KEY_ID

The ID_KEY_ID type specifies an opaque byte stream which may be used to pass vendor-specific information necessary to identify which pre-shared key should be used to authenticate Aggressive mode negotiations.

4.6.3 IPSEC Notify Message Types

ISAKMP defines two blocks of Notify Message codes, one for errors and one for status messages. ISAKMP also allocates a portion of each block for private use within a DOI. The IPSEC DOI defines the following private message types for its own use.

Notify Messages - Error Types	Value
RESERVED	8192

Notify Messages - Status Types	Value
RESPONDER-LIFETIME	24576
REPLAY-STATUS	24577
INITIAL-CONTACT	24578

Notification Status Messages MUST be sent under the protection of an ISAKMP SA: either as a payload in the last Main Mode exchange; in a separate Informational Exchange after Main Mode or Aggressive Mode processing is complete; or as a payload in any Quick Mode exchange. These messages MUST NOT be sent in Aggressive Mode exchange, since Aggressive Mode does not provide the necessary protection to bind the Notify Status Message to the exchange.

Nota Bene: a Notify payload is fully protected only in Quick Mode, where the entire payload is included in the HASH(n) digest. In Main Mode, while the notify payload is encrypted, it is not currently included in the HASH(n) digests. As a result, an active substitution attack on the Main Mode ciphertext could cause the notify status message type to be corrupted. (This is true, in general, for the last message of any Main Mode exchange.) While the risk is small, a corrupt notify message might cause the receiver to abort the entire negotiation thinking that the sender encountered a fatal error.

Implementation Note: the ISAKMP protocol does not guarantee delivery of Notification Status messages when sent in an ISAKMP Informational Exchange. To ensure receipt of any particular message, the sender SHOULD include a Notification Payload in a defined Main Mode or Quick Mode exchange which is protected by a retransmission timer.

4.6.3.1 RESPONDER-LIFETIME

The RESPONDER-LIFETIME status message may be used to communicate the IPSEC SA lifetime chosen by the responder.

When present, the Notification Payload MUST have the following format:

- o Payload Length - set to length of payload + size of data (var)
- o DOI - set to IPSEC DOI (1)
- o Protocol ID - set to selected Protocol ID from chosen SA
- o SPI Size - set to either sixteen (16) (two eight-octet ISAKMP cookies) or four (4) (one IPSEC SPI)
- o Notify Message Type - set to RESPONDER-LIFETIME (Section 4.6.3)
- o SPI - set to the two ISAKMP cookies or to the sender’s inbound IPSEC SPI
- o Notification Data - contains an ISAKMP attribute list with the responder’s actual SA lifetime(s)

Implementation Note: saying that the Notification Data field contains an attribute list is equivalent to saying that the Notification Data field has zero length and the Notification Payload has an associated attribute list.

4.6.3.2 REPLAY-STATUS

The REPLAY-STATUS status message may be used for positive confirmation of the responder’s election on whether or not he is to perform anti-replay detection.

When present, the Notification Payload MUST have the following format:

- o Payload Length - set to length of payload + size of data (4)
- o DOI - set to IPSEC DOI (1)
- o Protocol ID - set to selected Protocol ID from chosen SA
- o SPI Size - set to either sixteen (16) (two eight-octet ISAKMP cookies) or four (4) (one IPSEC SPI)
- o Notify Message Type - set to REPLAY-STATUS
- o SPI - set to the two ISAKMP cookies or to the sender’s inbound IPSEC SPI
- o Notification Data - a 4 octet value:

```
0 = replay detection disabled
1 = replay detection enabled
```

4.6.3.3 INITIAL-CONTACT

The INITIAL-CONTACT status message may be used when one side wishes to inform the other that this is the first SA being established with the remote system. The receiver of this Notification Message might then elect to delete any existing SA’s it has for the sending system under the assumption that the sending system has rebooted and no longer has access to the original SA’s and their associated keying material. When used, the content of the Notification Data field SHOULD be null (i.e. the Payload Length should be set to the fixed length of Notification Payload).

When present, the Notification Payload MUST have the following format:

- o Payload Length - set to length of payload + size of data (0)
- o DOI - set to IPSEC DOI (1)
- o Protocol ID - set to selected Protocol ID from chosen SA
- o SPI Size - set to sixteen (16) (two eight-octet ISAKMP cookies)
- o Notify Message Type - set to INITIAL-CONTACT
- o SPI - set to the two ISAKMP cookies
- o Notification Data - <not included>

4.7 IPSEC Key Exchange Requirements

The IPSEC DOI introduces no additional Key Exchange types.

5. Security Considerations

This entire memo pertains to the Internet Key Exchange protocol ([IKE]), which combines ISAKMP ([ISAKMP]) and Oakley ([OAKLEY]) to provide for the derivation of cryptographic keying material in a secure and authenticated manner. Specific discussion of the various security protocols and transforms identified in this document can be found in the associated base documents and in the cipher references.

6. IANA Considerations

This document contains many "magic" numbers to be maintained by the IANA. This section explains the criteria to be used by the IANA to assign additional numbers in each of these lists. All values not explicitly defined in previous sections are reserved to IANA.

6.1 IPSEC Situation Definition

The Situation Definition is a 32-bit bitmask which represents the environment under which the IPSEC SA proposal and negotiation is carried out. Requests for assignments of new situations must be accompanied by an RFC which describes the interpretation for the associated bit.

If the RFC is not on the standards-track (i.e., it is an informational or experimental RFC), it must be explicitly reviewed and approved by the IESG before the RFC is published and the transform identifier is assigned.

The upper two bits are reserved for private use amongst cooperating systems.

6.2 IPSEC Security Protocol Identifiers

The Security Protocol Identifier is an 8-bit value which identifies a security protocol suite being negotiated. Requests for assignments of new security protocol identifiers must be accompanied by an RFC which describes the requested security protocol. [AH] and [ESP] are examples of security protocol documents.

If the RFC is not on the standards-track (i.e., it is an informational or experimental RFC), it must be explicitly reviewed and approved by the IESG before the RFC is published and the transform identifier is assigned.

The values 249-255 are reserved for private use amongst cooperating systems.

6.3 IPSEC ISAKMP Transform Identifiers

The IPSEC ISAKMP Transform Identifier is an 8-bit value which identifies a key exchange protocol to be used for the negotiation. Requests for assignments of new ISAKMP transform identifiers must be accompanied by an RFC which describes the requested key exchange protocol. [IKE] is an example of one such document.

If the RFC is not on the standards-track (i.e., it is an informational or experimental RFC), it must be explicitly reviewed and approved by the IESG before the RFC is published and the transform identifier is assigned.

The values 249-255 are reserved for private use amongst cooperating systems.

6.4 IPSEC AH Transform Identifiers

The IPSEC AH Transform Identifier is an 8-bit value which identifies a particular algorithm to be used to provide integrity protection for AH. Requests for assignments of new AH transform identifiers must be accompanied by an RFC which describes how to use the algorithm within the AH framework ([AH]).

If the RFC is not on the standards-track (i.e., it is an informational or experimental RFC), it must be explicitly reviewed and approved by the IESG before the RFC is published and the transform identifier is assigned.

The values 249-255 are reserved for private use amongst cooperating systems.

6.5 IPSEC ESP Transform Identifiers

The IPSEC ESP Transform Identifier is an 8-bit value which identifies a particular algorithm to be used to provide secrecy protection for ESP. Requests for assignments of new ESP transform identifiers must be accompanied by an RFC which describes how to use the algorithm within the ESP framework ([ESP]).

If the RFC is not on the standards-track (i.e., it is an informational or experimental RFC), it must be explicitly reviewed and approved by the IESG before the RFC is published and the transform identifier is assigned.

The values 249-255 are reserved for private use amongst cooperating systems.

6.6 IPSEC IPCOMP Transform Identifiers

The IPSEC IPCOMP Transform Identifier is an 8-bit value which identifier a particular algorithm to be used to provide IP-level compression before ESP. Requests for assignments of new IPCOMP transform identifiers must be accompanied by an RFC which describes how to use the algorithm within the IPCOMP framework ([IPCOMP]). In addition, the requested algorithm must be published and in the public domain.

If the RFC is not on the standards-track (i.e., it is an informational or experimental RFC), it must be explicitly reviewed and approved by the IESG before the RFC is published and the transform identifier is assigned.

The values 1-47 are reserved for algorithms for which an RFC has been approved for publication. The values 48-63 are reserved for private use amongst cooperating systems. The values 64-255 are reserved for future expansion.

6.7 IPSEC Security Association Attributes

The IPSEC Security Association Attribute consists of a 16-bit type and its associated value. IPSEC SA attributes are used to pass miscellaneous values between ISAKMP peers. Requests for assignments of new IPSEC SA attributes must be accompanied by an Internet Draft which describes the attribute encoding (Basic/Variable-Length) and its legal values. Section 4.5 of this document provides an example of such a description.

The values 32001-32767 are reserved for private use amongst cooperating systems.

6.8 IPSEC Labeled Domain Identifiers

The IPSEC Labeled Domain Identifier is a 32-bit value which identifies a namespace in which the Secrecy and Integrity levels and categories values are said to exist. Requests for assignments of new IPSEC Labeled Domain Identifiers should be granted on demand. No accompanying documentation is required, though Internet Drafts are encouraged when appropriate.

The values 0x80000000-0xffffffff are reserved for private use amongst cooperating systems.

6.9 IPSEC Identification Type

The IPSEC Identification Type is an 8-bit value which is used as a discriminant for interpretation of the variable-length Identification Payload. Requests for assignments of new IPSEC Identification Types must be accompanied by an RFC which describes how to use the identification type within IPSEC.

If the RFC is not on the standards-track (i.e., it is an informational or experimental RFC), it must be explicitly reviewed and approved by the IESG before the RFC is published and the transform identifier is assigned.

The values 249-255 are reserved for private use amongst cooperating systems.

6.10 IPSEC Notify Message Types

The IPSEC Notify Message Type is a 16-bit value taken from the range of values reserved by ISAKMP for each DOI. There is one range for error messages (8192-16383) and a different range for status messages (24576-32767). Requests for assignments of new Notify Message Types must be accompanied by an Internet Draft which describes how to use the identification type within IPSEC.

The values 16001-16383 and the values 32001-32767 are reserved for private use amongst cooperating systems.

7. Change Log

7.1 Changes from V9

- add explicit reference to [IPCOMP], [DEFLATE], and [LZS]
- allow RESPONDER-LIFETIME and REPLAY-STATUS to be directed at an IPSEC SPI in addition to the ISAKMP "SPI"
- added padding exclusion to Secrecy and Integrity Length text
- added forward reference to Section 4.5 in Section 4.4.4
- update document references

7.2 Changes from V8

- update IPCOMP identifier range to better reflect IPCOMP draft
- update IANA considerations per Jeff/Ted's suggested text
- eliminate references to DES-MAC ID ([DESMAC])
- correct bug in Notify section; ISAKMP Notify values are 16-bits

7.3 Changes from V7

- corrected name of IPCOMP (IP Payload Compression)
- corrected references to [ESPCBC]
- added missing Secrecy Level and Integrity Level to Figure 1
- removed ID references to PF_KEY and ARCFOUR
- updated Basic/Variable text to align with [IKE]
- updated document references and add intro pointer to [ARCH]
- updated Notification requirements; remove aggressive reference
- added clarification about protection for Notify payloads
- restored RESERVED to ESP transform ID namespace; moved ESP_NULL
- added requirement for ESP_NULL support and [ESPNULL] reference
- added clarification on Auth Alg use with AH/ESP
- added restriction against using conflicting AH/Auth combinations

7.4 Changes from V6

The following changes were made relative to the IPSEC DOI V6:

- added IANA Considerations section
- moved most IANA numbers to IANA Considerations section
- added prohibition on sending (V) encoding for (B) attributes
- added prohibition on sending Key Length attribute for fixed length ciphers (e.g. DES)
- replaced references to ISAKMP/Oakley with IKE
- renamed ESP_ARCFOUR to ESP_RC4
- updated Security Considerations section
- updated document references

7.5 Changes from V5

The following changes were made relative to the IPSEC DOI V5:

- changed SPI size in Lifetime Notification text
- changed REPLAY-ENABLED to REPLAY-STATUS
- moved RESPONDER-LIFETIME payload definition from Section 4.5.4 to Section 4.6.3.1
- added explicit payload layout for 4.6.3.3
- added Implementation Note to Section 4.6.3 introduction
- changed AH_SHA text to require SHA-1 in addition to MD5
- updated document references

7.6 Changes from V4

The following changes were made relative to the IPSEC DOI V4:

- moved compatibility AH KPDK authentication method from AH transform ID to Authentication Algorithm identifier
- added REPLAY-ENABLED notification message type per Architecture
- added INITIAL-CONTACT notification message type per list
- added text to ensure protection for Notify Status messages
- added Lifetime qualification to attribute parsing section
- added clarification that Lifetime notification is optional
- removed private Group Description list (now points at [IKE])
- replaced Terminology with pointer to RFC-2119
- updated HMAC MD5 and SHA-1 ID references
- updated Section 1 (Abstract)
- updated Section 4.4 (IPSEC Assigned Numbers)
- added restriction for ID port/protocol values for Phase I

7.7 Changes from V3 to V4

The following changes were made relative to the IPSEC DOI V3, that was posted to the IPSEC mailing list prior to the Munich IETF:

- added ESP transform identifiers for NULL and ARCFOUR

- renamed HMAC Algorithm to Auth Algorithm to accommodate DES-MAC and optional authentication/integrity for ESP
- added AH and ESP DES-MAC algorithm identifiers
- removed KEY_MANUAL and KEY_KDC identifier definitions
- added lifetime duration MUST follow lifetype attribute to SA Life Type and SA Life Duration attribute definition
- added lifetime notification and IPSEC DOI message type table
- added optional authentication and confidentiality restrictions to MAC Algorithm attribute definition
- corrected attribute parsing example (used obsolete attribute)
- corrected several Internet Draft document references
- added ID_KEY_ID per ipsec list discussion (18-Mar-97)
- removed Group Description default for PFS QM ([IKE] MUST)

Acknowledgments

This document is derived, in part, from previous works by Douglas Maughan, Mark Schertler, Mark Schneider, Jeff Turner, Dan Harkins, and Dave Carrel. Matt Thomas, Roy Pereira, Greg Carter, and Ran Atkinson also contributed suggestions and, in many cases, text.

References

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

[ARCH] Kent, S., and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[DEFLATE] Pereira, R., "IP Payload Compression Using DEFLATE", RFC 2394, August 1998.

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

[ESPCBC] Pereira, R., and R. Adams, "The ESP CBC-Mode Cipher Algorithms", RFC 2451, November 1998.

[ESPNULL] Glenn, R., and S. Kent, "The NULL Encryption Algorithm and Its Use With IPsec", RFC 2410, November 1998.

[DES] Madson, C., and N. Doraswamy, "The ESP DES-CBC Cipher Algorithm With Explicit IV", RFC 2405, November 1998.

[HMACMD5] Madson, C., and R. Glenn, "The Use of HMAC-MD5 within ESP and AH", RFC 2403, November 1998.

[HMACSHA] Madson, C., and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, November 1998.

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

[IPCOMP] Shacham, A., Monsour, R., Pereira, R., and M. Thomas, "IP Payload Compression Protocol (IPComp)", RFC 2393, August 1998.

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

[LZS] Friend, R., and R. Monsour, "IP Payload Compression Using LZS", RFC 2395, August 1998.

[OAKLEY] Orman, H., "The OAKLEY Key Determination Protocol", RFC 2412, November 1998.

[X.501] ISO/IEC 9594-2, "Information Technology - Open Systems Interconnection - The Directory: Models", CCITT/ITU Recommendation X.501, 1993.

[X.509] ISO/IEC 9594-8, "Information Technology - Open Systems Interconnection - The Directory: Authentication Framework", CCITT/ITU Recommendation X.509, 1993.

Author’s Address

Derrell Piper
Network Alchemy
1521.5 Pacific Ave
Santa Cruz, California, 95060
United States of America

Phone: +1 408 460-3822
EMail: ddp@network-alchemy.com

Full Copyright Statement