IPSEC Working Group
INTERNET-DRAFT
draft-ietf-ipsec-isakmp-05.ps

Douglas Maughan, Mark Schertler
Mark Schneider, Jeff Turner
June 13, 1996

Internet Security Association and Key Management Protocol (ISAKMP)

Abstract

This memo describes a protocol utilizing security concepts necessary for establishing Security Associations (SA) and cryptographic keys in an Internet environment. A Security Association protocol that negotiates, establishes, modifies and deletes Security Associations and their attributes is required for an evolving Internet, where there will be numerous security mechanisms and several options for each security mechanism. The key management protocol must be robust in order to handle public key generation for the Internet community at large and private key requirements for those private networks with that requirement.

The Internet Security Association and Key Management Protocol (ISAKMP) defines the procedures for authenticating a communicating peer, creation and management of Security Associations, key generation techniques, and threat mitigation (e.g. denial of service and replay attacks). All of these are necessary to establish and maintain secure communications (via IP Security Service or any other security protocol) in an Internet environment.

Status of this memo

This document is being submitted to the IETF Internet Protocol Security (IPSEC) Working Group for consideration as a method for the establishment and management of security associations and their appropriate security attributes. Additionally, this document proposes a method for key management to support IPSP and IPv6. Publication of this document does not imply acceptance of the concepts discussed by the IPSEC Working Group. Comments are solicited and should be addressed to the authors and/or the working group mailing list at `ipsec@tis.com`.

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as “working draft” or “work in progress.”

To learn the current status of any Internet-Draft, please check the “1id-abstracts.txt” listing contained in the Internet- Drafts Shadow Directories on ds.internic.net (US East Coast), nic.nordu.net (Europe), ftp.isi.edu (US West Coast), or munnari.oz.au (Pacific Rim).

Distribution of this document is unlimited.

Contents

List of Figures

1 Introduction

This document describes an Internet Security Association and Key Management Protocol (ISAKMP). ISAKMP combines the security concepts of authentication, key management, and security associations to establish the required security for government, commercial, and private communications on the Internet.

1.1 The Need for Negotiation

ISAKMP extends the assertion in [DOW92] that authentication and key exchanges must be combined for better security to include security association exchanges. The security services required for communications depends on the individual network configurations and environments. Organizations are setting up Virtual Private Networks (VPN), also known as Intranets, that will require one set of security functions for communications within the VPN and possibly many different security functions for communications outside the VPN to support geographically separate organizational components, customers, suppliers, sub-contractors (with their own VPNs), government, and others. Departments within large organizations may require a number of security associations to separate and protect data (e.g. personnel data, company proprietary data, medical) on internal networks and other security associations to communicate within the same department. Nomadic users wanting to "phone home" represent another set of security requirements. These requirements must be tempered with bandwidth challenges. Smaller groups of people may meet their security requirements by setting up "Webs of Trust". ISAKMP exchanges provide these assorted networking communities the ability to present peers with the security functionality that the user supports in an authenticated and protected manner for agreement upon a common set of security attributes, i.e. an interoperable security association.

1.2 What can be Negotiated?

Security associations must support different encryption algorithms, authentication mechanisms, and key establishment algorithms for other security protocols, as well as IP Security. Security associations must also support host-oriented certificates for lower layer protocols and user-oriented certificates for higher level protocols. Algorithm and mechanism independence is required in applications such as e-mail, remote login, and file transfer, as well as in session oriented protocols, routing protocols, and link layer protocols. ISAKMP provides a common security association and key establishment protocol for this wide range of security protocols, applications, security requirements, and network environments.

ISAKMP is not bound to any specific cryptographic algorithm, key generation technique, or security mechanism. This flexibility is beneficial for a number of reasons. First, it supports the dynamic communications environment described above. Second, the independence from specific security mechanisms and algorithms provides a forward migration path to better mechanisms and algorithms. When improved security mechanisms are developed or new attacks against current encryption algorithms, authentication mechanisms and key exchanges are discovered, ISAKMP will allow the updating of the algorithms and mechanisms without having to develop a completely new KMP or patch the current one.

ISAKMP has basic requirements for its authentication and key exchange components. These requirements guard against denial of service, replay / reflection, man-in-the-middle, and connection hijacking attacks. This is important because these are the types of attacks that are targeted against protocols. Complete Security Association (SA) support, which provides mechanism and algorithm independence, and protection from protocol threats are the strengths of ISAKMP.

1.3 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 implementation of
the item is an absolute requirement of the specification.

- MUST NOT

This phrase means that the definition is an absolute prohibition
of the specification.

- SHOULD

This word or the adjective "RECOMMENDED" means that there might
exist valid reasons in particular circumstances to not implement
this item, but the full implications should be understood and the
case carefully weighed before not implementing this or not
implementing in a conforming manner.

- MAY

This word or the adjective "OPTIONAL" means that implementation of
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.

- CONFORMANCE and COMPLIANCE

Conformance to this specification has the same meaning as
compliance to this specification.  In either case, the
mandatory-to-implement, or MUST, items MUST be fully implemented
as specified here.  If any mandatory item is not implemented as
specified here, that implementation is not conforming and not
compliant with this specification.
```

The remainder of section 1 outlines the major components of ISAKMP, i.e. Security Associations and Management, Authentication, Public Key Cryptography, and Miscellaneous items. Sections 2 covers the the design of the protocol and payload formats. Section 3 describes how the payload formats of ISAKMP are composed together as exchange types to establish a security association, perform key exchanges and authentication mechanisms. Sections 4 through 6 define the modification, deletion, and notification services offered by the ISAKMP protocol. The appendices provide examples of an Internet Domain of Interpretation (DOI), and a list of potential security association attributes.

1.4 Security Associations and Management

A Security Association (SA) is a relationship between two or more entities that describes how the entities will utilize security services to communicate securely. This relationship is represented by a set of information that can be considered a contract between the entities. The information must be agreed upon and shared between all the entities. Sometimes the information alone is referred to as an SA, but this is just a physical instantiation of the existing relationship. The existence of this relationship, represented by the information, is what provides the agreed upon security information needed by entities to securely interoperate. All entities must adhere to the SA for secure communications to be possible. When accessing SA attributes, entities use a pointer or identifier refered to as the Security Parameter Index (SPI). See [RFC-1825] for details on IP Security Associations (SA) and Security Parameter Index (SPI) definitions. The entire ISAKMP SA establishment is asymetrical.

1.4.1 Security Associations and Registration

The SA attributes required and recommended for the IP Security (AH, ESP) are defined in [RFC-1825]. The attributes specified for an IP Security SA include, but are not limited to, authentication mechanism, cryptographic algorithm, algorithm mode, key length, and Initialization Vector (IV). Other protocols that provide algorithm and mechanism independent security MUST define their requirements for SA attributes. The separation of ISAKMP from a specific SA definition is important to ensure ISAKMP can establish SAs for all possible security protocols and applications.

`NOTE:` See Appendix C for a discussion of SA attributes that should be considered when defining a security protocol or application.

In order to facilitate easy identification of specific attributes (e.g. a specific encryption algorithm) among different network entites the attributes must be assigned identifiers and these identifiers must be registered by a central authority. The Internet Assigned Numbers Authority (IANA) provides this function for the Internet.

1.4.2 ISAKMP Requirements

Security Association (SA) establishment MUST be part of the key management protocol defined for IP based networks. The SA concept is required to support security protocols in a diverse and dynamic networking environment. Just as authentication and key exchange must be linked to provide assurance that the key is established with the authenticated party [DOW92], SA establishment must be linked with the authentication and the key exchange protocol.

ISAKMP provides the protocol exchanges to establish a security association between negotiation server entities followed by the establishment of a security association by the negotiation server entities in behalf of some protocol (e.g. ESP/AH). First, an initial protocol exchange allows a basic set of security attributes to be agreed upon. This basic set provides protection for subsequent ISAKMP exchanges. It also indicates the authentication method and key exchange that will be performed as part of the ISAKMP protocol. If a basic set of security attributes is already in place between the negotiation server entities, the initial ISAKMP exchange may be skipped and the establishment of a security association can be done directly. After the basic set of security attributes has been agreed upon, initial identity authenticated, and required keys generated, the established SA can be used for subsequent communications by the entity that invoked ISAKMP. The basic set of SA attributes that MUST be implemented to provide ISAKMP interoperability are defined in Appendix A.

1.5 Authentication

A very important step in establishing secure network communications is authentication of the entity at the other end of the communication. Many authentication mechanisms are available. Authentication mechanisms fall into two catagories of strength - weak and strong. Passwords are an example of a mechanism that provides weak authentication. The reason passwords are considered weak is the fact that most users pick passwords that are easy to guess and when used over an unprotected network are easily read by network sniffers. Digital signatures, such as the Digital Signature Standard (DSS) and the Rivest-Shamir-Adleman (RSA) signature, are public key based strong authentication mechanisms. When using public key digital signatures each entity requires a public key and a private key. Certificates are an essential part of a digital signature authentication mechanism. Certificates bind a specific entity's identity (be it host, network, user, or application) to its public keys and possibly other security-related information such as privileges, clearances, and compartments. Authentication based on digital signatures requires a trusted third party or certificate authority to create, sign and properly distribute certificates. For more detailed information on digital signatures, such as DSS and RSA, and certificates see [Schneier].

1.5.1 Certificate Authorities

Certificates require an infrastructure for generation, verification, management and distribution. The Internet Policy Registration Authority (IPRA) [RFC-1422] has been established to direct this infrastructure for the IETF. The IPRA certifies Policy Certification Authorities (PCA). PCAs control Certificate Authorities (CA) which certify users and subordinate entities. Current certificate related work includes the Domain Name System (DNS) Security Extensions [DNSSEC] which will provide signed entity keys in the DNS. The Public Key Infrastucture (PKIX) working group is specifying an Internet profile for X.509 certificates. There is also work going on in industry to develop X.500 Directory Services which would provide X.509 certificates to users. The U.S. Post Office is developing a (CA) hierarchy. The NIST Public Key Infrastructure Working Group has also been doing work in this area. The DOD Multi Level Information System Security Initiative (MISSI) program has begun deploying a certificate infrastructure for the U.S. Government. Alternatively, if no infrastructure exists, the PGP Web of Trust certificates can be used to provide user authentication and privacy in a community of users who know and trust each other.

1.5.2 Entity Naming

An entity's name is its identity and is bound to its public keys in certificates. The CA MUST define the naming semantics for the certificates it issues. See the UNINETT PCA Policy Statements [Berge] for an example of how a CA defines its naming policy. When the certificate is verified, the name is verified and that name will have meaning within the realm of that CA. An example is the DNS security extensions which make DNS servers CAs for the zones and nodes they serve. Resource records are provided for public keys and signatures on those keys. The names associatied with the keys are IP addresses and domain names which have meaning to entities accessing the DNS for this information. A Web of Trust is another example. When webs of trust are set up, names are bound with the public keys. In PGP the name is usually the entities e-mail address which has meaning to those, and only those, who understand e-mail. Another web of trust could use an entirely different naming scheme.

1.5.3 ISAKMP Requirements

Strong authentication MUST be provided on ISAKMP exchanges. Without being able to authenticate the entity at the other end, the Security Association (SA) and session key established are suspect. Without authentication you are unable to trust an entity's identification, this makes access control questionable. While encryption (e.g. ESP) and integrity (e.g. AH) will protect subsequent communications from passive eavesdroppers, without authentication it is possible that the SA and key may have been established with an adversary who performed an active man-in-the-middle attack and is now stealing all your personal data.

A digital signature algorithm MUST be used within ISAKMP's authentication component. However, ISAKMP does not mandate a specific signature algorithm or certificate authority (CA). ISAKMP allows an entity initiating communications to indicate which CAs it supports. After selection of a CA, the protocol provides the messages required to support the actual authentication exchange. The protocol provides a facility for identification of different certificate authorities, certificate types (e.g. X.509, PKCS #7, PGP, DNS SIG and KEY records), and the exchange of the certificates identified.

ISAKMP utilizes digital signatures, based on public cryptography, for authentication. There are other strong authentication systems available, which could be specified as additional optional authentication mechanisms for ISAKMP. Some of these authentication systems rely on a trusted third party called a key distribution center (KDC) to distribute secret session keys. An example is Kerberos, where the trusted third party is the Kerberos server, which holds secret keys for all clients and servers within its network domain. A client's proof that it holds its secret key provides authenticaton to a server.

The ISAKMP specification does not specify the protocol for communicating with the trusted third parties (TTP) or certificate directory services. These protocols are defined by the TTP and directory service themselves and are outside the scope of this specification.

1.6 Public Key Cryptography

Public key cryptography is the most flexible, scalable, and efficient way for users to obtain the shared secrets and session keys needed to support the large number of ways Internet users will interoperate. Many key generation algorithms, that have different properties, are available to users (see [DOW92] and [ANSI]). Properties of key exchange protocols include the key establishment method, authentication, symmetry, perfect forward secrecy, and back traffic protection.

`NOTE:` Cryptographic keys can protect information for a considerable length of time. However, this is based on the assumption that keys used for protection of communications are destroyed after use and not kept for any reaso

1.6.1 Key Exchange Properties

Key Establishment (Key Generation / Key Transport) The two common methods of using public key cryptography for key establishment are key transport and key generation. An example of key transport is the use of the RSA algorithm to encrypt a randomly generated session key (for encrypting subsequent communications) with the recipient's public key. The encrypted random key is then sent to the recipient, who decrypts it using his private key. At this point both sides have the same session key, however it was created based on input from only one side of the communications. The benefit of the key transport method is that it has less computational overhead than the following method. The Diffie-Hellman (D-H) algorithm illustrates key generation using public key cryptography. The D-H algorithm is begun by two users exchanging public information. Each user then mathematically combines the other's public information along with their own secret information to compute a shared secret value. This secret value can be used as a session key or as a key encryption key for encrypting a randomly generated session key. This method generates a session key based on public and secret information held by both users. The benefit of the D-H algorithm is that the key used for encrypting messages is based on information held by both users and the independence of keys from one key exchange to another provides perfect forward secrecy. Detailed descriptions of these algorithms can be found in [Schneier]. There are a number of variations on these two key generation schemes and these variations do not necessarily interoperate.

Key Exchange Authentication Key exchanges may be authenticated during the protocol or after protocol completion. Authentication of the key exchange during the protocol is provided when each party provides proof it has the secret session key before the end of the protocol. Proof can be provided by encrypting known data in the secret session key during the protocol exchange. Authentication after the protocol must occur in subsequent communications. Authentication during the protocol is preferred so subsequent communications are not initiated if the secret session key is not established with the desired party.

Key Exchange Symmetry A key exchange provides symmetry if either party can initiate the exchange and exchanged messages can cross in transit without affecting the key that is generated. This is desirable so that computation of the keys does not require either party to know who initiated the exchange. While key exchange symmetry is desirable, symmetry in the entire key management protocol may provide a vulnerablity to reflection attacks.

Back Traffic Protection / Perfect Forward Secrecy Back traffic protection is provided by a key exchange protocol if disclosure of long-term cryptographic keying material (e.g. public signature keys) does not compromise previously generated keys. Perfect forward secrecy is provided by the independent generation of each key such that subsequent keys are not dependent on any previous key. There is a subtle difference. Future session keys will NOT be obtainable if the long-term key is compromised if the key exchange provides perfect forward secrecy, while past session keys will NOT be obtainable if the current session key is compromised if the key exchange provides back traffic protecion.

1.6.2 ISAKMP Requirements

An authenticated key exchange MUST be supported by ISAKMP. Users SHOULD choose additional key establishment algorithms based on their requirements. ISAKMP does not specify a specific key exchange. However, [IO-Res] describes a proposal for using the Oakley key exchange [Oakley] in conjunction with ISAKMP. Requirements that should be evaluated when choosing a key establishment algorithm include establishment method (generation vs. transport), perfect forward secrecy, back traffic protection, computational overhead, key escrow, and key strength. Based on user requirements, ISAKMP allows an entity initiating communications to indicate which key exchanges it supports. After selection of a key exchange, the protocol provides the messages required to support the actual key establishment.

1.7 ISAKMP Protection

1.7.1 Anti-Clogging (Denial of Service)

Of the numerous security services available, protection against denial of service always seems to be one of the most difficult to address. A "cookie" or anti-clogging token (ACT) is aimed at protecting the computing resources from attack without spending excessive CPU resources to determine its authenticity. An exchange prior to CPU-intensive public key operations can thwart some denial of service attempts (e.g. simple flooding with bogus IP source addresses). Absolute protection against denial of service is impossible, but this anti-clogging token provides a technique for making it easier to handle. The use of an anti-clogging token was introduced by Karn and Simpson in [Karn].

1.7.2 Connection Hijacking

ISAKMP prevents connection hijacking by linking the authentication, key exchange and security association exchanges. This linking prevents an attacker from allowing the authentication to complete and then jumping in and impersonating one entity to the other during the key and security association exchanges.

1.7.3 Man-in-the-Middle Attacks

Man-in-the-Middle attacks include interception, insertion, deletion, and modification of messages, reflecting messages back at the sender, replaying old messages and redirecting messages. ISAKMP features prevent these types of attacks from being successful. The linking of the ISAKMP exchanges prevents the insertion of messages in the protocol exchange. The ISAKMP protocol state machine is defined so deleted messages will not cause a partial SA to be created, the state machine will clear all state and return to idle. The state machine also prevents reflection of a message from causing harm. The requirement for a new cookie with time variant material for each new SA establishment prevents attacks that involve replaying old messages. The ISAKMP strong authentication requirement prevents an SA from being established with other then the intended party. Messages may be redirected to a different destination or modified but this will be detected and an SA will not be established. The ISAKMP specification defines where abnormal processing has occurred and recommends notifying the appropriate party of this abnormality.

1.8 Multicast Communications

While future Internet communications will increasingly be of a multicast nature, this document is presenting a security association and key management protocol from the unicast point of view. It is expected that multicast communications will require the same security services as unicast communications and may introduce the need for additional security services. The issues of distributing SPIs for multicast traffic are presented in [RFC-1825]. Multicast security issues are also discussed in [BC]. Upon agreement and implementation of a security association protocol for the Internet unicast environment, we fully intend to examine any additional security requirements for multicast communications. For an introduction to the issues related to multicast security consult the Internet Drafts, [Spar94a] and [Spar94b], describing Sparta's research in this area.

2 Description of the Protocol

The Internet Security Association and Key Management Protocol (ISAKMP) defines procedures and packet formats to establish, negotiate, modify and delete Security Associations (SA). SAs contain all the information required for execution of various network security services, such as the IP layer services (such as header authentication and payload encapsulation), transport or application layer services, or self-protection of negotiation traffic. ISAKMP defines payloads for exchanging key generation and authentication data. These formats provide a consistent framework for transferring key and authentication data which is independent of the key generation technique, encryption algorithm and authentication mechanism.

ISAKMP is one part of a full "negotiation server". In order to perform complete negotiation of security associations, a negotiation server must also contain a policy database, for determining what to negotiate, and at least one key exchange protocol. The key exchange protocol has the role of generating strings of secret, random data which can be transformed into cryptographic keying material.

For a particular communication channel, the negotiation server determines (from the policy database) the SA attributes that must be negotiated, and the acceptable values for these attributes. The negotiation server then uses ISAKMP to set up a "skeleton" SA, and uses a key exchange protocol to establish secret data which can be used to fill in the cryptographic attributes (i.e., keys) of the SA.

ISAKMP is distinct from key exchange protocols in order to cleanly separate the details of security association management (and key management) from the details of key exchange. There may be many different key exchange protocols, each with different security properties. However, a common framework is required for agreeing to the format of SA attributes, and for negotiating, modifying, and deleting SAs. ISAKMP serves as this common framework.

Separating the functionality into three parts adds complexity to the security analysis of a complete negotiation server. However, the separation is critical for interoperability between systems with differing security requirements, and should also simplify the analysis of further evolution of a negotiation server.

ISAKMP is intended to support the negotiation of SAs for security protocols at all layers of the network stack (e.g., IPSEC, SSL, TLSP, OSPF, etc.). By centralizing the management of the security associations, ISAKMP reduces the amount of duplicated functionality within each security protocol. ISAKMP can also reduce connection setup time, by negotiating a whole stack of services at once.

2.1 Basic ISAKMP Concepts

Figure 1 is a high level view of the placement of ISAKMP within a system context in a network architecture. An important part of negotiating security services is to consider the entire "stack" of individual SAs as a unit. We generally refer to this as a "protection suite", but sometimes use the term "system SA".

2.1.1 ISAKMP Terminology

Security Protocol A Security Protocol consists of an entity at a single point in the network stack, performing a security service for network communication. For example, IPSEC ESP and IPSEC AH are two different security protocols. SSL is another example. Security Protocols may perform more than one service, for example providing integrity and confidentiality in the one module.

```
+-------------+                  +--------------+
! Negotiation !   Situation     !  Application !
!    Server   !<----            !    Process   !
+-------------+     !            +--------------+
!   ISAKMP    !     !            ! Appl Protocol!
+-------------+     ! SPI        +--------------+
       !            v                  !
+----------------------------------------------+
!                Socket Layer                  !
+----------------------------------------------+
!        Transport Protocol (TCP / UDP)        !
+----------------------------------------------+
!                     IP                       !
+----------------------------------------------+
!             Link Layer Protocol              !
+----------------------------------------------+
```

Figure 1: ISAKMP Relationships

Security Association (SA) A Security Association is a security-protocol-specific set of parameters that completely defines the services necessary to protect network traffic at that security protocol. These parameters can include algorithm identifiers, modes, cryptographic keys, etc. The SA is referred to by its associated security protocol (for example, "ISAKMP SA", "ESP SA", "SSL SA").

ISAKMP SA An SA used by the negotiation server to protect its own traffic. Frequently a negotiation will consist of two phases: the first phase will establish an ISAKMP SA that will protect further negotiation traffic, and the second phase will consist of multiple

Security Parameter Index (SPI) An identifier for an SA, relative to some security protocol. Each security protocol has its own "SPI-space", so a (security protocol, SPI) pair uniquely identifies an SA.

Negotiation Server A negotiation server is an application process which interacts with different policy databases (security, network access, cryptographic, authentication, etc.) and uses ISAKMP to establish a security association. It calls upon ISAKMP to deliver the data required to establish an SA and upon a key management protocol to generate keying material and authenticate the exchange. The negotiation server can be invoked manually by a user or automatically by an up-call from a security protocol when it requires an SA.

Situation A situation contains all of the security-relevant information that a system considers necessary to decide the security services required to protect the session being negotiated. For example, in the Internet Security DOI (see Appendix A), the situation consists of only the address of the peer being contacted. In other DOIs, the situation may include security classifications, modes of operation (normal vs. emergency), etc. The situation contains the identification and credential information required by the negotiation server to make policy decisions. The negotiation server returns a SPI when an SA is established.

Protection Suite A protection suite is a list of the security services that must be applied by various security protocols. For example, a protection suite may consist of DES encryption in IP ESP, and keyed MD5 in IP AH. All of the protections in a suite must be treated as a single unit. This is because security services in different security protocols can have subtle interactions, and the effects of a suite must be analyzed and verified as a whole.

Proposal A proposal is a list, in decreasing order of preference, of the protection suites that a system considers acceptable to protect traffic under a given situation.

Domain of Interpretation A Domain of Interpretation (DOI) defines payload formats, exchange types, and conventions for naming security-relevant information such as security policies or cryptographic algorithms and modes. This Domain of Interpretation is unrelated to and independent of the Sensitivity Domain discussed in appendix B. DOIs are discussed in more detail in section 2.1.2.

Payload ISAKMP defines several types of payloads, which are used to transfer information such as security association data, or key exchange data, in DOI-defined formats. A payload consists of a generic payload header and a string of octects that is opaque to ISAKMP. ISAKMP uses DOI-specific functionality to synthesize and interpret these payloads. Multiple payloads can be sent in a single ISAKMP message. See 2.4 for more details on the payload types, and A for the formats of the basic Internet-DOI payloads.

Exchange Type An exchange type is a specification of the number of messages in an ISAKMP exchange, and the payload types that are contained in each of those messages. Each exchange type is designed to provide a particular set of security services, such as anonymity of the participants, perfect forward secrecy of the keying material, authentication of the participants, etc. There are three basic ISAKMP exchange types, defined in section 3.4. Other exchange types can be added if required.

2.1.2 Domain of Interpretation

A Domain of Interpretation (DOI) identifier is used to interpret the payloads of ISAKMP payloads. The concept of a DOI is based on previous work by the IETF CIPSO Working Group, but extended beyond security label interpretation to include naming and interpretation of security services. A DOI defines:

- A "situation": the set of information that will be used to determine the required security services.
- The set of security policies that must be supported.
- A syntax for the specification of proposed security services.
- A scheme for naming security-relevant information, including encryption algorithms, key exchange algorithms, security policy attributes, and certificate authorities.
- The specific formats of the various payload contents.
- Additional exchange types 3.5.6, if required.

The rules for the Internet Security DOI are presented in Appendix A. Specifications of the rules for customized DOIs will be presented in separate documents.

A system MAY support multiple Domains of Interpretation at once, but SHOULD take care to map all situations, etc., onto the same local notion of a situation (essentially the union of all of the situations). All systems MUST support the Internet Security DOI.

2.1.3 Two-Phased Negotiation

There are two phases to negotiating most network or transport level SAs. In the first phase, two negotiation servers agree on how to protect further negotiation traffic between themselves, establishing an ISAKMP SA. This ISAKMP SA is then used to protect the negotiations for the actual SA being requested.

While the two-phased approach has a higher start-up cost for most simple scenarios, there are several reasons that it is beneficial for most cases.

First, negotiation servers can amortize the cost of the first phase across several negotiations. For example, after the first phase of negotiation, the encryption provided by the ISAKMP SA can provide identity protection, potentially allowing the use of simpler second-phase exchanges.

Second, having an ISAKMP SA in place considerably reduces the cost of ISAKMP management activity - without the "trusted path" that an ISAKMP SA gives you, the negotiation servers would have to go through a complete re-authentication for each notify, delete, or modify of any SA.

2.2 ISAKMP Header Format

An ISAKMP message has a fixed header format, shown in Figure 2, followed by a variable number of payloads. A fixed header simplifies parsing, providing the benefit of protocol parsing software that is less complex and easier to implement. The fixed header contains the information required by the protocol to maintain state, process payloads and possibly prevent denial of service or replay attacks. The ISAKMP payload types are discussed in sections 2.3 and 2.4.

The ISAKMP Header fields are defined as follows:

- Initiator Cookie (8 octets) - Cookie of entity that initiated SA establishment, SA modification or SA deletion.
- Responder Cookie (8 octets) - Cookie of entity that is responding to an SA establishment, SA modifi-

```
                     1                   2                   3
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Initiator                            !
!                            Cookie                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Responder                            !
!                            Cookie                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! Vers  ! XCHG  !             Flags             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: ISAKMP Header Format

cation or SA deletion request.

- Next Payload (1 octet) - Indicates the type of the first payload in the message. The format for each payload is defined in sections 2.3 and 2.4. The processing for the payloads is defined in section 3.1.

Next Payload Type	Value
NONE	0
Envelope (ENV)	1
Security Association (SA)	2
Key Exchange (KE)	3
Identification (ID)	4
Certificate (CERT)	5
Hash (HASH)	6
Signature (SIG)	7
Nonce (NONCE)	8
Notification (N)	9
Delete (D)	10
Modify (M)	11
Future Use	12-255

- Version (4 bits) - indicates the version of the ISAKMP protocol in use.
- Exchange Type (4 bits) - indicates the type of exchange being used. This dictates the message and payload orderings in the ISAKMP exchanges.

Exchange Type	Value
RESERVED	0
Base	1
Identity Protection	2
Authentication Only	3
Future Use	4 - 15

- Flags (2 octets) - indicates specific options that are set for the ISAKMP exchange. The flags listed below are specified in the Flags field beginning with the least significant bit, i.e the Encryption bit is bit 0 of the Flags field, the Collate bit is bit 1 of the Flags field, etc.
  - E(ncryption Bit) (1 bit) - If set (1), all payloads following the header are encrypted using the encryption algorithm in the ISAKMP SA. The ISAKMP Identifier is the combination of the initiator and responder cookie. If the E(ncryption Bit) is not set (0), the payloads are not encrypted.
  - C(ollate Bit) (1 bit) - If set (1), the proposals in the Security Association payload MUST be processed in a collated manner. For example, if ISAKMP is negotiating security associations for ESP and AH simultaneously, then the first proposal for ESP MUST only be selected if the first proposal for AH is also selected and so on with the subsequent proposals. If the C(ollate Bit) is not set (0), the proposals can be selected in any combination without constraint.
- Length (4 octets) - Length of total message (header + payloads) in octets.

2.3 Identifying Security Associations

While bootstrapping secure channels between systems, ISAKMP cannot assume the existence of security services, and must provide some protections for itself. Therefore, ISAKMP considers an ISAKMP Security Association to be different than other types, and manages ISAKMP SAs itself, in their own name space. ISAKMP uses the two cookie fields in the header to identify ISAKMP SAs, but uses the SPI field in the Envelope payload (called a negotiation SPI) and the Auxiliary-SPI in the Security Association payload to identify SAs for other security protocols.

The interpretation of these four fields is dependent on the operation taking place. In general, the first cookie field identifies the ISAKMP SA that the receiver of the message will use to identify a message. The second cookie field contains the ISAKMP SPI that was used by the sender of the message. The two SPI fields are used similarly and may depend on the protocol being secured. For example, in the Internet DOI the SPI field in the Envelope payload identifies a security-protocol SA on the receiver's side, and the Aux-SPI field in the SA payload identifies the security-protocol SA on the sender's side. The only exceptions to these rules occur during "startup" operations, in which the two sides have not yet established the SPIs they will use.

The following table describes how the cookies in the ISAKMP header and SPI fields in the Envelope and SA payloads must be set for various operations.

The symbols used in the table are as follows:

- ISA-SPI-I - The ISAKMP SPI that the initiator will use to identify an ISAKMP SA.
- ISA-SPI-R - The ISAKMP SPI that the responder will use to identify an ISAKMP SA.
- SP-SPI-I - The SPI that the ISAKMP initiator will use to identify a security association for some security protocol. This field is located in the Envelope payload.
- SP-SPI-R - The SPI that the ISAKMP responder will use to identify a security association for some security protocol. This is the Aux-SPI located in the SA payload.

Operation	Cookie-I	Cookie-R	SPI	Aux SPI
Start ISAKMP SA negotiation	ISA-SPI-I	0	0	0
Respond ISAKMP SA negotiation	ISA-SPI-I	ISA-SPI-R	0	0
Init other SA negotiation	ISA-SPI-R	ISA-SPI-I	SP-SPI-I	0
Respond other SA negotiation	ISA-SPI-I	ISA-SPI-R	SP-SPI-I	SP-SPI-R
Other (KE, Auth, ...)	ISA-SPI-R	ISA-SPI-I	SP-SPI-R	SP-SPI-I

For uniformity, all SPIs are 8 octets long. When negotiating security associations for security protocols that use 4-octet SPIs, the first four octets will be used, and the last four will be zero.

The SPI field in the Envelope payload contains the receiving entity's SPI. If none has been established yet (i.e., the ISAKMP daemons are still negotiating how to protect themselves), the SPI MAY be set to 0 or it MAY contain the ISAKMP SPI (or cookie).

When using ISAKMP to negotiate an SA for another protocol, the SPI field in the Envelope payload that the initiator will use to identify the SA is sent in the SA payload.

2.3.1 Envelope Payload

The Envelope payload is used to group related payloads into entities which must be treated as a whole. If a collection of payloads can convey a new meaning (i.e. if the whole is greater than the sum of the parts), then they should be enveloped. It is expected that all payloads discussed in section 2.4 could be placed in an envelope, except Modify, Notify, and Delete. It is possible to send multiple Envelope payloads in a single ISAKMP message, thus, allowing the security negotiation for several protocols at the same time. This capability fosters a coordinated security positioning across several protocols. Figure 3 shows the format of the Envelope payload.

Each enveloped entity is evaluated alone, components of the entities MAY NOT be swapped. The initiator of the protocol can impose a selection criterion on the responder by using the Collate bit (see section 2.2). When this bit is set, the responder MUST select the same ordinal proposal for all entities, e.g. proposal one for Envelope #1 and proposal one for Envelope #2.

The Envelope Payload fields are defined as follows:

```
                     1                   2                   3
 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  ! # of Payloads !         Protocol-Id           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                Security Parameter Index (SPI)                 !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Envelope Payload

- 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, then this field will be 0. This field provides the "chaining" capability.
- Number of Payloads (1 octet) - The number of payloads contained in the envelope. Each payload within the Envelope payload has a length field for determining its length.
- Protocol-Id (2 octets) - Specifies the protocol identifier for the current negotiation. Examples might include IPSEC ESP, IPSEC AH, OSPF, SSL, etc.
- SPI (8 octets) - Security Parameter Index. The receiving entity's SPI is always in this field, except for the SA payload (described in section 2.4.1). The use of the SPI field was described in section 2.3.

The payload type for the Envelope Payload is one (1).

2.4 ISAKMP Payloads

ISAKMP payloads provide modular building blocks for constructing ISAKMP messages. The presence and ordering of payloads in ISAKMP is defined by and dependent upon the *Exchange Type Field* located in the ISAKMP Header (see Figure 2).

Each payload begins with a generic header, shown in Figure 4, which provides a payload "chaining" capability and clearly defines the boundaries of a payload.

Section 2.2 lists the ISAKMP payloads which are described in more detail in sections 2.4.1 through 2.4.10.

```
                     1                   2                   3
 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        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Generic Payload Header

The Generic Payload Header fields are defined as follows:

- 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, then this field will be 0. This field provides the "chaining" capability.
- RESERVED (1 octet) - Unused, set to 0.
- Payload Length (2 octets) - Length in octets of the current payload, including the generic payload header.

2.4.1 Security Association Payload

The Security Association Payload is used to negotiate security attributes and to indicate the Domain of Interpretation (DOI) and Situation under which the negotiation is taking place. Figure 5 shows the format of the Security Association payload.

```
                     1                   2                   3
 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  (DOI)                 !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Auxillary SPI (Aux-SPI)                  !
!                                                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                              !
~                           Situation                          ~
!                                                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                              !
~                            Proposal                          ~
!                                                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Security Association Payload

The Security Association Payload fields are defined as follows:

- Next Payload (1 octet) - Identifies the *next* payload in the message. If the current payload is the last in the message, then this field will be 0.
- RESERVED (1 octet) - Unused, set to 0.
- Payload Length (2 octets) - Specifies the payload length in 4-octet units, including the generic header.
- Domain of Interpretation (4 octets) - Identifies the DOI (as described in Section 2.1.2) under which this negotiation is taking place. For the Internet, the DOI is one (1). All other DOI values are reserved for future use.
- Auxillary SPI (8 octets) - The use of the Auxiliary SPI field is described in section 2.3.

- Situation (variable length) - A DOI-specific field that identifies the situation under which this negotiation is taking place. Situation is used to make policy decisions regarding the security attributes being negotiated. Specifics for the Internet DOI Situation are detailed in Appendix A.
- Proposal (variable length) - A DOI-specific field that contains a list of protection suites being negotiated. Specifics for the Internet DOI proposal are detailed in Appendix A.

The payload type for the Security Association Payload is two (2).

2.4.2 Key Exchange Payload

The Key Exchange Payload supports a variety of key exchange techniques. Example key exchanges are Oakley [Oakley], Diffie-Hellman, the enhanced Diffie-Hellman key exchange described in X9.42 [ANSI], the Key Exchange Algorithm (KEA) on the FORTEZZA card, and the RSA-based key exchange used by PGP. Figure 6 shows the format of the Key Exchange payload.

```
                     1                   2                   3
 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 !   RESERVED1   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!              KEI              !           RESERVED2           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
!                       Key Exchange Data                       !
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Key Exchange Payload Format

The Key Exchange Payload fields are defined as follows:

- 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, then this field will be 0.
- RESERVED1 (1 octet) - Unused, set to 0.
- Payload Length (2 octets) - Length in octets of the current payload, including the generic header.
- Key Exchange Identifier (2 octets) - Identifies and guides interpretation of the Key Exchange Data. Values for this field are DOI-specific. For the Internet DOI, see Appendix A.
- RESERVED2 (2 octet) - Unused, set to 0.
- Key Exchange Data (variable length) - Data required to generate a session key.

The payload type for the Key Exchange Payload is three (3).

2.4.3 Identification Payload

The Identification Payload contains DOI-specific data used to exchange identification information. This information is used for determining the identities of communicating peers and may be used for determining authenticity of information. Figure 7 shows the format of the Identification Payload.

```
                     1                   2                   3
 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  !   ID Type     !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                   Identification Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Identification Payload Format

The Identification Payload fields are defined as follows:

- 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, then this field will be 0.
- ID Type (1 octet) - Specifies the type of Identification being used. This field is DOI-dependent.
- Payload Length (2 octets) - Length in octets of the current payload, including the generic header.
- Identification Data (variable length) - Contains identity information. The values for this field are DOI-specific and the format is specified by the ID Type field. For the Internet DOI, see Appendix A.

The payload type for the Identification Payload is four (4).

2.4.4 Certificate Payload

The Certificate Payload provides a means to transport certificates via ISAKMP and can appear in any ISAKMP message. The format of the certificate must be encoded in the payload. Figure 8 shows the format of the Certificate Payload.

`NOTE:` Certificate types and formats are not generally bound to a DOI - it is expected that there will only be a few certificate types, and that most DOIs will accept all of these types.

The Certificate Payload fields are defined as follows:

- 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, then this field will be 0.
- RESERVED (1 octet) - Unused, set to 0.

```
                     1                   2                   3
 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        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!     Certificate Authority     !       Certificate Type        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Certificate Data                        ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Certificate Payload Format

- Payload Length (2 octets) - Length in octets of the current payload, including the generic header.
- Certificate Authority (2 octets) - This field identifies the party that generated the certificates used for authentication. Authorities must be assigned an identifier by the Internet Assigned Numbers Authority (IANA). Before being assigned an identifier, an authority must publish an RFC defining the authority's domain. [RFC-1422] describes the Internet Policy Registration Authority (IPRA) and the procedures for achieving this registration.

  If PGP certificates, based on the "web of trust", are carried in the authentication payload the Authentication Authority value is one (1).

  Example certificate authorities that would have to register for an identifier are:

  - RSA Commercial Certificate Authority (http://www_csc.rsa.com/netsite)
  - Stable Large E-mail Database (SLED) (http://www.four11.com)
  - U.S. Postal Service.
- Certificate Type (2 octets) - This field indicates the certificate payload format. This field is used by authentication authorities that support more than one certificate type. The certificate types supported by an authentication authority must be defined in the RFC required for certificate authority registration. Examples are:
  - PKCS #7 certificates
  - PGP certificates
  - DNS Signed Keys
  - Kerberos Tokens
  - X.509 certificates
- Certificate Data (variable) - Actual certificate data. The type of certificate is indicated by the Certificate Type field.

The payload type for the Certificate Payload is five (5).

2.4.5 Hash Payload

The Hash Payload contains data generated by the hash function (selected during the SA payload exchange), over some part of the message and/or ISAKMP state. This payload is used to verify the integrity of the data in an ISAKMP message. Figure 9 shows the format of the Hash Payload.

```
                     1                   2                   3
 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        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                           Hash Data                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Hash Payload Format

The Hash Payload fields are defined as follows:

- 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, then this field will be 0.
- RESERVED (1 octet) - Unused, set to 0.
- Payload Length (2 octets) - Length in octets of the current payload, including the generic header.
- Hash Data (variable length) - Data that results from applying the hash routine to the ISAKMP message and/or state.

The payload type for the Hash Payload is six (6).

2.4.6 Signature Payload

The Signature Payload contains data generated by the digital signature function (selected during the SA payload exchange), over some part of the message and/or ISAKMP state. This payload is used to verify the integrity of the data in the ISAKMP message, and may be of use for non-repudiation services. Figure 10 shows the format of the Signature Payload.

The Signature Payload fields are defined as follows:

- 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, then this field will be 0.
- RESERVED (1 octet) - Unused, set to 0.
- Payload Length (2 octets) - Length in octets of the current payload, including the generic header.

```
                     1                   2                   3
 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        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Signature Data                        ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Signature Payload Format

- Signature Data (variable length) - Data that results from applying the digital signature function to the ISAKMP message and/or state.

The payload type for the Signature Payload is seven (7).

2.4.7 Nonce Payload

The Nonce Payload contains random data used to guarantee liveness during an exchange and protect against replay attacks. Figure 11 shows the format of the Nonce Payload. It is expected the Nonce payload will be used only with the Authentication Only exchange (see section 3.4.3). If nonces are used by a particular key exchange, it is expected the nonces will be defined as part of the key exchange data and not transmitted as a separate payload. See section 2.4.2 for more details and placement of the nonces.

```
                     1                   2                   3
 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        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                            Nonce Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: Nonce Payload Format

The Nonce Payload fields are defined as follows:

- 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, then this field will be 0.
- RESERVED (1 octet) - Unused, set to 0.
- Payload Length (2 octets) - Length in octets of the current payload, including the generic header.
- Nonce Data (variable length) - Contains the random data generated by the transmitting entity.

The payload type for the Nonce Payload is eight (8).

2.4.8 Notification Payload

The Notification Payload contains both ISAKMP and DOI-specific data used to transmit informational data, such as error conditions, to an ISAKMP peer. Figure 12 shows the format of the Notification Payload. Because the Notification payload is unrelated to other payloads, it is expected that it will not be placed in an Envelope payload. It is possible to send multiple Notification payloads in a single ISAKMP message.

Note that there are two types of notification. The first type is a notification to a negotiation server. Since server communication channels are identified by cookie pairs, this type of notification is identified by a valid cookie pair in the ISAKMP header. The SPI value in this case is the cookie of the initiator of the notification. This notification is handled by the basic ISAKMP state machine.

The second type of notification pertains to an ongoing negotiation, and is addressed to a valid cookie pair along with valid SPIs that define the current negotiation. One example for this type of notification is to indicate why a proposal was rejected.

```
                     1                   2                   3
 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        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!       Protocol-ID           !      Notify Message Type       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                Security Parameter Index (SPI)                !
!                                                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                              !
~                       Notification Data                      ~
!                                                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: Notification Payload Format

The Notification Payload fields are defined as follows:

- 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, then this field will be 0.
- RESERVED (1 octet) - Unused, set to 0.
- Payload Length (2 octets) - Length in octets of the current payload, including the generic header.
- Protocol-Id (2 octets) - Specifies the protocol identifier for the current notification. Examples might include IPSEC ESP, IPSEC AH, OSPF, SSL, etc.
- Notify Message Type (2 octets) - Specifies the type of notification message (see section 6.1). Additional text is placed in the Notification Data field if specified by the DOI.

- SPI (8 octets) - Security Parameter Index. The receiving entity's SPI. The use of the SPI field is described in section 2.3.
- Notification Data (variable length) - Informational or error data. Values for this field are DOI-specific. For the Internet DOI, see Appendix A.

The payload type for the Notification Payload is nine (9).

2.4.9 Delete Payload

The Delete Payload contains a protocol-specific security association identifier that the sender has removed from its security association database and is, therefore, no longer valid. Figure 13 shows the format of the Delete Payload. Because the Delete payload is unrelated to other payloads, it is expected that it will not be placed in an Envelope payload. It is possible to send multiple Delete payloads in a single ISAKMP message.

`NOTE:` The Delete Payload is not a request for the responder to delete an SA, but an advisory to the responder. If the responder chooses to ignore the message, the next communication from the responder to the requestor, using that security association, will fail.

```
                     1                   2                   3
 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  !   RESERVED1   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!        Protocol-ID            !           RESERVED2           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                Security Parameter Index (SPI)                 !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13: Delete Payload Format

The Delete Payload fields are defined as follows:

- 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, then this field will be 0.
- RESERVED1 (1 octet) - Unused, set to 0.
- Payload Length (2 octets) - Length in octets of the current payload, including the generic header.
- Protocol-ID (2 octets) - ISAKMP can establish security associations for various protocols, including ISAKMP and IPsec. This field indentifies which security association database to apply the delete request.
- RESERVED2 (2 octet) - Unused, set to 0.
- Security Parameter Index (8 octets) - Identifies the specific security association to delete. Values for this field are DOI and protocol specific. For the Internet DOI, see Appendix A.

The payload type for the Delete Payload is ten (10).

2.4.10 Modify Payload

The Modify Payload contains a protocol-specific security association identifier that the sender wants to change. Figure 14 shows the format of the Modify Payload. The Modify payload is used to determine which existing SA will be changed. The Modify payload MUST be followed by an Envelope payload containing at least an SA payload. The contents of the envelope will be determined by the value of the Modify Type field.

```
                     1                   2                   3
 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  !  Modify Type  !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!        Protocol-ID            !           RESERVED            !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                Security Parameter Index (SPI)                 !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: Modify Payload Format

The Modify Payload fields are defined as follows:

- 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, then this field will be 0.
- Modify Type (1 octet) - indicates the type of modification being performed. This dictates the message and payload orderings that will follow in the ISAKMP exchanges.

Modify Type	Value
RESERVED	0
New Key and New Attributes	1
New Key with Existing Attributes	2
New Attributes Only	3
Future Use	4 - 255

- Payload Length (2 octets) - Length in octets of the current payload, including the generic header.
- Protocol-ID (2 octets) - ISAKMP can establish security associations for various protocols, including ISAKMP and IPsec. This field indentifies which security association database to apply the Modify request.
- RESERVED (2 octet) - Unused, set to 0.
- Security Parameter Index (8 octets) - Identifies the specific security association to modify. Values for this field are DOI and protocol specific. For the Internet DOI, see Appendix A.

The payload type for the Modify Payload is eleven (11).

2.5 Miscellaneous

2.5.1 Transport Protocol

ISAKMP can be implemented over any transport protocol or over IP itself. Implementations MUST include support for ISAKMP on the User Datagram Protocol (UDP) on port 500. UDP Port 500 has been assigned to ISAKMP by the Internet Assigned Numbered Authority (IANA). Implementations MAY additionally support ISAKMP over other transport protocols or over IP itself.

2.5.2 RESERVED Fields

The existence of RESERVED fields are strictly used to preserve byte alignment. All RESERVED fields in the ISAKMP protocol MUST be set to zero (0) when a packet is issued. The receiver SHOULD check the RESERVED fields for zero (0) and discard the packet if other values are found.

2.5.3 Anti-Clogging Token ("Cookie") Creation

The details of cookie generation are implementation dependent, but MUST satisfy these basic requirements (originally stated by Phil Karn):

```
1.    The cookie must depend on the specific parties.  This prevents
      an attacker from obtaining a cookie using a real IP address and
      UDP port, and then using it to swamp the victim with Diffie-
      Hellman requests from randomly chosen IP addresses or ports.

2.    It must not be possible for anyone other than the issuing
      entity to generate cookies that will be accepted by that
      entity.  This implies that the issuing entity must use local
      secret information in the generation and subsequent
      verification of a cookie.  It must not be possible to deduce
      this secret information from any particular cookie.

3.    The cookie generation function must be fast to thwart attacks
      intended to sabotage CPU resources.
```

Karn's suggested method for creating the cookie is to perform a fast hash (e.g. MD5) over the IP Source and Destination Address, the UDP Source and Destination Ports and a locally generated secret random value. ISAKMP requires that the cookie be unique for each SA establishment, SA modify and SA delete to help prevent replay attacks, therefore the date and time MUST be added to the information hashed.

3 Security Association Establishment

Security Association (SA) Establishment is the process of agreeing upon and exchanging all the security information that is required in an SA. Section 3.1 describes the processing by both initiator and responder during ISAKMP message exchanges. Section 3.2 explains the ISAKMP negotiation phases and the security protection provided by each. Section 3.3 describes how ISAKMP messages are formed from ISAKMP exchange types and payloads. The three basic ISAKMP exchange types, which MUST be implemented, are described in section 3.4. An example of defining a new exchange type is presented in 3.5.

3.1 General Message Processing

Every ISAKMP message has basic processing applied to insure protocol reliability, and to minimize threats, such as denial of service and replay attacks.

When transmitting an ISAKMP message, the transmitting entity (initiator or responder) MUST do the following:

1. Set a timer and initialize a retry counter.
2. If the timer expires, the ISAKMP message is resent and the retry counter is decremented.
3. If the retry counter reaches zero (0), the event, *RETRY LIMIT REACHED*, is logged in the appropriate system audit file.
4. The ISAKMP protocol machine clears all states and returns to IDLE.

3.1.1 ISAKMP Header Processing

When creating an ISAKMP message, the transmitting entity MUST do the following:

1. Create initiator cookie. See section 2.5.3 for details.
2. Determine the relevant security characteristics of the session (the situation).
3. Generate a proposal for protecting a session under that situation.
4. Construct an ISAKMP Header.
5. Construct other ISAKMP payloads, depending on the exchange type.
6. Transmit the message to the destination host as described in section 3.1.

When an ISAKMP message is received, the receiving entity (initiator or responder) MUST do the following:

1. Verifies the Initiator and Responder "cookies". If the cookie validation fails, the message is discarded and the following actions are taken:
   (a) The event, *INVALID COOKIE*, is logged in the appropriate system audit file.
   (b) No response is sent to the initiating entity. This will cause the transmission timer of the initiating entity to expire and force retransmission of the message.

2. Check the Next Payload field to confirm it is valid. If the Next Payload field validation fails, the message is discarded and the following actions are taken:
   (a) The event, *INVALID NEXT PAYLOAD*, is logged in the appropriate system audit file.
   (b) No response is sent to the initiating entity. This will cause the transmission timer of the initiating entity to expire and force retransmission of the message.
3. Check the Version field to confirm it is correct. If the Version field validation fails, the message is discarded and the following actions are taken:
   (a) The event, *INVALID ISAKMP VERSION*, is logged in the appropriate system audit file.
   (b) No response is sent to the initiating entity. This will cause the transmission timer of the initiating entity to expire and force retransmission of the message.
4. Check the Exchange Type field to confirm it is valid. If the Exchange Type field validation fails, the message is discarded and the following actions are taken:
   (a) The event, *INVALID EXCHANGE TYPE*, is logged in the appropriate system audit file.
   (b) No response is sent to the initiating entity. This will cause the transmission timer of the initiating entity to expire and force retransmission of the message.
5. Check the Flags field to ensure it contains correct values. If the Flags field validation fails, the message is discarded and the following actions are taken:
   (a) The event, *INVALID FLAGS*, is logged in the appropriate system audit file.
   (b) No response is sent to the initiating entity. This will cause the transmission timer of the initiating entity to expire and force retransmission of the message.
6. Using the value in the Next Payload field, the message is processed.

3.1.2 Envelope Payload Processing

When creating an Envelope Payload, the transmitting entity MUST do the following:

1. Determine the protocol identifier for which this negotiation is being performed.
2. Generate a unique pseudo-random SPI.
3. Construct an Envelope payload, depending on the additional payloads being transmitted.

When an Envelope payload is received, the receiving entity (initiator or responder) MUST do the following:

1. Check the Next Payload field to confirm it is valid. If the Next Payload field validation fails, the message is discarded and the following actions are taken:
   (a) The event, *INVALID NEXT PAYLOAD*, is logged in the appropriate system audit file.
   (b) No response is sent to the initiating entity. This will cause the transmission timer of the initiating entity to expire and force retransmission of the message.
2. Verify the Protocol Identifier. If the value in the Protocol-ID field is invalid, the message is discarded and the following actions are taken:

   (a) The event, *INVALID PROTOCOL IDENTIFIER*, is logged in the appropriate system audit file.
   (b) No response is sent to the initiating entity. This will cause the transmission timer of the initiating entity to expire and force retransmission of the message.

3. Verify the SPI value If the SPI validation fails, the message is discarded and the following actions are taken:
   (a) The event, *INVALID SPI*, is logged in the appropriate system audit file.
   (b) No response is sent to the initiating entity. This will cause the transmission timer of the initiating entity to expire and force retransmission of the message.

3.1.3 Security Association Payload Processing

When creating a Security Association Payload, the transmitting entity MUST do the following:

1. Determine the Domain of Interpretation for which this negotiation is being performed.
2. Generate a unique pseudo-random Auxiliary-SPI.
3. Determine the situation within the determined DOI for which this negotiation is being performed.
4. Determine the proposal(s) within the situation.
5. Construct a Security Association payload.
6. Transmit the message to the initiating host as described in section 3.1.

When a Security Association payload is received, the receiving entity (initiator or responder) MUST do the following:

1. Determine if the Domain of Interpretation (DOI) is supported. If not, the protocol machine must send a Notify message with a Message Type of DOI-NOT-SUPPORTED and return to IDLE.
2. Determine if the given situation can be protected. If not, the protocol machine must send a Notify message with a Message Type of SITUATION-NOT-SUPPORTED and return to IDLE.
3. Determine if the situation can use any of the proposed protection suites, or proposals, to protect the session. If none of the proposed suites are acceptable, then the protocol machine must send a NOTIFY message with a Message Type of NO-PROPOSAL-CHOSEN and return to IDLE.
4. Verify the Auxiliary SPI value. If the Auxiliary SPI validation fails, the message is discarded and the following actions are taken:
   (a) The event, *INVALID AUXILIARY SPI*, is logged in the appropriate system audit file.
   (b) No response is sent to the initiating entity. This will cause the transmission timer of the initiating entity to expire and force retransmission of the message.
5. Process the remaining payloads as defined by the Next Payload field.
6. Construct an ISAKMP message containing the ISAKMP Header and any additional payloads as dictated by the exchange type.
7. Transmit the message to the initiating host as described in section 3.1.

3.2 Protection and the Phases of Negotiation

ISAKMP offers two “phases” of negotiation: in the first phase, two (or more) ISAKMP servers negotiate an ISAKMP-level SA, which will be used to protect further ISAKMP traffic. This exchange usually occurs once between two hosts.

The second phase of negotiation is used to establish security associations for other security protocols. This phase can be used to protect many negotiations.

At the beginning of each phase of negotiation, a host can require certain security properties of the next phase. The following properties are taken from the Oakley specification [Oakley], in which they are called “options”:

- Identity Confidentiality
- Perfect Forward Secrecy of keying material
- Perfect Forward Secrecy of identities
- Proof of Negotiation

ISAKMP uses a combination of the properties of any existing ISAKMP channels, exchange types, and key exchange mechanisms to provide these properties. For example, it is likely that the first phase of negotiation will result in a channel that is encrypted. Therefore, no special steps need to be taken to protect the identities of the parties for which ISAKMP is negotiating.

On the other hand, if the channel established during the first phase is not adequate to protect identities, then ISAKMP must choose an exchange type and key exchange mechanism combination that can provide those properties during the exchange.

Also, ISAKMP may be able to choose simpler key exchange mechanisms for the second phase if the first phase has established strong properties for the ISAKMP channel.

Finally, ISAKMP may choose an exchange type/key exchange mechanism combination that provides enough material for an SA in a single phase. Such a combination is likely to be expensive, and will generally not be necessary, as the “phase 1/phase 2 ratio” will probably be very low. However, if it is deemed necessary, it is available.

Note that an exchange type, by itself, cannot provide perfect forward secrecy (PFS). PFS is a property of the combination of exchange type and key exchange mechanism. A DOI designer must explicitly identify the properties that each exchange type/key exchange combination will exhibit.

Note that different parties are being authenticated during each of the phases of negotiation. During the first phase, the parties being authenticated are the negotiation servers/hosts, while during the second phase, users or application level programs are being authenticated. This second authentication is only necessary if the policy requires it - it may be that the ISAKMP server can speak for the applications.

3.3 Building ISAKMP messages

ISAKMP supplies the basic syntax of a message exchange, but it is up to a DOI to specify the valid sequences of payloads that make up a successful exchange, and how to build and interpret those payloads. The basic building blocks for ISAKMP messages are the payload types described in sections 2.2, 2.3, and 2.4.

Exchange types are used in ISAKMP to define the content and ordering of a series of ISAKMP messages that are designed to meet certain requirements. Most exchange types will include all of the basic payload types - ENV, SA, KE, ID, SIG - and may include others. The primary difference between the exchange types is the ordering of the messages and the payload ordering within each message.

Certificate Payloads contain certificates in support of an actual key exchange. They MUST be accepted at any point after the SA attributes have been agreed upon by both sides.

The following notation is used to describe the ISAKMP exchange types, shown in the next section, with the message formats and associated payloads:

```
HDR is an ISAKMP header whose exchange type defines the payload orderings
ENV{} is the envelope payload. The brackets are not part of the
   protocol but are included to illustrate the payloads which
   are enveloped.
SA is an SA negotiation payload with one or more proposals. An
   initiator MAY provide multiple proposals for negotiation; a responder
   MUST reply with only one.
KE is the key exchange payload.
IDx is the identity payload for "x". x can be: "ii" or "ir"
   for the ISAKMP initiator and responder, respectively, or  x can
   be: "ui", "ur" (when the ISAKMP daemon is a proxy negotiator),
   for the user initiator and responder, respectively.
CERT is the certificate payload.
HASH is the hash payload.
SIG is the signature payload. The data to sign is exchange-specific.
NONCE is the nonce payload.

=> signifies "initiator to responder" communication
<= signifies "responder to initiator" communication
```

Payload encryption (noted by a '*' after the ISAKMP header) MUST begin immediately after the ISAKMP header and all payloads following the ISAKMP header MUST be encrypted.

3.4 ISAKMP Exchange Types

There are currently three Exchange Types defined for ISAKMP. They are:

1. Base Exchange
2. Identity Protection Exchange

3. Authentication Only Exchange

These exchanges define the content and flow of ISAKMP messages during SA establishment. The diagrams in 3.4.1, 3.4.2, and 3.4.3 show the message ordering for each exchange type as well as the payloads included in each message, and provide basic notes describing what has happened after each message exchange.

As discussed above, these exchange types can be used in either phase of negotiation. However, they may provide different security properties in each of the phases. With each of these exchanges, the combination of cookies and SPI fields identifies whether this exchange is being used in the first or second phase of a negotiation.

Note that the exchanges shown do not show “optional payloads”, like certificates. Certificate payloads SHOULD be included in an exchange whenever an appropriate directory service (e.g. Secure DNS [DNSSEC]) is not available to distribute certificates. The inclusion of certificate payloads SHOULD be after the SA payloads have been exchanged and agreed upon.

All ISAKMP implementations MUST implement all 3 exchanges (Base, Identity Protection, Authentication Only) specified in this section.

3.4.1 Base Exchange

The Base exchange combines the Key Exchange and Authentication payloads together, trading off the number of messages transmitted against protection of the identity of the negotiating parties. The Base exchange by itself does not provide protection of the parties involved in the negotiation, since the Authentication payloads are sent before the key exchange is completed. The following diagram shows the messages, payloads sent in each message, and notes for the Base exchange.

BASE EXCHANGE

Initiator	Direction	Responder	Note
HDR; ENV{SA}	=>		Begin ISAKMP-SA or Proxy negotiation
	<=	HDR; ENV{SA}	
			Basic SA agreed upon
HDR; ENV{KE; IDii; SIG}	=>		
			Initiator Identity Verified by Responder
	<=	HDR; ENV{KE; IDir; SIG}	
			Responder Identity Verified by Initiator Key Generated SA established

In the first message, the initiator has generated a proposal that it considers adequate to protect traffic for the given situation. The situation and proposal are included in the Security Association payload.

The responder receives the first message and follows the general message processing procedures outlined in section 3.1. If the responder accepts any of the protection suites proposed, the responder sends the second message consisting of the ISAKMP Header and Security Association payloads. If the responder does not accept any of the protection suites proposed, the responder must send a notification of the rejection, and the exchange is terminated.

In the third message, the initiator transmits an ISAKMP Header followed by Key Exchange, Identification, and Signature payloads. The Key Exchange payload contains the Key Exchange Identifier (KEI) and the associated Key Exchange Data. The Identification payload contains the initiator's identity data. The Signature payload contains data generated by the digital signature function which was selected during the Security Association payload exchange. The signature data should be calculated over the message contents, i.e. the ISAKMP Header, Key Exchange payload, and the Identification payload. If additional payloads are included, such as the Certificate payload (which might be necessary to ensure a valid signature), they should be placed prior to the Signature payload so the signature is calculated over the entire message.

The responder receives the third message and follows the general message processing procedures outlined in section 3.1. Using the digital signature function selected during the Security Association payload exchange, the responder will calculate a digital signature over the received message and verify this result with the previously received message. If this result is valid, the responder sends the fourth message consisting of the ISAKMP Header, followed by the Key Exchange, Identification, and Signature payloads. The Key Exchange payload contains the Key Exchange Identifier (KEI) and the associated Key Exchange Data. The Identification payload contains the responder's identity data. The Signature payload contains data generated by a digital signature function which was selected during the Security Association payload exchange. The responder will calculate a digital signature over the contents of the message, i.e. the ISAKMP Header, Key Exchange payload, and the Identification payload.

Also, if the third message is valid, the responder has enough information to begin the key generation function. Depending on the KEI contained in the third message, the fourth message may be transmitted before or after the keying material is generated. If the fourth message is sent before the keying material is generated, the key generation function will take place on the responder's host after the message is transmitted. If any errors occur during this message phase, either with the digital signature calculation or the key generation, no message is transmitted to the initiator, thus, forcing retransmission of the previous message.

Upon receipt of the fourth message, the initiator verifies the identity of the responder using the agreed upon digital signature function in a manner similar to that described in the previous paragraphs. If the result is valid, the initiator must perform the key generation function before proceeding.

`NOTE:` Both sides of this key exchange will function according to the specification of the KEI. It is important that the key exchange be completed before any further communications are performed.

If the results of the digital signature calculation and key generation are valid, then a Security Association (SA) has been established. If any errors take place during this message phase, no message is transmitted to the responder, thus, forcing retransmission of the previous message.

3.4.2 Identity Protection Exchange

The Identity Protection exchange starts the same as the base exchange, but separates the key exchange payload from the identification and authentication payloads into separate messages. In this exchange, the key exchange payload is transmitted in an ISAKMP message first, followed by a separate message to perform the identification and authentication functions. The benefit of this exchange is the ability to communicate with a person without disclosing either party's identity to passive attackers on the network. The following diagram shows the messages, payloads sent in each message, and notes for the Identity Protection exchange.

IDENTITY PROTECTION EXCHANGE

Initiator	Direction	Responder	NOTE
HDR; ENV{SA}	=>		Begin ISAKMP-SA or Proxy negotiation
	<=	HDR; ENV{SA}	
			Basic SA agreed upon
HDR; ENV{KE}	=>		
	<=	HDR; ENV{KE}	
			Key Generated
HDR*; ENV{IDii; SIG}	=>		
			Initiator Identity Verified by Responder
	<=	HDR*; ENV{IDir; SIG}	
			Responder Identity Verified by Initiator SA established

In the first message, the initiator has generated a proposal that it considers adequate to protect traffic for the given situation. The situation and proposal are included in the Security Association payload.

The responder receives the first message and follows the general message processing procedures outlined in section 3.1. If the responder accepts any of the protection suites proposed, the responder sends the second message consisting of the ISAKMP Header and Security Association payloads. If the responder does not accept any of the protection suites proposed, the responder must send a notification of the rejection, and the exchange is terminated.

In the third message, the initiator transmits an ISAKMP Header followed by the Key Exchange payload. This payload contains the Key Exchange Identifier (KEI) and the associated Key Exchange Data.

The responder receives the third message and follows the general message processing procedures outlined in section 3.1. If the third message is valid, the responder has enough information to begin the key generation function. Also, the responder sends the fourth message consisting of the ISAKMP Header and Key Exchange payload to the initiator. Depending on the KEI contained in the third message, the fourth message may be transmitted before or after the keying material is generated. If the fourth message is sent before the keying material is generated, the key generation function will take place on the responder's host after the message is transmitted. If any errors occur during this message phase, no message is transmitted to the initiator, thus, forcing retransmission of the previous message.

Upon receipt of the fourth message, the initiator must perform the key generation function before proceeding. If any errors take place during this message phase, no message is transmitted to the responder, thus, forcing retransmission of the previous message.

`NOTE:` Both sides of this key exchange will function according to the specification of the KEI. It is important that the key exchange be completed before continuing, thus, ensuring the identity protection properties.

In the fifth message, the initiator transmits an ISAKMP Header followed by Identification and Signature payloads. The Signature payload contains data generated by a digital signature function which was selected during the Security Association payload exchange. The signature data should be calculated over the message contents and include the previously generated key, i.e. the ordering of the information for the signature should be the ISAKMP Header, Identification payload, and the key generated during the previous message exchanges. If additional payloads are included, such as the Certificate payload, they should be placed prior to the Signature payload so the signature is calculated over the entire message.

The responder receives the fifth message and follows the general message processing procedures outlined in section 3.1. Using the digital signature function selected during the Security Association payload exchange, the responder will calculate a digital signature over the received message and verify this result with the previously received message. If this result is valid, the responder sends the sixth message consisting of the ISAKMP Header, Identification payload, and Signature payload. The responder will calculate a digital signature over the contents of the message, including the previously generated key, i.e. the ordering of the information for the signature should be the ISAKMP Header, Identification payload, and the generated key. If the result is not valid, no message is transmitted to the initiator, thus, forcing retransmission of the previous message.

Upon receipt of the sixth message, the initiator verifies the identity of the responder using the agreed upon digital signature function in a manner similar to that described in the previous paragraph. If the result is valid, then a Security Association (SA) has been established. If the result is not valid, no message is transmitted to the responder, thus, forcing retransmission of the previous message.

3.4.3 Authentication Only Exchange

The Authentication only exchange starts the same as the base exchange. In this exchange, only authentication information is transmitted. The benefit of this exchange is the ability to perform only authentication without the computational expense of computing keys. Using this exchange during negotiation, none of the transmitted information will be encrypted. However, the information may be encrypted in other places. For example, if encryption is negotiated during the first phase of a negotiation and the authentication only exchange is used in the second phase of a negotiation, then the authentication only exchange will be encrypted by the ISAKMP SAs negotiated in the first phase. The following diagram shows the messages, payloads sent in each message, and notes for the Authentication Only exchange.

AUTHENTICATION ONLY EXCHANGE

Initiator	Direction	Responder	NOTE
HDR; ENV{SA}	=>		Begin ISAKMP-SA or Proxy negotiation
	<=	HDR; ENV{SA}	
			Basic SA agreed upon
HDR; ENV{IDii; NONCE; SIG}	=>		
			Initiator Identity Verified by Responder
	<=	HDR; ENV{IDir; NONCE; SIG}	
			Responder Identity Verified by Initiator SA established

In the first message, the initiator has generated a proposal that it considers adequate to protect traffic of the given situation. The situation and proposal are included in the Security Association payload.

The responder receives the first message and follows the general message processing procedures outlined in section 3.1. If the responder accepts any of the protection suites proposed, the responder sends the second message consisting of the ISAKMP Header and Security Association payloads. If the responder does not accept any of the protection suites proposed, the responder must send a notification of the rejection, and the exchange is terminated.

In the third message, the initiator transmits an ISAKMP Header, followed by Identification, Nonce, and Signature payloads. The Nonce payload is included in the Authentication Only exchange to ensure the existence of live, random data. This provides protection against replay attacks when using the Authentication Only exchange. The Signature payload contains data generated by the digital signature function which was selected during the Security Association payload exchange. The signature data should be calculated over the entire message contents of the current message, i.e. ISAKMP Header, Identification payload, Nonce payload, and Signature payload header. If additional payloads are included, such as the Certificate payload, they should be placed prior to the Signature payload so the signature is calculated over the entire message.

The responder receives the third message and follows the general message processing procedures outlined in section 3.1. Using the digital signature function selected during the Security Association payload exchange, the responder will calculate a digital signature over the received message and verify this result with the previously received message. If this result is valid, the responder sends the fourth message consisting of the ISAKMP Header, followed by Identification, Nonce and Signature payloads. The responder will calculate a digital signature over the entire contents of the ISAKMP message, i.e. ISAKMP Header, Identification payload, Nonce payload, and Signature payload header. This calculation will be placed in the Signature payload. If the result of the calculation performed on the received message (i.e. third message) is not valid, no message is transmitted to the initiator, thus, forcing retransmission of the previous message.

Upon receipt of the fourth message, the initiator verifies the identity of the responder using the agreed upon digital signature function in a manner similar to that described in the previous paragraph. This is done by checking the contents of the Identification, Nonce and Signature payloads. If the result is valid, then a Security Association (SA) has been established. If the result is not valid, no message is transmitted to the responder, thus, forcing retransmission of the previous message.

3.5 Defining a new Domain of Interpretation

The basic Internet DOI may be sufficient to meet the security requirements of a large portion of the internet community. However, some groups may have a need to customize some aspect of a DOI, perhaps to add a different set of cryptographic algorithms, or perhaps because they want to make their security-relevant decisions based on something other than a host id or user id. Also, a particular group may have a need for a new exchange type, for example to support key management for multicast groups.

This section discusses guidelines for defining a new DOI. The full specification for the internet DOI can be found in A.

Defining a new DOI is likely to be an time-consuming process. If at all possible, it is recommended that the designer begin with an existing DOI and customize only the parts that are unacceptable.

If a designer chooses to start from scratch, the following MUST be defined:

- A “situation”: the set of information that will be used to determine the required security services.

- The set of security policies that must be supported.
- A scheme for naming security-relevant information, including encryption algorithms, key exchange algorithms, security policy
- A syntax for the specification of proposed security services. attributes, and certificate authorities.
- The specific formats of the various payload contents.
- Additional exchange types 3.5.6, if required.

3.5.1 Situation

The situation is the basis for deciding how to protect a communications channel. It must contain all of the data that will be used to determine the types and strengths of protections applied in an SA. For example, a US Department of Defense DOI would probably use unpublished algorithms and have additional special attributes to negotiate. These additional security attributes would be included in the situation.

3.5.2 Security Policies

Security policies define how various types of information must be categorized and protected. The DOI must define the set of security policies supported, because both parties in a negotiation must trust that the other party understands a situation, and will protect information appropriately, both in transit and in storage. In a corporate setting, for example, both parties in a negotiation must agree to the meaning of the term "proprietary information" before they can negotiate how to protect it.

Note that including the required security policies in the DOI only specifies that the participating hosts understand and implement those policies in a full system context.

3.5.3 Naming Schemes

Any DOI must define a consistent way to name cryptographic algorithms, certificate autorities, etc. This can usually be done by using IANA naming conventions, perhaps with some private extensions.

3.5.4 Syntax for Specifying Security Services

In addition to simply specifying how to name entities, the DOI must also specify the format for complete proposals of how to protect traffic under a given situation.

3.5.5 Payload Specification

The DOI must specify the format of each of the payload types. For several of the payload types, ISAKMP has included fields that would have to be present across all DOI (such as a certificate authority in the certificate payload, or a key exchange identifier in the key exchange payload).

3.5.6 Defining new Exchange Types

If the basic exchange types are inadequate to meet the requirements within a DOI, a designer can define up to thirteen extra exchange types per DOI. The designer creates a new exchange type by choosing an unused exchange type value, and defining a sequence of messages composed of strings of the ISAKMP payload types.

Note that any new exchange types must be rigorously analyzed for vulnerabilities. Since this is an expensive and imprecise undertaking, a new exchange type should only be created when absolutely necessary.

4 Security Association Modification

Security Association modification provides the ability to update security association attributes and parameters within an existing SA without having to establish a new SA. The use of this exchange can provide performance benefits without sacrificing the security of the existing communication. The most common use of this exchange will be to re-key an existing SA.

The modification of security information is protected by the existing security association. It is envisioned that the following operations would be performed using the Modify function:

- Generate new keying material and negotiate new security attributes.
- Generate new keying material and use existing security attributes.
- Use existing security association and keying material and negotiate new security attributes.

Further details of SA modification will be described in this section as they are solidified during prototype development.

4.1 Modification Procedures

The procedure for exchanging information to modify an SA are similiar to the SA establishment exchange. One major difference is that when a Modify payload is transmitted, it MUST be followed by an Envelope payload which would contain an SA payload and, possibly, a KE payload.

5 Security Association Deletion

During communications it is possible that hosts may be compromised or that information may be intercepted during transmission. Determining whether this has occurred is not an easy task and is outside the scope of this Internet-Draft. However, if it is discovered that transmissions are being compromised, then it is necessary to delete the current SA and establish a new SA.

The Delete message, consisting of an ISAKMP Header (shown in Figure 2) and one or more Delete payload(s) (shown in Figure 13), provides a controlled method of informing a peer entity that the initiating entity has deleted the SA(s). Chaining of multiple Delete payloads permits the deletion of any number of SAs with a single message. The receiving entity SHOULD clean up its local SA database. The receiving entity may be using the SA for secure communications with more than one party and would not want to actually delete the SA from its database in this case. However, upon receipt of a Delete message the SAs listed in the SA Identifier field of the Delete payload cannot be used with the initiating entity. The SA Establishment procedure must be invoked to re-establish secure communications.

5.1 Deletion Procedures

When issuing a Delete message, the issuing entity (initiator or responder) MUST do the following:

1. Determine the ISAKMP SA identifier (cookie) for the receiving entity. A security association MUST exist between the negotiation servers to protect the Delete request. If a security association does not exist, then a security association MUST be established.
2. Construct the Delete message by generating a Delete payload for each SA to be deleted. The protocol field in the Delete payload associates the SA Identifier to a specific protocol's (for example, ISAKMP or IPsec) SA database.
3. Depending on the ISAKMP SA Attributes, apply the agreed upon security services.
   (a) If the ISAKMP SA requires authentication, the Delete message is created and signed using the agreed algorithm. The result is placed in a Signature payload and chained to the end of the Delete message.
   (b) If the ISAKMP SA requires encryption, all payloads in the Delete message are encrypted and the Encrypted bit in the Flags field of the ISAKMP Header is set to one (1).
4. Transmit the packet to the destination host.
5. Update the protocol-specific SA database to reflect the SPI deletions.

Upon receipt of a Delete message, the receiving entity (initiator or responder) MUST do the following:

1. Check the ISAKMP header as described in section 3.1.
2. Depending on the ISAKMP SA Attributes, apply the agreed upon security services in the following order.
   (a) If the ISAKMP SA requires encryption, decrypt all payloads. If the decryption fails, the message will be unreadable. However, the following actions are taken:
      i. The event is logged in the appropriate system audit file.
      ii. Because the Delete message is a unidirectional message a retransmission will not be performed. The local security policy will dictate the procedures for continuing. However, we RECOMMEND that a Notify message (see section 6) with an Error Message Type of DECRYPTION-FAILED be sent to the originator of the message.
   (b) If the ISAKMP SA requires authentication, the message, minus the signature payload, is processed and the calculated signature is compared to the signature contained in the Signature payload of the message. If these signatures are not identical, the message is discarded and the following actions are taken:

i. The event is logged in the appropriate system audit file.
ii. Because the Delete message is a unidirectional message a retransmission will not be performed. The local security policy will dictate the procedures for continuing. However, we recommend that the SPIs in the Delete message be checked to see if the originator was the communicating party. If so, then these SAs can be deleted from the protocol-specific SA database. We also RECOMMEND that a Notify message (see section 6) with an Error Message Type of INVALID-SIGNATURE be sent to the originator of the message. If the SPIs do not match those of the originator, then no further action should be taken.

`NOTE:` Both of these cases, failed decryption and failed authentication must be handled carefully. If not, this may become an opening for attack.

3. Process all the Delete payloads found in the message.
4. Update the protocol-specific SA database to reflect the SPI deletions.

6 Notification Message

The Notify message, consisting of an ISAKMP Header (shown in Figure 2) and one or more Notify payload(s) (shown in Figure 12), contains information one party wants to send to another. The Notify message is unidirectional.

6.1 Notify Message Types

Notification information can be error messages specifying why a SA could not be established. It can also be status data that a process managing an SA database wishes to communicate with a peer process. For example, a secure front end or security gateway may use the Notify message to synchronize SA communication. The table below lists the Nofitication messages and their corresponding values.

NOTIFY MESSAGES - ERROR TYPES

Errors	Value
DOI-NOT-SUPPORTED	1
SITUATION-NOT-SUPPORTED	2
INVALID-COOKIE	3
INVALID-VERSION-NO	4
INVALID-EXCHANGE-TYPE	5
INVALID-SPI	6
ATTRIBUTES-NOT-SUPPORTED	7
NO-PROPOSAL-CHOSEN	8
BAD-PROPOSAL-SYNTAX	9
INVALID-SIGNATURE	10
DECRYPTION-FAILED	11
BAD-ID-RANGE	12

NOTIFY MESSAGES - STATUS TYPES

Status	Value
CONNECTED	32769

6.2 Notification Procedures

When issuing an Notify message, the issuing entity (initiator or responder) MUST do the following:

1. Determine SPI of receiving entity. When performing notifications between ISAKMP negotiation servers, the SPI will be an ISAKMP SPI (cookie) that defines the ISAKMP-to-ISAKMP channel, otherwise the SPI will be protocol-specific. For protocol-specific notifications, both ISAKMP SPIs (cookies) and protocol-specific SPIs must be valid.
2. Construct a Notify message by generating a Notify payload for each notification to be sent.
3. Depending on the SA Attributes, apply the agreed upon security services.
   (a) If the ISAKMP SA requires authentication, the Notify message is created and signed using the agreed algorithm. The result is placed in a Signature payload and chained to the end of the Notify message.
   (b) If the ISAKMP SA requires encryption, all payloads in the Notify message are encrypted and the Encrypted bit in the Flags field of the ISAKMP Header is set to one (1).
4. Transmit the packet to the destination host as described in section 3.1.

Upon receipt of a Notify message, the receiving entity (initiator or responder) MUST do the following:

1. Check the ISAKMP header as described in section 3.1.
2. Depending on the SA Attributes, apply the agreed upon security services in the following order.
   (a) If the SA requires encryption, decrypt all payloads. If decryption fails the message will be unreadable. However, the following actions are taken:
      i. The event is logged in the appropriate system audit file.
      ii. Because the Notify message is a unidirectional message a retransmission will not be performed. The local security policy will dictate the procedures for continuing. However, we RECOMMEND that a Notify message (see section 6) with an Error Message Type of DECRYPTION-FAILED be sent to the originator of the message.
   (b) If the ISAKMP SA requires authentication, the message, minus the signature payload, is processed and the calculated signature is compared to the signature contained in the Signature payload of the message. If these signatures are not identical, the message is discarded and the following actions are taken:
      i. The event is logged in the appropriate system audit file.
      ii. Because the Notify Message is a unidirectional message a retransmission will not be performed. The local security policy will dictate the procedures for continuing. However, we RECOMMEND that the SPIs in the Notify message be checked to see if the originator was the communicating party. If so, then the Notify payloads can be processed. We also RECOMMEND that a Notify message (see section 6) with an Error Message Type of INVALID-SIGNATURE be sent to the originator of the message. If the SPIs do not match those of the originator, then no further action should be taken.
3. Process all the Notify payloads found in the Notify message.
4. Depending on the DOI-specific Notify Message Type, additional processing may be necessary.

7 Conclusions

The Internet Security Association and Key Management Protocol (ISAKMP) is a well designed protocol aimed at the Internet of the future. The massive growth of the Internet will lead to great diversity in network utilization, communications, security requirements, and security mechanisms. ISAKMP contains all the features that will be needed for this dynamic and expanding communications environment.

ISAKMP's Security Association (SA) feature coupled with authentication and key establishment provides the security and flexibility that will be needed for future growth and diversity. This security diversity of multiple key exchange techniques, encryption algorithms, authentication mechanisms, security services, and security attributes will allow users to select the appropriate security for their network, communications, and security needs. The SA feature allows users to specify and negotiate security requirements with other users. An additional benefit of supporting multiple techniques in a single protocol is that as new techniques are developed they can easily be added to the protocol. This provides a path for the growth of Internet security services. ISAKMP supports both publicly or privately defined SAs, making it ideal for government, commercial, and private communications.

ISAKMP provides the ability to establish SAs for multiple security protocols and applications. These protocols and applications may be session-oriented or sessionless. Having one SA establishment protocol that supports multiple security protocols eliminates the need for multiple, nearly identical authentication, key exchange and SA establishment protocols when more than one security protocol is in use or desired. Just as IP has provided the common networking layer for the Internet, a common security establishment protocol is needed if security is to become a reality on the Internet. ISAKMP provides the common base that allows all other security protocols to interoperate.

ISAKMP follows good security design principles. It is not coupled to other insecure transport protocols, therefore it is not vulnerable or weakened by attacks on other protocols. Also, when more secure transport protocols are developed, ISAKMP can be easily migrated to them. ISAKMP also provides protection against protocol related attacks. This protection provides the assurance that the SAs and keys established are with the desired party and not with an attacker.

ISAKMP also follows good protocol design principles. Protocol specific information only is in the protocol header, following the design principles of IPv6. The data transported by the protocol is separated into functional payloads. As the Internet grows and evolves, new payloads to support new security functionality can be added without modifying the entire protocol.

A The Basic Internet Domain Of Interpretation

A.1 Background/Rationale

This is the base DOI that all ISAKMP implementations must support.

A.2 Basic Internet DOI Assigned Value

The Basic Internet DOI Assigned Number is one (1).

A.3 Situation

A situation contains the DOI-specific, security relevant information that is necessary to decide which security services are required to protect a communication session. For the Internet DOI, the situation is identity-based, and consists of either an IP address or a fully qualified domain name (FQDN). The representation of the situation in an SA Payload is shown below.

A.4 Security Policies

The Basic Internet DOI must support access control decisions based on internet IP addresses, and/or FQDNs.

A.5 Security Service Identifiers

This sections specifies identifiers for use in this DOI.

A.5.1 Supported Security Protocols

Values for supported security protocols are specified in the most recent "Assigned Numbers" RFC [STD-2]. Presented in the following tables are the values for ESP; AH; AH/HMAC-MD5; AH/HMAC-MD5 with Replay Detection; and the Combined DES-CBC, HMAC and Replay Prevention Security Tranforms.

Note that these transforms are all defined with respect to a particular security protocol.

Supported Security Protocols

Protocol	Assigned Value
RESERVED	0
IPSEC ESP	1
IPSEC AH	2
ISAKMP	3

Security protocol values 4-1024 are reserved for IANA use. Values 1025-15360 are reserved for future use. Values 15360-16384 are reserved for private use.

A.6 Attribute Value Assigned Numbers

This section defines attribute values, relative to each of the security protocol values.

A.6.1 IPSEC ESP

ESP Transforms

Transform	Assigned Value
RESERVED	0
RFC-1829, Transport Mode	1
RFC-1829, Tunnel Mode	2
DES-CBS-HMAC-Replay	3

Encryption Transform values 4-1024 are reserved for IANA Use. Values 1025-15360 are reserved for future use. Values 15360-16384 are reserved for private use.

SA Attribute Values These values are optionally transmitted in the ESP proposal, depending on the transform value selected. Not all attribute values apply to all transforms.

Attribute values are always transmitted as Type/Length/Value (TLV) constructs. All TLV lengths are multiples of 4. If the value does not fill the last 4 bytes, the TLV will be padded out with 0's to the four byte boundary. The table below specifies the tag values, and the types of the values.

ESP Attribute Values

Attribute TAG	Tag Value	Type	Notes
RESERVED	0		
Cryptographic Synch	1	Multiple Precision Integer	
Replay Protection	2	8 Bit Boolean	HMAC-MD5-Replay only
Key Length (in bits)	3	16 bit integer	
IV Length (in bits)	4	16 bit integer	

A.6.2 IPSEC AH

Tranform Values

AH Transforms

Transform	Assigned Value
RESERVED	0
RFC-1828	1
Keyed HMAC-MD5	2
Keyed HMAC-SHA	3
Keyed HMAC-MD5-Replay	4

Authentication Transform values 5-1024 are reserved for IANA use. Values 1025-15360 are reserved for future use. Values 15360-16384 are reserved for private use.

SA Attribute Values These values are optionally transmitted in the AH proposal, depending on the transform value selected. Not all attribute values apply to all transforms.

Attribute values are always transmitted as Type/Length/Value (TLV) constructs. All TLV lengths are multiples of 4. If the value does not fill the last 4 bytes, the TLV will be padded out with 0's to the four byte boundary. The table below specifies the tag values, and the types of the values.

AH Attribute Values

Attribute TAG	Tag Value	Type	Notes
RESERVED	0		
Replay Protection	1	8 Bit Boolean	HMAC-MD5-Replay only

A.6.3 ISAKMP

ISAKMP Attribute Values

Attribute TAG	Tag Value	Type	Notes
RESERVED	0		
Group Identifier	1	32 bit	
Encryption Algorithm	2	32 bit	Encapsulates Mode
Hash Algorithm	3	32 bit	
Authentication Algorithm	4	32 bit	

Key Exchange Identifier (KEI) Values

Key Exchange Identifiers (KEI)

Key Exchange	Assigned Value
Reserved	0
RSA	1
X9.42	2
Oakley	3

KEI values 4-1024 are reserved for IANA use. Values 1025-15360 are reserved for future use. Values 15360-16384 are reserved for private use.

A.7 Proposal Formats

This section defines the proposal formats for each of the security protocols. Each proposal groups the security attributes needed to perform a security function together. The proposal and attribute formats are defined so additions or modifications to the proposals or attributes will not require a modification to the protocol.

A.7.1 Basic Proposal Format

Figure 15 shows the generic SA proposal format which contains the SA attributes. There can be one or more SA attributes in each SA proposal. One or more SA proposals can be sent for each security protocol, but only one response per security protocol is allowed. A negative response, such as: IMPROPER SA PROPOSAL FORMAT, is returned in a NOTIFY payload.

```
                     1                   2                   3
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Proposal #   ! Proposal Len  !          Protocol #           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        SA Attributes                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15: SA Proposal Format

The SA Proposal fields are defined as follows:

- Proposal Number (1 octet) - The ranking of this proposal among the other protection suites being proposed for this security protocol. The first (i.e., most preferred) proposal has the number 1, and the values increase by one in order of decreasing preference.
- Proposal Length (1 octet) - Specifies the proposal length in 4-octet units. Each IP Security proposal is an integer multiple of 4 octets long.
- Protocol Number (1 octet) - Identifies the security protocol requiring the SA attributes proposed.
- SA Attributes - Variable length field containing the attributes for an SA.

A.7.2 IPSEC ESP

The ESP SA is defined in [RFC-1825]. This section defines the format for the ESP SA proposals.

Note that not all fields of a security association are negotiated. In particular, key lifetimes and SA lifetimes are purely a local issue, and should not be negotiated.

Also, since the algorithm used in the particular transform is tightly bound to that transform, it is not necessary to separately negotiate the algorithm used - agreement on the transform implies agreement on the algorithm, mode, and key length.

A transform value MUST be sent in the proposal. Other attributes are optional.

```
                     1                   2                   3
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   Proposal #  ! Proposal Len  !       Protocol # (value=1)    !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!       ESP Transform ID        !          RESERVED             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
~                       Attribute TLVs                          ~
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16: ESP Proposal Format

A.7.3 IPSEC AH

The AH SA are defined in [RFC-1825]. This section defines the format for the AH SA proposals.

Note that not all fields of the security association are negotiated. In particular, key lifetimes and SA lifetimes are purely a local issue, and should not be negotiated.

Also, since the algorithm used in the particular transform is tightly bound to that transform, it is not necessary to separately negotiate the algorithm used - agreement on the transform implies agreement on the algorithm, as well as key lengths.

A transform value MUST be sent in the proposal. Other attributes are optional.

A.7.4 ISAKMP

A.8 Payload Formats

This section defines the syntax for all of the payloads, under the Basic Internet DOI.

```
                     1                   2                   3
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   Proposal #  ! Proposal Len  !      Protocol # (value=2)     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!       Auth Transform ID       !          RESERVED             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
~                        Attribute TLVs                         ~
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 17: AH Proposal Format

A.8.1 Security Association Payload

The Security Association Payload is used to transmit the DOI under which a negotiation is to take place, the situation of the traffic that is to be protected, and the initiator's proposal for protecting traffic of that situation. The SA Payload also has a field that is used to identify the SPI that the initiator will be using to name the SA being established.

Situation There are six identification types: IPV4_ADDR (value 1), FQDN (value 2), USER_FQDN (value 3), IPV4_ADDR_RANGE (value 4), IPV6_ADDR (value 5), IPV6_ADDR_RANGE (value 6).

The IPV4_ADDR contains a single 4 byte IPv4 address.

FQDN is a fully qualified domain name, as used by the DNS protocol. Its form is an US-ASCII character string. The domain components are separated by "." characters, as in DNS.

USER_FQDN is a user id followed by a "." character, followed by a fully qualified domain name, as used by

```
                     1                   2                   3
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   Proposal #  ! Proposal Len  !       Protocol # (value=1)    !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!    ISAKMP Transform ID        !          RESERVED             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
~                        Attribute TLVs                         ~
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18: ISAKMP Proposal

```
                     1                   2                   3
 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        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          DOI (value = 1)                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               Auxillary SPI (value = Initiator SPI)           !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                           Situation                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                            Proposal                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 19: Internet DOI Security Association Payload

the DNS protocol's MBOX record. Its form is an US-ASCII character string.

IPV4_ADDR_RANGE specifies that two values are included. The first is an IPv4 address value, the second is an IPv4 netmask (Ones in the netmask indicate that the corresponding bit in the address is fixed, while zeros indicate a wildcard bit. To indicate a single host, a netmask of all ones is used.) If an ISAKMP peer rejects the address range that it receives, it MUST return a NOTIFY message of type "BAD-ID-RANGE". The NOTIFY payload MAY be followed by an ID payload of type IPV4_ADDR_RANGE containing the original address recieved and a new netmask indicating an acceptable range.

IPV6_ADDR specifies a single 8 byte IPv6 address.

IPV6_ADDR_RANGE is similar to IPV4_ADDR_RANGE but with two 8 byte values.

```
                     1                   2                   3
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  ID type      !   RESERVED    !      Situation Length         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
!                      Identification Data                      !
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 20: Internet DOI Situation

Proposals The proposals can be of any of the formats defined in section B.6.

Note that there is no indication of overall length of the proposal section. The length is the sum of the lengths of each of the proposals.

A.8.2 Key Exchange Payload

The Key Exchange Payload takes the form defined in section 2.4.2, with the KEI containing the value agreed upon in the SA exchange, and the key exchange data containing data generated by the appropriate key exchange mechansism.

XXX could just negotiate KEI (we do already), and then it would be implicit, can remove 32 bits.

A.8.3 Certificate Payload

The Certificate Payload takes the form defined in section 2.4.4.

A.8.4 Hash Payload

The Hash Payload takes the form defined in section 2.4.5, with the hash data containing data generated by the appropriate hash algorithm.

A.8.5 Signature Payload

The Signature Payload takes the form defined in section 2.4.6, with the sigature data containing data generated by the appropriate hash algorithm.

A.8.6 Nonce Payload

The Nonce Payload takes the form defined in section 2.4.7.

A.8.7 Notification Payload

The Notification Payload takes the form defined in section 2.4.8.

A.8.8 Delete Payload

The Delete Payload takes the form defined in section 2.4.9.

A.8.9 Identification Payload

The Identification Payload allows ISAKMP peers to identify the parties on whose behalf the exchange is taking place.

```
                     1                   2                   3
 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 !   ID type     !        Payload Length         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
!                     Identification Data                       !
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 21: ID Payload Format

There are six identification types:

1. IPV4_ADDR (value 1)
2. FQDN (value 2)
3. USER_FQDN (value 3)
4. IPV4_ADDR_RANGE (value 4)
5. IPV6_ADDR (value 5)
6. IPV6_ADDR_RANGE (value 6)

The IPV4_ADDR contains a single 4 byte IPv4 address.

FQDN is a fully qualified domain name, as used by the DNS protocol. Its form is an US-ASCII character string. The domain components are separated by "." characters, as in DNS.

USER_FQDN is a user id followed by a "." character, followed by a fully qualified domain name, as used by the DNS protocol's MBOX record. Its form is an US-ASCII character string.

IPV4_ADDR_RANGE specifies that two values are included. The first is an IPv4 address value, the second is an IPv4 netmask (Ones in the netmask indicate that the corresponding bit in the address is fixed, while zeros indicate a wildcard bit. To indicate a single host, a netmask of all ones is used.) If an ISAKMP peer rejects the address range that it receives, it MUST return a NOTIFY message of type "BAD-ID-RANGE". The NOTIFY payload MAY be followed by an ID payload of type IPV4_ADDR_RANGE containing the original address recieved and a new netmask indicating an acceptable range.

IPV6_ADDR specifies a single 8 byte IPv6 address.

IPV6_ADDR_RANGE is similar to IPV4_ADDR_RANGE but with two 8 byte values.

A.9 Additional Exchange Types

There are no additional exchange types defined for the Basic Internet DOI.

B The Labeled Internet Domain Of Interpretation

B.1 Background/Rationale

In addition to the basic identity-based controls that the Basic Internet DOI provides, many organizations also require the use of mandatory labels as a basis for access control. For example, the US DoD uses labels to classify information, and [CW87] discusses the use of labels in commercial settings.

To support the requirement of labeling, the Labeled Internet DOI introduces the following concepts:

Sensitivity Level A sensitivity level is an identifier denoting the relative sensitivity of information. Sensitivity Levels are interpreted in the context of a particular DOI. This permits different organizations to have different labelling schemes and permits organizations to tailor their labelling scheme to match their own policies. ISAKMP implementations MUST support a Sensitivity Label field 8 bits long. ISAKMP implementations SHOULD support label fields as large as 32 bits.

Compartment Bitmap This is a variable length bitmap, which augments the sensitivity labels with need-to-know compartments. Each bit in the map represents a separate compartment. Implementations MUST support any length up to and including 256 bits for the Compartment Bitmap attribute. Implementations MAY support longer length Compartment Bitmap attributes. The semantics of the bitmap are determined by the DOI.

For example, a commercial firm might use compartment 1 to indicate 'Mergers & Acquisitions', compartment 2 to indicate 'Human Resources' and compartment 3 to indicate 'Engineering'. See [CW87] for more information on commercial security policies.

B.2 Labeled Internet DOI Assigned Value

The Labeled Internet DOI Assigned Number is two (2).

B.3 Situation

A Basic Internet DOI situation contains identity information, either an IP address or a fully qualified domain name (FQDN), required to make identity-based access control decisions.

The Labeled Internet DOI augments the Basic DOI situation with a sensitivity label and a compartment bitmap.

Representation of the situation in an SA Payload is shown below.

B.4 Security Policies

The Labeled Internet DOI must support mandatory label-based access control policies, as well as the Basic DOI's identity-based policies.

B.5 Security Service Identifiers

The Security Service identifiers are identical to those in the Basic Internet DOI. The two DOIs share the same mechanisms, but differ in policy on how to employ the mechanisms.

B.6 Proposal Formats

The proposal formats are identical to those in the Basic Internet DOI.

B.7 Payload Formats

This section defines the proposal formats for each of the security protocols. Each proposal groups the security attributes needed to perform a security function together. The proposal and attribute formats are defined so additions or modifications to the proposals or attributes will not require a modification to the protocol.

This section only defines the payload formats that differ from the the Basic Internet DOI. Where there is no difference, the reader is referred to Basic Internet DOI specification.

B.7.1 Security Association Payload

The Labeled Internet DOI Security Association Payload differs from the Basic DOI SA Payload only in the format of the situation. For the full payload format, see the Basic DOI.

B.7.2 Key Exchange Payload

This payload takes the form defined in the Basic Internet DOI specification.

B.7.3 Certificate Payload

This payload takes the form defined in the Basic Internet DOI specification.

```
                     1                   2                   3
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!            RESERVED           !      Situation Length         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
!                     Identification Data                       !
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!    Label Length (in bytes)    !           RESERVED            !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
!                        Sensitivity Label                      !
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Comp. Bitmap Length (in bits) !           RESERVED            !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
!                       Compartment Bitmap                      !
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 22: Internet DOI Situation

B.7.4 Hash Payload

This payload takes the form defined in the Basic Internet DOI specification.

B.7.5 Signature Payload

This payload takes the form defined in the Basic Internet DOI specification.

B.7.6 Nonce Payload

This payload takes the form defined in the Basic Internet DOI specification.

B.7.7 Notification Payload

This payload takes the form defined in the Basic Internet DOI specification.

B.7.8 Delete Payload

This payload takes the form defined in the Basic Internet DOI specification.

B.7.9 Identification Payload

This payload takes the form defined in the Basic Internet DOI specification.

B.8 Additional Exchange Types

There are no additional exchange types defined for the Labeled Internet DOI.

C Security Association Attributes

This appendix contains a list of security attributes that should be considered when defining a Security Association (SA) for a security protocol or application. As an example, the security attributes culled from this list and required for an IP Security (AH, ESP) SA are defined in [RFC-1825]. The separation of ISAKMP from a specific SA definition is important to ensure ISAKMP can establish SAs for all possible security functionality. Each security function will be required to maintain a database of current SAs. This list is based upon an e-mail message [Kent94] to the IPSEC mail list from Steve Kent.

The authors welcome input on what are meaningful security attributes for an SA.

1. SAID.INBOUND
2. SAID.OUTBOUND
3. ENCAPSULATION
4. INBOUND-CRITERIA
   (a) IP-DESTINATION-ADDRESS
   (b) IP-SOURCE-ADDRESS
   (c) NEXT-PROTOCOL
   (d) IP-SECURITY-LABEL
   (e) TRANSPORT-DESTINATION-PORT
   (f) TRANSPORT-SOURCE-PORT
5. PEER-ADDRESS
6. AUTHENTICATION
   (a) ENABLED
   (b) MECHANISM
       - DIGITAL SIGNATURE
         i. KEY.INBOUND (Peer's Public Key)
         ii. KEY.OUTBOUND (Initator's Private Key)
7. ENCRYPTION
   (a) ENABLED
   (b) ALGORTIHM
   (c) KEY.INBOUND
   (d) KEY.OUTBOUND
   (e) IV.INBOUND
   (f) IV.OUTBOUND
8. INTEGRITY
   (a) ENABLED
   (b) PLAINTEXT
   (c) DIRECTION.ENABLED

   (d) DIRECTION.VALUE
   (e) ALGORITHM
   (f) KEY.OUTBOUND
   (g) KEY.INBOUND

9. COMPRESSION
   (a) ENABLED
   (b) ALGORITHM

10. REPLAY
   (a) ENABLED
   (b) SIZE
   (c) NUMBER.OUTBOUND
   (d) NUMBER.INBOUND
   (e) WINDOW.SIZE
   (f) WINDOW

11. FRAGMENTATION
   (a) INBOUND
   (b) OUTBOUND

12. KEY-MANAGEMENT
   (a) NEGOTIATED
   (b) TECHNIQUE
   (c) PARAMETERS
   (d) REKEY
      - GRACE
      - NEXT-SA
      - TIME-BASED
         i. ENABLE
         ii. TRIGGER
      - TRAFFIC-BASED
         i. ENABLE
         ii. PACKET-COUNT.INBOUND
         iii. PACKET-COUNT.OUTBOUND
         iv. TRIGGER.INBOUND
         v. TRIGGER.OUTBOUND

Security Considerations

Cryptographic analysis techniques are improving at a steady pace. The continuing improvement in processing power makes once computationally prohibitive cryptographic attacks more realistic. New cryptographic algorithms and public key generation techniques are also being developed at a steady pace. New security services and mechanisms are being developed at an accelerated pace. A consistent method of choosing from a variety of security services and mechanisms and to exchange attributes required by the mechanisms is important to security in the complex structure of the Internet. However, a system that locks itself into a single cryptographic algorithm, key exchange technique, or security mechanism will become increasingly vulnerable as time passes.

UDP is an unreliable datagram protocol and therefore its use in ISAKMP introduces a number of security considerations. Since UDP is unreliable, but a key management protocol must be reliable, the reliability is built into ISAKMP. While ISAKMP utilizes UDP as its transport mechanism, it doesn't rely on any UDP information (e.g. checksum, length) for its processing.

Another issue that must be considered in the development of ISAKMP is the effect of firewalls on the protocol. Many firewalls filter out all UDP packets, making reliance on UDP questionable in certain environments.

A number of very important security considerations are presented in [RFC-1825]. One bears repeating. Once a private session key is created, it must be safely stored. Failure to properly protect the private key from access both internal and external to the system completely nullifies any protection provided by the IP Security services.

Acknowledgements

Hilarie Orman, via the Oakley key exchange protocol, has significantly influenced the design of ISAKMP.

Dan Harkins and Dave Carrel of Cisco Systems provided design assistance with the protocol.

Marsha Gross, Bill Kutz, Mike Oehler, and Pete Sell provided significant input and review to this document.

Scott Carlson ported the TIS DNSSEC prototype to FreeBSD for use with the ISAKMP prototype.

Jeff Turner and Steve Smalley contributed to the prototype development and integration with ESP and AH.

Mike Oehler and Pete Sell performed interoperability testing with other ISAKMP implementors.

Thanks to Carl Muckenhirn of SPARTA, Inc. for his assistance with LaTeX.

References

[ANSI] ANSI, *X9.42: Public Key Cryptography for the Financial Services Industry – Establishment of Symmetric Algorithm Keys Using Diffie-Hellman*, Working Draft, April 19, 1996.

[RFC-1825] Randall Atkinson, *Security Architecture for the Internet Protocol*, RFC-1825, August, 1995.

[BC] Ballardie, A. and J. Crowcroft, *Multicast-specific Security Threats and Countermeasures*, Proceedings of 1995 ISOC Symposium on Networks & Distributed Systems Security, pp. 17-30, Internet Society, San Diego, CA, February 1995.

[Berge] Berge, N.H., *UNINETT PCA Policy Statements*, Internet-Draft, work in progress, November, 1995.

[CW87] Clark, D.D. and D.R. Wilson, *A Comparison of Commercial and Military Computer Security Policies*, Proceedings of the IEEE Symposium on Security & Privacy, Oakland, CA, 1987, pp 184-193.

[DOW92] W. Diffie, M.Wiener, P. Van Oorschot, *Authentication and Authenticated Key Exchanges*, Designs, Codes, and Cryptography, 2, 107-125, Kluwer Academic Publishers, 1992.

[DNSSEC] Eastlake III, D. and C. Kaufman, *Domain Name System Protocol Security Extensions*, Internet-Draft, work in progress, Feb, 1996.

[Karn] Karn, P. and B. Simpson, *The Photuris Key Management Protocol*, Internet-Draft, work in progress, February, 1996.

[RFC-1422] Steve Kent, *Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management*, RFC-1422, February 1993.

[Kent94] Steve Kent, *IPSEC SMIB*, e-mail to ipsec@ans.net, August 10, 1994.

[Oakley] H. K. Orman, *The Oakley Key Determination Protocol*, Internet-Draft, work in progress, May 1996.

[IO-Res] Harkins, D. and D. Carrel, *The Resolution of ISAKMP with Oakley*, Internet-Draft, work in progress, June 1996.

[STD-2] Reynolds, J. and J. Postel, *Assigned Numbers*, STD 2, October, 1994.

[Schneier] Bruce Schneier, *Applied Cryptography - Protocols, Algorithms, and Source Code in C (Second Edition)*, John Wiley & Sons, Inc., 1995.

[Spar94a] Harney H., C. Muckenhirn, and T. Rivers, *Group Key Management (GKMP) Architecture*, SPARTA, Inc., Internet-Draft, September, 1994.

[Spar94b] Harney H., C. Muckenhirn, and T. Rivers, *Group Key Management (GKMP) Specification*, SPARTA, Inc., Internet-Draft, September, 1994.

Addresses of Authors

The authors can be contacted at:

Douglas Maughan
Phone: 301-688-0847
E-mail: `wdmaugh@tycho.ncsc.mil`

Mark Schneider
Phone: 301-688-0851
E-mail: `mss@tycho.ncsc.mil`

Jeff Turner
Phone: 301-688-0849
E-mail: `sjt@epoch.ncsc.mil`

National Security Agency
ATTN: R23
9800 Savage Road
Ft. Meade, MD. 20755-6000

Mark Schertler
Terisa Systems, Inc.
4984 El Camino Real
Los Altos, CA. 94022
Phone: 415-919-1773
E-mail: mjs@terisa.com