IPSEC Working Group
INTERNET-DRAFT
draft-ietf-ipsec-isakmp-04.txt, .ps

Douglas Maughan, Mark Schertler
National Security Agency
February 21, 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

Contents

1 Introduction **4**
1.1 Authentication . . . 4
1.1.1 Certificate Authorities . . . 5
1.1.2 Entity Naming . . . 5
1.1.3 ISAKMP Requirements . . . 5
1.2 Security Associations and Management . . . 6
1.2.1 Security Associations and Registration . . . 6
1.2.2 ISAKMP Requirements . . . 7
1.3 Public Key Cryptography . . . 7
1.3.1 Key Exchange Properties . . . 7
1.3.2 ISAKMP Requirements . . . 8
1.4 ISAKMP Protection . . . 8
1.4.1 Anti-Clogging (Denial of Service) . . . 8
1.4.2 Connection Hijacking . . . 9
1.4.3 Man-in-the-Middle Attacks . . . 9
1.5 Multicast Communications . . . 9

2 Description of the Protocol **10**
2.1 ISAKMP Architecture . . . 10
2.2 ISAKMP Packet Exchanges . . . 11
2.2.1 Base Exchange . . . 11
2.2.2 Identity Protection Exchange . . . 11
2.2.3 Authentication Only Exchange . . . 12
2.3 ISAKMP Details . . . 13
2.3.1 Basic ISAKMP Concepts . . . 13
2.3.2 ISAKMP Header Format . . . 14
2.3.3 SPI Usage . . . 16
2.3.4 General Message Processing . . . 17
2.3.5 Transport Protocol . . . 18
2.3.6 RESERVED Fields . . . 18
2.3.7 Anti-Clogging Token ("Cookie") Creation . . . 18

3 Security Association Establishment **19**
3.1 Security Association Initialization . . . 19
3.1.1 SA Initialization Procedures . . . 20
3.2 Authentication and Key Exchange . . . 22
3.2.1 Authentication Payload Format . . . 22
3.2.2 Key Exchange Payload Format . . . 23
3.2.3 Authentication and Key Exchange Procedures . . . 24
3.3 Security Association Negotiation . . . 25
3.3.1 SA Negotiation Procedures . . . 26

4 Security Association Modification **28**
4.1 Modification Procedures . . . 28

5 Security Association Deletion **28**
5.1 Deletion Procedures . . . 29

6 Notification Message **30**
6.1 Notify Message Types . . . 31

6.2 Notification Procedures 31

7 Conclusions 33

A IP Security DOI 34

A.1 IP Security Proposal Formats 34
A.2 ESP SA and AH SA Proposals 36
A.3 Oakley Proposal 39
A.4 Attribute Class Assigned Numbers 41
A.5 Attribute Value Assigned Numbers 42
A.5.1 Sensitivity Level Assigned Numbers 42
A.5.2 Key Exchange Identifiers (KEI) Assigned Numbers 42
A.5.3 Encryption Transform Assigned Numbers 42

B ISAKMP Scenarios 43

B.1 Oakley Scenario 43
B.2 Virtual Private Network Scenario 45

C Security Association Attributes 47

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. ISAKMP extends the assertion in [DOW92] that authentication and key exchanges must be combined for better security to include security association exchanges. The security required for communications depends on the individual network configurations and environments. Organizations are setting up Virtual Private Networks (VPN) 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 inter-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 it supports in an authenticated and protected manner for agreement upon a common interoperable security association.

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 exchanges 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.

1.1 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. Reasons for this include the fact that most users pick easy to guess passwords 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 digital signatures each entity requires a public and a private key. Certificates are an essential part of a digital signature authentication mechanism. Certificates bind a specific entities 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].

1.1.1 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.

1.1.2 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 usaully 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.

1.1.3 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. Encryption (e.g. ESP) and integrity (e.g. AH) will protect subsequent communications from passive eavesdroppers, but the SA and key may be established with an adversary who performed an active man-in-the-middle attack and is now stealing all your personnal 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 it's network domain. A clients proof it holds it's secret key provides its 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.

1.2 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 SAs and SPIs definitions.

1.2.1 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 SA attributes requirements. 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.

1.2.2 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 entities. 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 on the communicating entities the initial ISAKMP exchange may be skipped and the key and authentication exchanges issued directly. After the basic set of security attributes has been agreed upon, initial identity authenticated, and required keys generated, another security attribute exchange takes place to establish the complete SA which will 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. *These atributes will be moved to a separate document to maintain separation of protocol and attributes.*

1.3 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.

1.3.1 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 then 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. Assuming checks for weak values neither party can force the session key to a predetermined value. 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 effecting 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 KMP may provide a vulnerablity to reflection attacks. The entire ISAKMP SA establishment is asymetrical.

Back Traffic Protection / Perfect Forward Secrecy Perfect forward secrecy 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. Back traffic protection 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. Past session keys will NOT be obtainable if the long-term key is compromised in perfect forward secrecy; Past session keys will NOT be obtainable if the current session key is compromised in back traffic protecion.

The difficulty of numerical factoring of large numbers has proven that 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 reason.

1.3.2 ISAKMP Requirements

An authenticate 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. 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.

1.4 ISAKMP Protection

1.4.1 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. Phil Karn in his Internet-Draft [Karn] has introduced a mechanism to provide a measure of denial of service protection through the use of a "cookie" exchange. This anti-clogging token (ACT) is aimed at protecting the computing resources from attack without spending excessive CPU resources to determine its authenticity. As described in [Karn], 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).

As noted by Karn, absolute protection against denial of service is impossible, but this anti-clogging token provides a technique for making it easier to handle.

1.4.2 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.

1.4.3 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.

1.5 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 IP security services, such as the IP Authentication Header (AH), the IP Encapsulating Security Payload (ESP), and routing protocol authentication mechanisms. ISAKMP includes packet formats for exchanging key generation and authentication data. These formats provide a consistent method of transferring key and authentication data that is independent of the key generation technique, encryption algorithm or authentication mechanism.

2.1 ISAKMP Architecture

The following figure is a high level view of the placement of ISAKMP in a network architecture.

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

Figure 1: ISAKMP Relationships

The negotiation server is an application process which interfaces with the different policy databases (security, network access, cryptographic, authentication, etc.) that a system may require. It calls upon ISAKMP to deliver the data required to establish an SA and key 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. 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.

2.2 ISAKMP Packet Exchanges

The Exchange field in the ISAKMP header currently has three values defined: the base exchange, the identity protection exchange, and the authentication only exchange. These exchanges define the flow of the ISAKMP packets during SA establishment. The diagrams in 2.2.1, 2.2.2, and 2.2.3 show the packet exchange ordering for each exchange type and provide basic notes describing what has happened after each packet exchange. These exchanges are a high level summary of the packet flow, they do not show processing or error handling. Detailed connection establishment processing is defined in sections 3 through 6.

2.2.1 Base Exchange

Sections 3.1 through 3.3 describe the three basic phases: SA Initialization, Key Exchange and Authentication, and SA Negotiation, that comprise the base exchange. The base exchange contains the minimum number of packet exchanges in order to reduce latency associated with SA establishment.

Initiator	Base Exchange Direction	Responder	Note
ISA_INIT_REQ	=>		
	<=	ISA_INIT_RESP	
			Basic SA selected
ISA_AUTH&KE_REQ	=>		
	<=	ISA_AUTH&KE_RESP	
			Identity Verified Key Generated Encryption Begun
ISA_NEG_REQ	=>		
	<=	ISA_NEG_RESP	SA Completed

2.2.2 Identity Protection Exchange

The identity protection exchange starts and ends the same as the base exchange, but separates the key exchange payload and the authentication payloads into separate packets. In this exchange, the key exchange is transmitted first followed by the strong authentication exchange. 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 ISA_KE_REQ and ISA_KE_RESP packets used for the key exchange portion of this exchange contain an ISAKMP header followed by the key exchange payload. The ISA_AUTH_REQ and ISA_AUTH_RESP packet used for the authentication portion of this exchange contain an ISAKMP header followed by the authentication payload.

Identity Protection Exchange

Initiator	Direction	Responder	Note
ISA_INIT_REQ	=>		
	<=	ISA_INIT_RESP	
			Basic SA selected
ISA_KE_REQ	=>		
	<=	ISA_KE_RESP	
			Key Generated Encryption Begun
ISA_AUTH_REQ	=>		
	<=	ISA_AUTH_RESP	
			Identity Verified
ISA_NEG_REQ	=>		
	<=	ISA_NEG_RESP	SA Completed

2.2.3 Authentication Only Exchange

The authentication only exchange starts and ends the same as the base exchange. In this exchange, the authentication information is the only information transmitted. The benefit of this exchange is the ability to perform only an authentication exchange without the computational expense of computing keys. Using this exchange, none of the transmitted information will be encrypted.

The ISA_AUTH_REQ and ISA_AUTH_RESP packet used for the authentication only exchange contain an ISAKMP header followed by the authentication payload.

Authentication Only Exchange

Initiator	Direction	Responder	Note
ISA_INIT_REQ	=>		
	<=	ISA_INIT_RESP	
			Basic SA selected
ISA_AUTH_REQ	=>		
	<=	ISA_AUTH_RESP	
			Identity Verified
ISA_NEG_REQ	=>		
	<=	ISA_NEG_RESP	SA Completed

2.3 ISAKMP Details

The following sections contain the details of ISAKMP. Sections 2.3.1 through 2.3.7 cover details that are pertinent to the entire protocol. Sections 3 through 6 define the establishment, modification, and deletion services, defined as exchanges, offered by the protocol. The appendices provide examples of SAs and key exchanges.

2.3.1 Basic ISAKMP Concepts

Domain of Interpretation The 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. The DOI defines:

- The set of information that will be used to determine the required security services (this information is called a situation).
- The set of security policies that must be supported.
- Syntax rules for the specification of proposed security services. A set of security services is called a protection suite.
- A common scheme for identifying cryptographic mechansisms, including encryption algorithms, key exchange algorithms, and certificate authorities.
- A naming scheme for the cryptographic algorithms supported within the domain, and for common Key Exchange Identifiers.

Specifications of the rules for individual DOIs will be presented in separate documents. The rules for the Internet Security DOI is contained in Appendix A.

A system may support multiple Domains of Interpretation. All systems MUST support the Internet Security DOI.

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.

Protection Suite A protection suite is a list of the security services that must be applied at 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.

2.3.2 ISAKMP Header Format

ISAKMP has a fixed header format (shown in Figure 2) followed by a variable length payload, optional digital signature, and optional padding. 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 prevent attacks (e.g. denial of service and replay). Based on the message type, each header is followed by a payload specific to the message type. The payload for each message is defined in sections 3 through 6. Following the payload portion of the ISAKMP packet is a digital signature. This field is dependent on the negotiation of Security Association attributes and may not be present.

```
                     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                        ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Message Type  ! Exch  ! Vers  !          Length               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                 Security Parameter Index (SPI)                !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                        Auxillary (SPI)                        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: ISAKMP Header Format

- Message Type (1 octet) - Indicates the type of message. A suffix of REQ denotes a Request message type and an RESP suffix denotes a *Response* message type. The format and processing for each message is defined in sections 3 through 6.

ISAKMP Message	Message Type
RESERVED	0
ISA_INIT_REQ	1
ISA_INIT_RESP	2
ISA_KE_REQ	3
ISA_KE_RESP	4
ISA_AUTH_REQ	5
ISA_AUTH_RESP	6
ISA_AUTH&KE_REQ	7
ISA_AUTH&KE_RESP	8
ISA_NEG_REQ	9
ISA_NEG_RESP	10
ISA_MODIFY_REQ	11
ISA_MODIFY_RESP	12
ISA_NOTIFY	13
ISA_DELETE	14
ISA_NEW_GROUP_REQ	15
ISA_NEW_GROUP_RESP	16
IANA Use	17-127
Future Use	128-255

- Exchange (4 bits) - indicates the type of exchange, see section 2.2 for a description of the Message Types exchanged in each of these Exchange Types.

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

- Version (4 bits) - indicates the version of the ISAKMP protocol in use.
- Length (2 octets) - Length of total message (header + payload) in octets.
- SPI (4 octets) - Security Parameter Index. The receiving entity's SPI is always in this field, except for the ISA_INIT packets. The use of the SPI field is described in Section 2.3.3
- Auxillary SPI (4 octets) - The use of the Auxiliary SPI field is described in 2.3.3
- Initiator Cookie (8 octets) - Cookie of entity that initiated SA establishment, SA modify or SA delete.
- Responder Cookie (8 octets) - Cookie of entity that is responding to an SA establishment, SA modify or SA delete request.

2.3.3 SPI Usage

While bootstrapping secure channels between systems, ISAKMP cannot assume the existence of security services, and must provide some protections for itself. Therefore, ISAKMP distinguishes two different types of SPIs. The first type of SPI, called a negotiation SPI, refers to a "local" security association, implemented by the ISAKMP service itself. The second type is called a protection SPI, and is used to refer to the SA being developed on behalf of other security protocols. Negotiation SPIs are meaningless outside of the negotiation server, while protection SPIs will be used by protocols such as AH and ESP.

Although SPIs are classified two different ways, all SPIs must be selected from the same SPI-space, so that the ISAKMP service can uniquely identify an SA based on a SPI.

In general, the SPI field in the ISAKMP header contains the receiving entity's negotiation SPI. The only exception to this is the ISA_INIT_REQUEST message, because the receiver has not yet established a reciving SPI for the session. In the ISA_INIT_REQUEST message, the the SPI field contains the SPI that the sender will be using for the session.

The Auxiliary SPI field is necessary because ISAKMP needs both a handle on the internal "negotiation SA", in order to protect or unprotect messages from ISAKMP peers, as well as a handle for the protection SA that is being developed.

The following table describes the contents of the two SPI fields for each of the message types:

ISAKMP Message	SPI	Auxiliary SPI
ISA_INIT_REQ	REQ NEG SPI	0
ISA_INIT_RESP	REQ NEG SPI	REC NEG SPI
ISA_KE_REQ	REC NEG SPI	REQ SPI
ISA_KE_RESP	REQ NEG SPI	REC SPI
ISA_AUTH_REQ	REC NEG SPI	REQ NEG SPI
ISA_AUTH_RESP	REQ NEG SPI	REC NEG SPI
ISA_AUTH&KE_REQ	REC NEG SPI	REQ NEG SPI
ISA_AUTH&KE_RESP	REQ NEG SPI	REC NEG SPI
ISA_NEG_REQ	REC NEG SPI	REQ PROT SPI
ISA_NEG_RESP	REC NEG SPI	REC PROT SPI
ISA_MODIFY_REQ	REC NEG SPI	REC SPI
ISA_MODIFY_RESP	REQ NEG SPI	REQ SPI
ISA_NOTIFY	REC NEG SPI	REC SPI
ISA_DELETE	REQ NEG SPI	REQ SPI
ISA_NEW_GROUP_REQ	REC NEG SPI	0
ISA_NEW_GROUP_RESP	REQ NEG SPI	0

Notes:

REQ NEG SPI = Requestor's negotiation SPI
REC NEG SPI = Receiver's negotiation SPI
REQ PROT SPI = Requestor's protection SPI
REC PROT SPI = Receiver's protection SPI
REQ SPI = Requestor's SPI (either negotiation or protection)
REC SPI = Receiver's SPI (either negotiation or protection)

For KE messages: if the messages are establishing keys for a negotiation session, the SPI is a negotiation SPI. Otherwise, the Auxiliary SPI is a protection SPI.

For MODIFY, NOTIFY, and DELETE messages: the Auxiliary SPI can refer to either type of SPI.

2.3.4 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 packet, the transmitting entity (initiator or responder) does the following:

1. Sets a timer and initializes a retry counter.
2. If the timer expires, the ISAKMP packet 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.

When an ISAKMP packet is received, the receiving entity (initiator or responder) does 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 Message Type field to confirm it is valid. If the Message Type field validation fails, the message is discarded and the following actions are taken:
   (a) The event, *INVALID MESSAGE 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.
3. Check the Exchange field to confirm it is valid for the Message Type requested. If the Exchange 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.
4. Check SPI to ensure it is valid for the Message Type and Exchange being performed. 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.

5. The message payload is processed. Individual message processing is described in sections 3 through 6. Depending on the Message Type, a valid message results in a response being sent to the transmitting entity (message originator). The procedures for sending these responses are also outline in sections 3 through 6.

2.3.5 Transport Protocol

ISAKMP can be implemented over any transport protocol or IP itself. The User Datagram Protocol (UDP) is minimum requirement for interoperability. The ISAKMP well-known port is TBD.

2.3.6 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.

2.3.7 Anti-Clogging Token ("Cookie") Creation

Phil Karn's Internet Draft [Karn] states that cookie generation is implementation dependent, but must satisfy some basic requirements:

```
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. The following sections, 3.1 to 3.3, describe the three basic phases that comprise SA Establishment: SA Initialization, Key and Authentication information exchange, and SA Negotiation.

3.1 Security Association Initialization

The initialization exchange of SA establishment is composed of the ISA_INIT_REQ and ISA_INIT_RESP packets shown in figure 3. The ISA_INIT packets exchange "cookies", and options for a key generation technique, an encryption algorithm and an authentication mechanism. The "cookies" are used to prevent replay and denial of service attacks. Authentication and encryption for the ISAKMP exchanges are provided by the authentication mechanism and encryption algorithm selected. The key generation technique selected creates keys for use by the authentication mechanism and encryption algorithm. The keys may also be used as any of the following: actual session keys, to create the session keys, or to protect the exchange of the actual session keys for the SA. If the key, encryption algorithm, and authentication mechanism are only used to protect ISAKMP exchanges, then new options can be picked during the negotiation phase (described in Section 3.3) for use in protecting the actual data communications. If encryption is not required for the SA, the encryption algorithm options are not exchanged.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          ISAKMP Header                        ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !  Payload Len  !           RESERVED            !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Domain of Interpretation                  !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                           Situation                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                           Proposal                            ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: ISA_INIT_REQ and ISA_INIT_RESP Packet Format

- ISAKMP Header - Described in Section 2.3.2
- Next Payload (1 octet) - Identifies the next payload in an ISAKMP message if more then one is carried in a message.
- Payload Length (1 octet) - Specifies the payload length in 4-octet units.

- Situation - Variable length field containing the situation for an SA (described in section 2.3.1).
- Proposal - Variable length field containing a list of proposed protection suites for an SA (described in section 2.3.1).

The format and content of both the situation and proposal is DOI-specific. The format of the Internet Security situation and proposal is described in Appendix A.

3.1.1 SA Initialization Procedures

When issuing an ISA_INIT_REQ message, the initiating entity does the following:

1. Create initiator cookie. See Section 2.3.7 for details.
2. Generate a unique pseudo-random negotiation SPI. See Section 2.3.2 for details.
3. Determine the relevant security characteristics of the session (the situation).
4. Generate a proposal for protecting a session under that situation.
5. Construct an ISA_INIT_REQ packet.
6. Transmit the packet to the destination host as described in Section 2.3.4.

When an ISA_INIT_REQ message is received, the receiving entity does the following:

1. Check the ISAKMP header as described in Section 2.3.4.
2. Unpack the ISA_INIT_REQ payload.
3. Determine if the given situation can be protected. If not, the protocol machine must send a rejection notification and return to IDLE.
4. Determine if it can use any of the proposed protection suites to protect the session. If none of the proposed suites are acceptable, then the protocol machine must send a rejection notification, clear all state and return to IDLE.
5. Create responder cookie. See Section 2.3.7 for details.
6. Generate a unique pseudo-random SPI. See Section 2.3.2 for details.
7. Construct an ISA_INIT_RESP packet containing the situation and the chosen protection suite.
8. Transmit the packet to the initiating host as described in Section 2.3.4.

When an ISA_INIT_RESP message is received, the receiving entity (original initiator) does the following:

1. Check the ISAKMP header as described in Section 2.3.4.
2. Unpack the ISA_INIT_RESP payload.
3. Determine that the situation returned is the same as the one sent. If not, the protocol machine must send a rejection notification and possibly resend the ISA_INIT_REQ message.

4. Determine if the returned protection suite is among the set of valid choices. If the entire proposal was rejected, the event PROPOSAL_REJECTED is logged to the appropriate audit file. If an invalid protection suite was returned, the receiving entity does the following:

   (a) The event, *INVALID ATTRIBUTES*, is logged in the appropriate system audit file.
   (b) Clear all state and return to IDLE. Any further communication must start the SA initialization procedures from the beginning.

   If a valid protection suite was selected, the receiving entity does the following:

   (a) Configure protocol machine based on protection suite selected.
   (b) Transition to Authentication and Key Exchange (see Section 3.2).

3.2 Authentication and Key Exchange

During the authentication and key exchange phase, information required to confirm the identities of the parties wishing to establish the SA and establish session keys for use during the SA establishment is exchanged. Depending on the key exchange algorithms, the original key may be used during data communications or a new one may be created and exchanged during the negotiation phase (described in section 3.3). This original or new key would be used in protecting the actual data communications.

The packets that carry the authentication and key exchange payloads have the format shown in Figure 4. When the ISA_AUTH&KE_REQ and ISA_AUTH&KE_RESP packets are used, the Authentication Payload SHOULD be processed first to strongly authenticate the packet issuer, followed by the processing of the Key Exchange Payload. The authentication and key exchange payloads (shown in Figures 5 and 6) are general formats which support many types of authentication and key exchange mechanisms. The detailed specification of these fields will be specified in companion RFCs. These companion RFCs will define the standard authentication and key exchange mechanisms that need to be implemented to assure compliance with this specification. The format for the Internet Security DOI key exchange and authentication payloads is described in A

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                         ISAKMP Header                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !  Payload Len  !            RESERVED           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
!                    Authentication Payload                     !
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !  Payload Len  !            RESERVED           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
!                     Key Exchange Payload                      !
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: ISA_AUTH&KE_REQ and ISA_AUTH&KE_RESP Packet Format

3.2.1 Authentication Payload Format

This section specifies the encoding of the authentication payload for the ISA_AUTH_REQ, ISA_AUTH_RESP, ISA_AUTH&KE_REQ, and ISA_AUTH&KE_RESP messages. As described in section 2.2.3, when the ISA_AUTH_REQ and ISA_AUTH_RESP packets are transmitted alone, the key exchange payload is not present. The format of these messages is shown in Figure 5.

```
                     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  !  Payload Len  !           RESERVED            !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Authentication Authority     !          Reserved             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Authentication Type          !           Length              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
!                     Authentication Data                       !
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Authentication Payload Format

- Authentication 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.

- Authentication Type (2 octets) - This field indicates the authentication payload format. This field is used by authentication authorities that support more than one certificate type. The authentication types supported by an authentication authority must be defined in the RFC required for authentication authority registration. Examples are:

  – PKCS #7 certificates
  – PGP certificates
  – DNS Signed Keys
  – Kerberos Tokens
  – X.509 certificates

- Length (2 octets) - Length of the Authentication Data field in octets.

- Authentication Data (variable) - Actual authentication data. The type of certificate is indicated by the Authentication Type field.

3.2.2 Key Exchange Payload Format

A variety of key exchanges can be supported in the key exchange phase. Some examples of key exchanges which may be used in this protocol 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. This protocol will also support key exchanges that include key escrow or data recovery techniques, but does not mandate their use.

ISAKMP supports both public and private key generation techniques. Both types must register with IANA to obtain a Key Exchange Identifier (KEI). Before published public key exchanges can obtain a KEI, an RFC defining the key exchange payload format and key generation procedures MUST be submitted. Private key exchanges SHOULD be documented in an RFC when registering for a KEI.

The encoding of the key exchange payload is dependent on the specific key exchange and, therefore, is not specified in this Internet draft. Each key exchange must define the following information: (a) System parameters, (b) Key establishment algorithm, and (c) Key derivation procedure (dependent on key exchange type). See [Oakley] for an example of a key exchange that can be executed during the ISAKMP key exchange phase.

As described in section 2.2.2, when the ISA_KE_REQ and ISA_KE_RESP packets are transmitted alone, the authentication payload is not present. Once the key exchange is completed, then the authentication payload is sent separately using the format described in section 3.2.1

```
                     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  !  Payload Len  !           RESERVED            !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               KEI             !           Length              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
!                     Key Exchange Data                         !
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Key Exchange Payload Format

- KEI (2 octets) - Key Exchange Identifier
- Length (2 octets) - Length of payload in octets
- Key Exchange Data (variable) - Data (i.e. public values) required to create session key.

3.2.3 Authentication and Key Exchange Procedures

When issuing an ISA_AUTH&KE_REQ packet, the initiating entity will do the following:

1. Create the ISAKMP Header.
2. Create the authentication payload.
3. Create the key exchange payload based on KEI.
4. Construct an ISA_AUTH&KE_REQ packet.

5. Generate an authentication signature using the authentication attributes and options selected in the initialization phase.
6. Transmit the packet to the responding host as described in Section 2.3.4.

When an ISA_AUTH&KE_REQ packet is received, the receiving entity will do the following:

1. Check the ISAKMP header as described in Section 2.3.4.
2. Verify the initiator's signature. The ISA_AUTH&KE_REQ packet is processed and the calculated signature is compared to the signature contained in the ISA_AUTH&KE_REQ packet. If these signatures are not identical, the message is discarded and the following actions are taken:
   (a) The event, *INVALID SIGNATURE*, 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. Unpack the ISA_AUTH&KE_REQ packet.
4. Create the ISAKMP Header.
5. Create the authentication payload.
6. Create the key exchange payload based on KEI.
7. Construct an ISA_AUTH&KE_RESP packet.
8. Generate an authentication signature, to authenticate responder to initiator, using the authentication attributes and options selected.
9. Transmit the packet to the initiating host as described in Section 2.3.4.
10. Begin key calculation in the background, if necessary.

When an ISA_AUTH&KE_RESP message is received, the receiving entity (original initiator) will do the following:

1. Check the ISAKMP header as described in Section 2.3.4.
2. Verify the initiator's signature. The ISA_AUTH&KE_RESP packet is processed and the calculated signature is compared to the signature contained in the ISA_AUTH&KE_RESP packet. If these signatures are not identical, the message is discarded and the following actions are taken:
   (a) The event, *INVALID SIGNATURE*, 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. Calculate key, if necessary.
4. Transition to Security Association Negotiation.

3.3 Security Association Negotiation

The SA Negotiation phase allows the initiating entity to present SA attributes that it wishes to use for secure communications to a responding entity. These SA attributes may include additional options for the attributes agreed upon during the initialization phase, as well as additional attributes required for an SA. As an example, the SA parameters for the IP AH and IP ESP security mechanisms are cited in the Security Architecture for the Internet Protocol [RFC-1825]. The format for the ISA_NEG_REQ and ISA_NEG_RESP packets is the same as the ISA_INIT_REQ and ISA_INIT_RESP shown in Figure 3. All fields shown in Figure 3 exist for the ISA_NEG_REQ and ISA_NEG_RESP packets.

3.3.1 SA Negotiation Procedures

When issuing an ISA_NEG_REQ packet, the initiating entity does the following:

1. Determine SA attributes to be negotiated. This may include changing some attributes from the original SA initialization.
2. Construct an ISA_NEG_REQ packet.
3. Depending on the SA Attributes established in the SA initialization phase, apply the agreed upon security services.
   (a) If the SA requires authentication, the ISA_NEG_REQ packet is processed (or signed) and the signature placed as noted in Figure 2.
   (b) If the SA requires encryption and the encryption algorithm is a block encryption algorithm, then padding up to the block size MUST be placed as noted in Figure 2.
   (c) If the SA requires encryption, the ISA_NEG_REQ payload and Signature are encrypted.
4. Transmit the packet to the responding host as described in Section 2.3.4.

When an ISA_NEG_REQ packet is received, the receiving entity does the following:

1. Check the ISAKMP header as described in Section 2.3.4.
2. Depending on the SA Attributes, apply the agreed upon security services.
   (a) If the SA requires encryption, decrypt the ISA_NEG_REQ payload and Signature. If the decryption fails, the message is discarded and the following actions are taken:
      i. The event, *DECRYPTION FAILED*, is logged in the appropriate system audit file.
      ii. 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.
   (b) If the SA requires authentication, the ISA_NEG_REQ packet is processed and the calculated signature is compared to the signature contained in the ISA_NEG_REQ packet. If these signatures are not identical, the message is discarded and the following actions are taken:
      i. The event, *INVALID SIGNATURE*, is logged in the appropriate system audit file.
      ii. 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. Unpack the ISA_NEG_REQ packet payload and determine the highest priority SA attributes supported. If none of the SA attribute options are supported, the ISA_NEG_RESP packet will have the value zero (0) in the Number of Sets field and an SA will not be established.
4. If the SA negotiation is requesting a key change or new authentication mechanism, then generate the appropriate information and include it as an attribute in the ISA_NEG_RESP payload.

5. Construct an ISA_NEG_RESP packet.
6. Depending on the SA Attributes, apply the agreed upon security services.
   (a) If the SA requires authentication, the ISA_NEG_RESP packet is processed and the signature placed as noted in Figure 2.
   (b) If the SA requires encryption and the encryption algorithm is a block encryption algorithm, then padding up to the block size MUST be placed as noted in Figure 2.
   (c) If the SA requires encryption, the ISA_NEG_RESP payload and Signature are encrypted.
7. Transmit the packet to the initiating host as described in Section 2.3.4.
8. If required, begin calculation of the new session key in the background.
9. Return appropriate data (i.e. SA, SPI) to negotiation server, clear all state, and return to IDLE.

When an ISA_NEG_RESP message is received, the receiving entity (original initiator) does the following:

1. Check the ISAKMP header as described in Section 2.3.4.
2. Depending on the SA Attributes, apply the agreed upon security services.
   (a) If the SA requires encryption, decrypt the ISA_NEG_RESP payload and Signature. If the decryption fails, the message is discarded and the following actions are taken:
      i. The event, *DECRYPTION FAILED*, is logged in the appropriate system audit file.
      ii. 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.
   (b) If the SA requires authentication, the ISA_NEG_RESP packet is processed and the calculated signature is compared to the signature contained in the ISA_NEG_RESP packet. If these signatures are not identical, the message is discarded and the following actions are taken:
      i. The event, *INVALID SIGNATURE*, is logged in the appropriate system audit file.
      ii. 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. Unpack the ISA_NEG_RESP payload and verify the SA attributes selected by responder are valid. If the attribute sets (or lists) are invalid or the responder rejected all proposed attribute sets (or lists), the receiving entity does the following:
   (a) The event, *INVALID ATTRIBUTES*, is logged in the appropriate system audit file.
   (b) Clear all state and return to IDLE.

   If the attribute set (or list) is valid, the receiving entity does the following:

   (a) Configure the protocol machine based on the attribute set (or list) selected.
4. If required, begin calculation of the new session key in the background.
5. Return appropriate data (i.e. SA, SPI) to negotiation server, clear all state, and return to IDLE.

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 format for the ISA_MODIFY packet is the same as the ISA_INIT_REQ and ISA_INIT_RESP shown in Figure 3.

4.1 Modification Procedures

The procedure for exchanging information to modify an SA are similiar to the SA negotiation exchange. The details of SA modification will be described in this section as they are solidified during prototype development.

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 ISA_DELETE packet (shown in Figure 7) provides a controlled method of informing a peer entity that the initiating entity has deleted an SA(s). The ISA_DELETE packet allows for 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 an ISA_DELETE packet the SAs listed in the SPIs field of the packet cannot be used with the initiating entity. The SA Establishment procedure must be invoked to re-establish secure communications.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          ISAKMP Header                        ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!          SPI Count            !             Length            !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
!                               SPIs                            !
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: SA Delete Payload Format

- SPI Count - Number of security associations to be deleted

- Length - length of payload in octets
- SPIs - Initiator's Security Parameter Index(s) to be deleted

5.1 Deletion Procedures

When issuing an ISA_DELETE packet, the issuing entity (initiator or responder) does the following:

1. Create initiator cookie. See Section 2.3.7 for details.
2. Determine SPI of receiving entity.
3. Construct the ISA_DELETE packet.
4. Depending on the SA Attributes, apply the agreed upon security services.
   (a) If the SA requires authentication, the ISA_DELETE packet is processed and the signature placed as noted in Figure 2.
   (b) If the SA requires encryption, the ISA_DELETE payload and Signature are encrypted.
5. Transmit the packet to the destination host as described in Section 2.3.4.
6. Update the local SA database to reflect the SPI deletions.

Upon receipt of an ISA_DELETE packet, the receiving entity (initiator or responder) does the following:

1. Check the ISAKMP header as described in Section 2.3.4.
2. Depending on the SA Attributes, apply the agreed upon security services in the following order.
   (a) If the SA requires encryption, decrypt the ISA_DELETE payload and Signature. If the decryption fails, the message is discarded and the following actions are taken:
      i. The event is logged in the appropriate system audit file.
      ii. Because the ISA_DELETE packet 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 ISA_DELETE packet be checked to see if the originator was the communicating party. If so, then these SAs can be deleted from the local SA database. We also recommend that an ISA_NOTIFY packet with an Error Message Type (see Section 6) be sent to the originator of the ISA_DELETE packet. If the SPIs do not match those of the originator, then no further action should be taken.
   (b) If the SA requires authentication, the ISA_DELETE packet is processed and the calculated signature is compared to the signature contained in the ISA_DELETE packet. 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 ISA_DELETE packet 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 ISA_DELETE packet be checked to see if the originator was the communicating party. If so, then these SAs can be deleted from the local SA database. We also recommend that an ISA_NOTIFY packet with an Error Message Type (see Section 6) be sent to the originator of the ISA_DELETE packet. If the SPIs do not match those of the originator, then no further action should be taken.

3. Unpack the ISA_DELETE payload.
4. Update the local SA database to reflect the SPI deletions.

6 Notification Message

The ISAKMP ISA_NOTIFY packet contains information one party wants to send to another. 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 ISA_NOTIFY message to synchronize SA communication (see Appendix B.2). The ISA_NOTIFY packet is unidirectional.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          ISAKMP Header                        ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!      Notify Message Type      !            Length             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
!                          Notify Payload                       !
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: ISA NOTIFY Payload Format

- Notify Message Type (2 octets)

Notification	Notify Message Type
RESERVED	0
Error	1-16383
Reserved for Future Use	16384-32767
Status	32768-49151
DOI Specific	49152-65536

- Length (2 octets) - length of payload in octets
- Notify Payload (variable) - Value dependent on the Notify Message Type

6.1 Notify Message Types

Notify Messages - Errors Types

Errors	Value	Payload
DOI-NOT-SUPPORTED	1	
SITUATION-NOT-SUPPORTED	2	
INVALID-COOKIE	3	
INVALID-VERSION-NO	4	
INVALID-MESSAGE-TYPE	5	
INVALID-EXCHANGE-TYPE	6	
INVALID-SPI	7	
ATTRIBUTES-NOT-SUPPORTED	8	
NO-PROPOSAL-CHOOSEN	9	
BAD-PROPOSAL-SYNTAX	10	
ATTRIBUTES-NOT-SUPPORTED	11	
INVALID-SIGNATURE	12	
DECRYPTION-FAILED	13	

Notify Messages - Status Types

Status	Value	Payload
CONNECTED	32769	

6.2 Notification Procedures

When issuing an ISA_NOTIFY message, the issuing entity (initiator or responder) does the following:

1. Create initiator cookie. See Section 2.3.7 for details.
2. Determine SPI of receiving entity.
3. Construct ISA_NOTIFY packet.
4. Depending on the SA Attributes, apply the agreed upon security services.
   (a) If the SA requires authentication, the ISA_NOTIFY packet is processed and the signature placed as noted in Figure 2.
   (b) If the SA requires encryption, the ISA_NOTIFY payload and Signature are encrypted.
5. Transmit the packet to the destination host as described in Section 2.3.4.

Upon receipt of an ISA_NOTIFY message, the receiving entity (initiator or responder) does the following:

1. Check the ISAKMP header as described in Section 2.3.4.
2. Depending on the SA Attributes, apply the agreed upon security services in the following order.
   (a) If the SA requires encryption, decrypt the ISA_NOTIFY payload and Signature. If the decryption fails, the message is discarded and the following actions are taken:

   i. The event is logged in the appropriate system audit file.
   ii. Because the ISA_NOTIFY packet is a unidirectional message a retransmission will not be performed. The local security policy will dictate the procedures for continuing.

 (b) If the SA requires authentication, the ISA_NOTIFY packet is processed and the calculated signature is compared to the signature contained in the ISA_NOTIFY packet. 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 ISA_NOTIFY packet is a unidirectional message a retransmission will not be performed. The local security policy will dictate the procedures for continuing.

3. Unpack the ISA_NOTIFY payload.

4. Depending on the 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 IP Security DOI

The IP Security DOI Assigned Number for IPv4 is one (1). The situation for DOI 1 is an IPv4 address. The IP Security DOI Assigned Number for IPv6 is two (2). The situation for DOI 2 is an IPv6 address.

A.1 IP Security Proposal Formats

This section defines the IP Security syntax for SA proposals and security attributes. The SA proposals for a security protocol (i.e. ESP) are carried in an SA payload. The SA payload is sent in the following messages: ISA_INIT_REQ, ISA_INIT_RESP, ISA_NEG_REQ, ISA_NEG_RESP, ISA_MOD_REQ, and ISA_MOD_RESP. This syntax 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 do not require a modification to the protocol.

Figure 9 shows the SA proposal format which contains the SA attributes. There can be one or more SA attribute in each SA proposal. There can one or more SA proposals 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 an ISA_NOTIFY message.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Protocol #   !  Proposal #   !  Proposal Len !    RESERVED   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
+                                                               +
.                                                               .
.                         SA Attributes                         .
.                                                               .
+                                                               +
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: SA Proposal Format

- Protocol Number (1 octet) - Identifies the security protocol requiring the SA attributes proposed. Uses the same values as the IPv4 Protocol field [RFC-1700].
- Proposal Number (1 octet) - Unique proposal identifier for the given security protocol.
- Proposal Length (1 octet) - Specifies the proposal length in 4-octet units. Each IP Security proposal is an integer multiple of 4 octets long.
- SA Attributes - Variable length field containing the attributes for an SA.

Figure 10 shows the SA attribute format. The most significant bit of the Attribute Class defines a grouping of attributes within a proposal. The second most significant bit indicates whether the attribute is of type basic or variable percision integer (VPI). Negative responses, such as: UNKNOWN SA ATTRIBUTE, are returned in an ISA_NOTIFY message.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!S!T!                         !  TYP=0 VPI Length              !
!E!Y!     Attribute Class     !  TYP=1 SA Attribute Value      !
!T!P!                         !                                !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                      TYP=0 VPI Present                        .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Attribute Format

- Attribute Class (2 octets) - Unique identifier for each general class of attribute type. ENCRYPTION ALGORITHM is an example of an attribute class. (See A.4 for the assigned attribute class values for ESP, AH, and Oakley.)

The most significant bit (SET) of the Attribute Class is for indicating a grouping of attributes within a proposal. If the SET bit is one (1) the following attribute belongs with the current attribute. There can be two or more attributes in a group. If the SET bit is zero (0) either the attribute is the last in a set or is an individual attribute. Attributes should be grouped together when a security policy decision must be made based on how attributes relate to each other, in addition to individual meaning.

The second most significant bit (TYP) of the Attribute Class is for indicating whether the attribute is a basic type or a variable percision integer (VPI). If the TYP bit is a zero (0) then the attribute is a VPI type. If the TYP bit is a one (1) then the attribute is a basic type.

Figure 11 shows the basic SA attribute format and Figure 12 shows the VPI SA attribute format.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!S!1!     Attribute Class     !      SA Attribute Value        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: Basic Attribute Format

- Value (2 octets) - The value of the SA attribute as defined by the Attribute class. (See A.5 for the assigned attribute values for IP Security.)

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!S!0!     Attribute Class       !          VPI Length           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                              VPI                              .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: VPI Attribute Format

- VPI Length (2 octets) - Specifies the VPI's length in 4-octet units. Each VPI is an integer multiple of 4 octets long.
- VPI - Variable Percision Integer. The field is aligned so the most significant bit is in the first 4-octet word following the VPI Length.

A.2 ESP SA and AH SA Proposals

The ESP and AH SAs are defined in [RFC-1825]. This section defines the format for the ESP and AH SA proposals. The attribute class fields are as they would appear in an ESP or AH SA Proposal. The attribute value and VPI fields contain examples of the information they would contain.

Note: The Lifetime fields (Key and SA) can be either basic or VPI attributes. Therefore when parsing the Attribute Class, the TYP bit must always be checked.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!      AH       !  Proposal #   !  Proposal Len !    RESERVED   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!1!1!    Authentication Alg     !             MD5               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!    Authentication Mode    !            KEYED              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!      Auth Key Exch Id     !    Oakley New Group Mode      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!0!        Key Lifetime       !                              1!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       Time (in seconds)                       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!0!         SA Lifetime       !                              1!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       Time (in seconds)                       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!0!   IP Source Address(es)   !                              1!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                        IPv4 Address                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!      Sensitivity Level    !           SECRET              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13: AH Proposal Format

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!      ESP      !  Proposal #   !  Proposal Len !    RESERVED   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!1!1!    Encryption Algorithm   !              DES              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!       Encryption Mode     !              CBC              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!    Encryption Transform   !            RFC-1828           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!       Enc Key Exch Id     !  Oakley EXTERNAL KEY MODE     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!0!    Crypotgraphic Synch    !             Length            !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                              MPI                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!     Replay Protection     !       Present / Absent        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!1!1!     Authentication Alg    !              MD5              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!     Authentication Mode   !             KEYED             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!      Auth Key Exch Id     !  Oakley PRIVATE GROUP MODE    !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!        Key Lifetime       !         Time (in seconds)     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!0!        SA Lifetime        !                              1!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       Time (in seconds)                       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!0!   IP Source Address(es)   !                              4!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                        IPv6 Address                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!     Sensitivity Level     !            SECRET             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: ESP Proposal Format

A.3 Oakley Proposal

The Oakley proposal format contains the SA attributes that are exchanged in the ISA_INIT messages in order to establish the required security attributes for the key and authentication exchange. See [Oakley] for further details.

Note: The three figures 15, 16, and 17 are all combine to make one proposal. They are shown seperately for reading and formatting ease.

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!    Oakley     !  Proposal #   !  Proposal Len !    RESERVED   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          EHA  Format                          !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                         Group Format                          !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15: Oakley Proposal Format

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!     Auth / Priv  Flag     !            PRIV               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!   Encryption Algorithm    !             DES               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!       Hash Algorithm      !             MD5               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!1!1!    Authentication Alg     !             RSA               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!1!    Authentication Mode    !            KEYED              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16: Oakley Proposal - EHA Format

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!1!1!     Group Description     !             MODP              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!1!0!        Field Size         !            Length             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                              MPI                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!1!0!          Prime            !            Length             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                              MPI                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!1!0!       Generator1          !            Length             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                              MPI                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!1!0!       Generator2          !            Length             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                              MPI                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!1!0!         Curve-p1          !            Length             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                              MPI                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!1!0!         Curve-p2          !            Length             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                              MPI                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!1!0!   Largest Prime Factor    !            Length             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                              MPI                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!1!0!      Order of Group       !            Length             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                              MPI                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!0!0!    Strength of Group      !            Length             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                              MPI                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 17: Oakley Proposal - Group Format

A.4 Attribute Class Assigned Numbers

Values for attribute classes are specified in the most recent "Assigned Numbers" RFC [RFC-1700]. Presented in the following tables are the values for ESP, AH, and Oakley SAs. In the Attribute Type Column, a "B" means basic encoding and "V" mean Variable Percision Integer.

AH and ESP Attribute Classes

Class	Assigned Value	Attribute Type
RESERVED	0	x
RESERVED	1	x
Authentication Algorithm	2	B
Authentication Mode	3	B
Authentication KEI(s)	4	B
Encryption Algorithm	5	B
Encryption Mode	6	B
Encryption Transform	7	B
Encyption KEI(s)	8	B
Size of cryptographic synchronization or IV	9	B/V
Replay Protection	10	B
Key Lifetime	11	B/V
Rekey Value	12	B/V
SA Lifetime	13	B/V
IP Source Address(es)	14	V
Sensitivity Level	15	B

Oakley Attributes Classes

Class	Assigned Value	Attribute Type
Auth / Private Flag	16	B
Hash Algorithm	17	B
Group Description	18	B
Group Type	19	B
Field Element Size	20	V
Print (P) or Irreducible Field Polynomial	21	V
Generator (1 or 2 values)	22	V
Curve Parameters (2 values)	23	V
Largest Prime Factor of the Group Size	24	V
Order of the Group	25	V
Strength of Group	26	V

Attribute class values 27-1024 are reserved for IANA Use. Attribute class values 1025-15360 are reserved for future use. Attribute class values 15360-16384 are reserved for private use.

A.5 Attribute Value Assigned Numbers

A.5.1 Sensitivity Level Assigned Numbers

Sensitivity Level

Level	Assigned Value
Not In Use	0
Unclassified	1
FOUO	2
Undefined	3
Confidential	4
Secret	5
Top Secret	6

Sensitivity values 7-1024 are reserved for IANA Use. Values 1025-15360 are reserved for future use. Values 15360-16384 are reserved for private use.

A.5.2 Key Exchange Identifiers (KEI) Assigned Numbers

Key Exchange Identifiers (KEI)

Key Exchange	Assigned Value
Reserved	0
Oakley Main Mode	1
Oakley ISAKMP Mode	2
Oakley Quick Mode	3
Oakley External Mode	4

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

A.5.3 Encryption Transform Assigned Numbers

Encryption Transforms

Transform	Assigned Value
Reserved	0
RFC-1829	1
DES-CBC w/Replay	2

Encryption Transform values 3-1024 are reserved for IANA Use. Values 1025-15360 are reserved for future use. Values 15360-16384 are reserved for private use.

B ISAKMP Scenarios

Examples scenerios are are presented to help illustrate the ISAKMP's ability to support multiple authentication methods and key exchanges.

B.1 Oakley Scenario

		Oakley Scenario		
Entity #1	NS #1	INTERNET	NS #2	Entity #2
		Establish Initial SA Between NSs		
		ISA_INIT_REQ		
		============>		
		ISA_INIT_RESP		
		<============		
		Oakley Key Exchange Between NSs		
		ISA_KE_REQ		
		===============>		
		ISA_KE_RESP		
		<===============		
		Oakley Authentication Exchange		
		ISA_AUTH_REQ		
		===============>		
		ISA_AUTH_RESP		
		<===============		
		ISA_AUTH_REQ		
		===============>		
		ISA_AUTH_RESP		
		<===============		
		Protected Traffic NS#1 to NS#2		

Oakley Scenario continued

```
Entity #1   NS #1            INTERNET                 NS #2   Entity #2

                     SA Established NS#1 to NS#2

                     Establish SA Between Entities

                            ISA_NEG_REQ
                          ============>
                            ISA_NEG_RESP
                          <============

                     Oakley External Key Exchange
                           Between Entities

                            ISA_KE_REQ
                        ==============>
                            ISA_KE_RESP
                       <===============
                            ISA_KE_REQ
                        ==============>


                          Protected Traffic
                         Entity#1 to Entity#2
                       <==============>
```

The diagrams above only shows ISAKMP messages exchanges. Shown are the exchanges to initiate SAs between entities and negotiation servers and the exchanges for the Oakley key exchange and authentication. The formats and contents of the messages can be found in [Oakley] and Appendix A. See Section 2.1 for the relationship of ISAKMP to the protocol stack.

When an entity, which can be a process, application, security protocol, etc., wishes to establish communications with a peer entity a call is made to the negotiation server (NS). NS#1 checks the local security policy to determine if an SA is required. If an SA is required, then NS#1 checks if it has the appropriate SAs established with the peer NS (NS#2). If a negotiation SA (NS-to-NS SA) is exists, NS#1 can proceed to the start of the second diagram. If a negotiation SA needs to be established, the NSs exchange ISA_INIT messages to determine the security attributes, key exchange, and authentication to be used for the negotiation SA. In our example the Oakley key exchange and authentication is choosen. The ISA_KE and ISA_AUTH messages are exchanged according to the rules defined in the key exchange. Oakley requires two key exchange messages and four authentication messages. Once these exchanges are complete a negotiation SA between NSs is established. In the second diagram the negotiation SA is used to protect the remaining exchanges shown. The NSs now exchange ISA_NEG messages to create a SA for the entity itself. In our example an Oakley External Key Exchange is now performed to establish a new key for the entity to entity SA. Once this SA is established, protected communications takes place.

B.2 Virtual Private Network Scenario

This scenario shows how ISAKMP can be used in a Virtual Public Network (VPN). The ability to establish SAs for more than just ESP and AH and one of the uses of the ISA_NOTIFY message are also illustrated.

Virtual Public Network Scenario

End System #1	SFE #1	INTERNET	SFE #2	End System #2
Establish ES#1 To SFE#1 Connection				
SYN				
===>		Establish Connection Between SFEs		
		SYN		
		===>		
		SYN, ACK		
		<=======		
		ACK		
		===>		
		Establish SA Between SFEs		
		ISA_INIT_REQ		
		============>		
		ISA_INIT_RESP		
		<============		
		ISA_KE&AUTH_REQ		
		==============>		
		ISA_KE&AUTH_RESP		
		<===============		
		Secure Connection Between SFEs		Establish SFE#2 to ES#2 Connection
				SYN
				===>
				SYN, ACK
				<=======
				ACK
				===>
		ISA_NOTIFY(Status == Connected)		
		<====================		
SYN, ACK				
<=======				
ACK				
===>				
		Protected Traffic ES#1 to ES#2		
		<==============>		

The diagram shows an End System (ES) using a connection oriented protocol (we use TCP as an example) establishing a connection with another ES. Both ES are behind Secure Front Ends (SFE) (e.g. firewalls). The connection establishment from End System #1 (ES#1) is intercepted by its Secure Front End (SFE #1). SFE#1 establishes a connection and then a Security Association (SA), using normal ISAKMP SA establishment procedures, with SFE #2. Next SFE #2 establishes a connection with ES #2. Upon successful completion SFE #2 sends an ISA_NOTIFY with Status equal Connected. SFE #1 completes it's connection with ES #1 and normal end to end communications takes place secured between SFE #1 and SFE #2. If SFE #2 had been unable to establish a connection with ES #2 it would have returned an ISA_NOTIFY with Status equal Not Connected with an optional reason code.

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 computational 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 soley rely on any UDP information (e.g. checksum, length) for its processing.

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

A number of very important security considerations are presented in [RFC-1825]. One bares 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 protect provided by the IP Security services.

Acknowledgements

Marsha Gross, Bill Kutz, Mike Oehler, Mark Schneider, 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 have contributed to the prototype development and integration with ESP and AH.

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, October 26, 1995.

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

[BC] Ballarie, 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.

[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.

[RFC-1212] McCloghrie K. and M. Rose, *Concise MIB Definitions*, RFC-1212, March 26, 1991.

[RFC-1213] McCloghrie K. and M. Rose, *Management Information Base for Network Management of TCP/IP-based Internets: MIB-II*, RFC-1213, March 26, 1991.

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

[RFC-1700] Reynolds, J. and J. Postel, *Assigned Numbers*, STD 2, RFC-1700, October, 1994.

[RFC-1155] Rose M. and K. McCloghrie, *Structure and Identification of Management Information for TCP/IP-based Internets*, RFC-1155, May, 1990.

[Secu] SECUREWARE INC., *Peer Authentication and Key Management Protocol Specification*, Version 2.2, October 27, 1995.

[Schneier] Bruce Schneier, *Applied Cryptography - Protocols, Algorithms, and Source Code in C*, John Wiley & Sons, Inc., 1994.

[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 two authors are with:

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

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

Mark Schertler
Phone: 301-688-0849
E-mail: mjs@tycho.ncsc.mil