Consultative Committee for Space Data Systems

RESEARCH AND DEVELOPMENT FOR SPACE DATA SYSTEM STANDARDS

Next Generation Space Internet (NGSI)—End-to-End Security for Space Mission Communications

CCSDS 733.5-O-1

EXPERIMENTAL SPECIFICATION

April 2003

AUTHORITY

Issue:	Current Issue
Date:	April 2003
Location:	Matera, Italy

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and reflects the consensus of technical panel experts from CCSDS Member Agencies. The procedure for review and authorization of CCSDS Reports is detailed in the *Procedures Manual for the Consultative Committee for Space Data Systems*.

This document is published and maintained by:

CCSDS Secretariat
Office of Space Communication (Code M-3)
National Aeronautics and Space Administration
Washington, DC 20546, USA

PREFACE

This document is a CCSDS Experimental Specification. Its Experimental status indicates that it is part of a research or development effort based on prospective requirements, and as such it is not considered a Standards Track document. Experimental Recommendations are intended to demonstrate technical feasibility in anticipation of a 'hard' requirement that has not yet emerged. Experimental work may be rapidly transferred onto the Standards Track should a hard requirement emerge in the future.

FOREWORD

This Experimental Specification describes the implementation of end-to-end security for space mission communications within the proposed Next Generation Space Internet (NGSI) architecture.

Through the process of normal evolution, it is expected that expansion, deletion, or modification to this Report may occur. This Experimental Specification is therefore subject to CCSDS document management and change control procedures which are defined in the *Procedures Manual for the Consultative Committee for Space Data Systems*. Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this report should be addressed to the CCSDS Secretariat at the address on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- National Aeronautics and Space Administration (NASA)/USA.
- National Space Development Agency of Japan (NASDA)/Japan.
- Russian Space Agency (RSA)/Russian Federation.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Communications Research Centre (CRC)/Canada.
- Communications Research Laboratory (CRL)/Japan.
- Danish Space Research Institute (DSRI)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Federal Service of Scientific, Technical & Cultural Affairs (FSST&CA)/Belgium.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space and Astronautical Science (ISAS)/Japan.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Oceanic & Atmospheric Administration (NOAA)/USA.
- National Space Program Office (NSPO)/Taipei.
- Space & Upper Atmosphere Research Commission/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 733.5-O-1	Next Generation Space Internet—End-to-End Security for Space Mission Communications	April 2003	Current Issue

CONTENTS

1 INTRODUCTION

1.1 PURPOSE

The Internet community has developed both a security protocol and a key management standard. The space community has developed a security protocol standard, but has yet to develop a key management standard. This Experimental Specification discusses the alternatives and draws conclusions regarding a key management standard for space communications. It also discusses a means by which the Internet security protocol and the space security protocol may be made to interoperate.

1.2 REFERENCES

The following documents are referenced in this Experimental Specification. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Experimental Specification are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS Recommendations.

[1] *Space Communications Protocol Specification (SCPS)—File Protocol (SCPS-FP).* Recommendation for Space Data System Standards, CCSDS 717.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1999.

[2] *Space Communications Protocol Specification (SCPS)—Transport Protocol (SCPS-TP).* Recommendation for Space Data System Standards, CCSDS 714.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1999.

[3] *Space Communications Protocol Specification (SCPS)—Security Protocol (SCPS-SP).* Recommendation for Space Data System Standards, CCSDS 713.5-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1999.

[4] *Space Communications Protocol Specification (SCPS)—Network Protocol (SCPS-NP).* Recommendation for Space Data System Standards, CCSDS 713.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1999.

[5] *SDNS Secure Data Network System Security Protocol 3 (SP3)*. SDN.301, revision 1.3, July 1988.

[6] Le, F., and Faccin, S. *Dynamic Diffie Hellman-based Key Distribution for Mobile IPv6.* draft-le-mobileip-dh-00.txt, April 2001.

[7] Mills, D., *Network Time Protocol Security Model and Authentication Scheme*. University of Delaware, May 2001.

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

[9] Caronni, G., Lubich, H., Aziz, A., Markson, T., and Skrenta, R. *SKIP—Securing the Internet*. http://www.skip-vpn.org/wet-ice.html

[10] Aziz, A., Markson, T., and Prafullchandra, H. *Simple Key-Management for Internet Protocols (SKIP)*. http://www.skip-vpn.org/spec/SKIP.htm

[11] Karn, P., and Simpson, W. *Photuris: Session-Key Management Protocol*. RFC 2522 (experimental), March 1999.

[12] Piper, D. *The Internet IP Security Domain of Interpretation for ISAKMP*. RFC 2407, November 1998.

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

2 OVERVIEW

2.1 BACKGROUND

The Internet Engineering Task Force (IETF) Internet Protocol (IP) Security (IPSEC) Working Group has developed a set of standard Internet security protocols. These standards provide confidentiality (encryption) and authentication (via the Encapsulating Security Payload [ESP] protocol) or only authentication (via the Authentication Header [AH] protocol). In addition, the IETF IPSEC working group has standardized a key management protocol, the Internet Key Exchange (IKE), which is based on two other IETF protocols, the Internet Security Association and Key Management Protocol (ISAKMP) and the Oakley Key Exchange Protocol.

The space community, via the Consultative Committee for Space Data Systems (CCSDS) has developed the Space Communications Protocol Specification (SCPS) series of protocol recommendations. The CCSDS Recommendations are also International Organization for Standardization (ISO) standards. The SCPS protocols consist of:

a) a file transfer protocol (reference [1]) that is fully interoperable with the Internet File Transfer Protocol (FTP);

b) a transport protocol (reference [2]) that is fully interoperable with the Internet Transmission Control Protocol (TCP);

c) a security protocol (reference [3]) that is based on the Internet IPSEC ESP but was designed for bit-efficiency and is therefore not directly IPSEC interoperable; and

d) a network protocol (reference [4]) that take the place of many different network layer protocols such as IP or the CCSDS path layer protocol.

From a security perspective, the SCPS Security Protocol (SCPS-SP) is a functional ‘cousin’ of the IPSEC ESP protocol. SCPS-SP is inspired by ESP, as well as by several of ESP’s predecessors, such as the US Department of Defense’s (DoD) ‘Security Protocol at Layer 3’ (reference [5]) (a security protocol developed by DoD for use at ISO Layer 3). However, IPSEC ESP is a ‘heavy-weight’ protocol. That is, ESP adds a minimum of 10 bytes of overhead to each IP packet. SCPS-SP was designed to provide a set of security services equivalent to IPSEC ESP, with less options, but in a much more bit-efficient manner. SCPS-SP adds only 2 bytes of overhead per IP packet. Because IPSEC ESP and SCPS-SP are different protocols, they are not directly interoperable. However, interoperability can be accomplished via a ‘trusted’ gateway, as will be discussed later in this document.

Key management is a major security component missing in the space communications community. It is assumed that space-community ground-based assets will utilize ground-based Internet protocol standards and their ground-based security equivalents (e.g., IPSEC and IKE). The space-based assets may use the same ground-based protocols (e.g., TCP/IP), or may elect to use the space-optimized versions of the Internet protocols (e.g., SCPS) to obtain better performance with less per-packet overhead. However, SCPS does not specify a

key management standard. One of the major goals of the Next Generation Space Internet (NGSI) project is to analyze the existing key management protocols and determine a course of action for the adoption of a standard by the space communications community.

2.2 IPSEC/SCPS-SP GATEWAY

As part of the development of the SCPS protocols, a 'reference implementation' of each protocol was produced. The reference implementation code was tested in the laboratory, in 'bent-pipe' tests over communications satellites, and in flight tests aboard the United Kingdom's (UK) Space Technology Research Vehicle (STRV).

As a side effect of the reference code development, a 'SCPS gateway' was developed which provides interoperability between the SCPS protocols and the ground-based Internet protocols. The 'SCPS side' of the gateway includes the SCPS network protocol (reference [4]), the security protocol (reference [3]), the transport protocol (reference [2]), and the file protocol (FP). On the 'Internet side' of the gateway, IP, TCP and FTP were implemented.

NOTE – In reality TCP and SCPS-TP are the same, since SCPS-TP is TCP with negotiated extensions to allow it to operate with better performance in space communications environments. Figure 2-1 illustrates the peer-to-peer layers of the SCPS gateway protocol stacks.

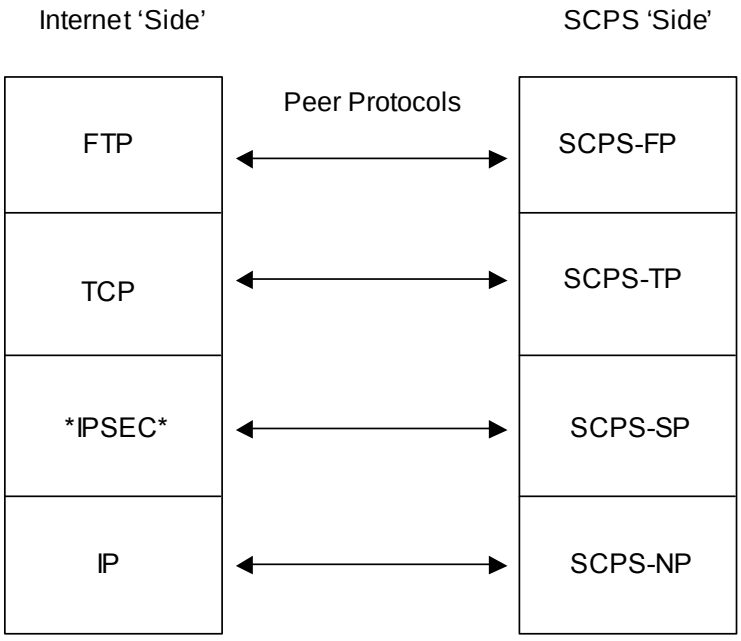

Figure 2-1: SCPS Gateway Protocol Stacks

The protocol originally missing from the Internet side of the gateway is the Internet IPSEC ESP protocol. As a result, the gateway was not capable of supporting secure interoperations if both end-systems were not using SCPS-SP. With the addition of IPSEC ESP to the Internet side of the gateway, secure interoperability between IPSEC ESP and SCPS-SP could be accomplished. IPSEC has been added successfully to the SCPS Gateway and interoperation has been demonstrated both in the lab and over the Internet.

It should be noted that when using a gateway to perform such secure interoperation, there is a momentary loss of end-to-end security. That is, the gateway becomes a trusted entity because it has to take the payload data received in one 'envelope' (e.g., a SCPS-NP packet) and put it into another envelope (e.g., an IP packet). However, in order to move the payload from SCPS-NP to IP when the payload data is encrypted by SCPS-SP at its origination point, it must first be decrypted on the SCPS side of the gateway by SCPS-SP and then re-encrypted using IPSEC ESP for forwarding onward as an IP packet to its final (IPSEC-aware) destination. As a result, the payload data is 'exposed' in a non-encrypted form in the gateway momentarily as it moves between protocol stacks. Therefore, the gateway must be 'trusted' to ensure that it does not inadvertently disclose any information to those who are not authorized to receive it. In some communities, this loss of end-to-end security may pose a problem, which can be solved by the use of application-level encryption such as the Secure Sockets Layer (SSL) or by the Internet's version of SSL called Transport Layer Security (TLS). In this way, the payload data would be encrypted on an application-to-application basis and then would also be encrypted by SCPS-SP and/or IPSEC ESP. Application layer security only protects the payload content and none of the protocol headers, whereas SCPS-SP and IPSEC protect the transport layer header as well.

In order to bring about IPSEC/SCPS-SP interoperability, the gateway must create two Security Associations (SA): one for use between an end-system and SCPS-SP, and one for use between an end-system and IPSEC ESP. A security association establishes the security parameters that will be used in the secure connection by the communicating entities. The gateway interoperation will require two security association negotiations, but both may use the same encryption keys if that is allowed by the security policy.

Such an interoperable gateway has been created by modifying the existing SCPS gateway to add IPSEC functionality. The SCPS gateway principally runs on a FreeBSD UNIX platform, but it could be run on just about any version of UNIX (including Linux) with the addition of 'divert sockets'. IPSEC implementations, available as open-source software, compliments of the KAME and FreeS/WAN project are available for both FreeBSD and Linux, respectively. In either case, the UNIX/Linux kernel supporting the gateway is built to include IPSEC. This gateway modification has already been tested locally in the laboratory as well as over the Internet in a collaborative effort between the US and the UK. Both have built and tested IPSEC-SCPS gateways providing an encrypted Virtual Private Network over the Internet.

2.3 KEY MANAGEMENT

When entities communicate securely, the network traffic transmitted between them is encrypted. Encryption is accomplished by the use of an encryption algorithm (which does not change) and a cryptographic key (which does change). It is assumed that a potential

adversary knows the intimate details about the encryption algorithm and its implementation. However, the cryptographic key(s) is assumed to be unknown and must be highly protected. This is quite similar to a door lock. It is assumed that the internal lock mechanism is well known and understood by burglars, but without the lock's unique key, only a brute force attack will open a door lock (e.g., a lock pick or a strong kick).

Key management is the mechanism by which communicating entities exchange and agree upon Traffic Encryption Keys (TEK), which are the keys used to encrypt traffic during a communications session.

As has been previously stated, the IETF has specified a standard key management protocol for the Internet (IKE). However, there are other key management systems in use and in development. For example, the financial community, under the auspices of the American National Standards Institute (ANSI), has developed a series of key management standards in their X9 working group. The current ANSI X9 symmetric key exchange standard has been withdrawn, and two other key exchanges are being developed in its place: one for Diffie Hellman public key, and one for elliptic curve public key.

Also, various IETF working groups are examining/specifying key exchange mechanisms specifically for their application environments because they perceive that IKE does not meet their needs. That is, for the most part, IKE has too much capability for their needs. The IETF IPSEC working group is also grappling revisions of IKE because it has been found to be complicated and provides too much flexibility for most Internet key management needs. The IETF is now entertaining suggestions for the next generation of IKE.

The problem is what key management standard should be adopted for use in space communications systems? Should the community adopt an existing standard and if so which one(s)? Alternatively, should the space community develop an entirely new key management standard because of its unique bandwidth-constrained requirements? The pros and cons of both alternatives, along with additional background information, will be examined in more detail in the remainder of this section.

2.4 ANSI KEY EXCHANGE STANDARDS

The ANSI Key Exchange standards that are currently under development have limited applicability outside of the financial community for which they are being developed. The ANSI key exchange documents discuss a limited subset of key management; only the key exchange mathematical functions are specified. Therefore, the ANSI specifications would only be suitable for use in the space community to perform key exchanges if a higher-level key management framework existed. Both IPSEC and SCPS-SP require a way to negotiate security associations in addition to exchanging cryptographic keys and, therefore, require more than what the financial community is developing.

2.5 OTHER IETF KEY EXCHANGE STANDARDS

Several IETF working groups (e.g., Mobile IPv6, Secure Network Time) have requirements for simple (and fast) key exchange mechanisms. They have a need to authenticate their

communication partners—and no more. They perceive that they do not need (or want) the negotiation capabilities of IKE, nor in some cases, can they 'afford' the overhead of security association negotiations. As a result, they have invented their own simple key exchange mechanisms based on public key exchanges.

For example, the Mobile IPv6 (reference [6]) working group is examining a low-overhead, bandwidth-efficient means of securely authenticating binding updates between mobile systems and correspondent nodes. The mobile node sends binding updates to a correspondent node to update its current location. The binding update results in a change of the routing from a home agent to a mobile node. Before accepting a new binding update, the correspondent node must authenticate its source. In this example strong authentication must be performed, but there is no need to negotiate other security association parameters as would be required for the establishment of an IPSEC security association. This demonstrates that the working group has a good reason for the development of an alternative and simple key exchange mechanism.

The Internet Network Time Protocol (NTP) (reference [7]) is used to synchronize time among various systems over a network. NTP sends time updates via the Internet, but this mechanism can be easily spoofed, particularly when the timeserver is operating in a broadcast mode. As a result, the IETF's Secure Network Time working group has a requirement for authentication of the timeserver to client systems. NTP has included authentication of timeservers for many years using a shared secret key mechanism. However, NTP has a critical need to use only very fast security mechanisms in order to not delay (and thereby skew) time update messages. NTP has the capability to measure the network latency and send out time update adjustments, taking into account transmission delays while still maintaining microsecond synchronization. Therefore, once NTP's time synchronization message is ready to be sent, any additional delays in transmission due to cryptographic operations are not taken into account and will result in inaccuracies. For example, an inaccurate time update will occur if the message is delayed while a security association is established, keys are exchanged, and the message is digitally signed to authenticate its source. As is the case with Mobile IPv6, Secure NTP requires only a relatively simple means of authenticating timeserver sources.

2.6 IKE/ISAKMP

The IETF's IPSEC working group developed two protocols to provide security at the network layer (layer 3): the Encapsulating Security Payload (ESP) and the Authentication Header (AH) protocols. Both protocol specifications require that conforming implementations shall be cryptographically keyed by either automated or manual means. For initial testing purposes, IPSEC was keyed using manual keying techniques. However, this was only a stopgap measure while awaiting an automated key management standard.

The Internet Security Association and Key Management Protocol (ISAKMP) (reference [8]) is a good negotiation protocol that supports the negotiation of security parameters and cryptographic keys to establish security associations between communicating entities. ISAKMP established the framework by which negotiations could be accomplished without specifying cryptographic algorithms or key exchange mechanisms. It was designed to be key

exchange independent/neutral. One of its competitors, the Simple Key Management for Internet Protocol (SKIP) (references [9] and [10]) did not meet the requirements set forth by the IPSEC working group because it was stateless and, therefore, required key management data to be transmitted with each IP packet. SKIP provided a means of quickly keying communicating entities since it did not require any out-of-band setup. However, it was costly because it resulted in additional per-packet overhead. Its other major competitor, Photuris (reference [11]), which was the leading IETF key management protocol candidate at one point, did not provide a general negotiation framework, as did ISAKMP. Photuris was much more specific in terms of the algorithms and modes allowed to be negotiated, whereas ISAKMP provided a more generalized framework that could be extended as necessary.

ISAKMP did not specify negotiation parameters, algorithms, etc. As a result, a Domain of Interpretation (DOI) (reference [12]) was written to define the specific protocol identifiers used during an ISAKMP negotiation. For example, ISAKMP does not specify cryptographic algorithm identifiers. Rather, these are defined in the Internet DOI (e.g., ESP using DES as a cryptographic algorithm is defined in reference [12] with the value 2).

In addition, since ISAKMP did not specify a key exchange, but the Oakley Key Determination Protocol (reference [13]) did, the main points of Oakley were incorporated with ISAKMP to form IKE. As a result, IKE is essentially a ‘profile’ for the use of ISAKMP with the Internet DOI and the Oakley key exchange.

3 SPACE COMMUNICATIONS KEY MANAGEMENT

3.1 OPTIONS

As evidenced by the material presented in section 2, a great deal of work has already been done in the areas of key management and key exchanges. Therefore, the space communications community could elect to:

a) use one of the existing key management protocols;

b) modify (profile) one of one of the existing protocols; or

c) develop a space-community unique protocol.

For space communications, a key exchange protocol as was defined by Oakley (and as is being defined by ANSI) could be used. Most modern key exchange protocols are concerned with the creation of traffic encryption keys using public key encryption techniques (e.g., Diffie Hellman) or via the use of signed digital certificates (e.g., X.509), which contain authenticated public keys. However, as the IETF determined, more than just the key exchange is needed to adequately service protocols such as IPSEC (and by the same extension, SCPS-SP). There is more information that must be negotiated and exchanged beyond the traffic encryption key(s) (e.g., algorithm, mode of operation, hash, key length).

Both the IPSEC protocols and SCPS-SP require the creation of security associations. A security association is the result of a negotiation between two parties who wish to communicate securely. Negotiation might be a bit too strong a word to use since in fact, under IKE, one party presents one or more 'proposals', one of which must be accepted in its entirety by the other party. For example, System A might send three proposals to System B: one using 3DES for encryption and Secure Hashing Algorithm-1 (SHA-1) for hashing; one using DES for encryption and SHA-1 for hashing; and one using DES for encryption and Message Digest 5 (MD5) for hashing. System B would have to accept one of the proposals with no modifications, or completely decline to establish the security association. See figure 3-1.

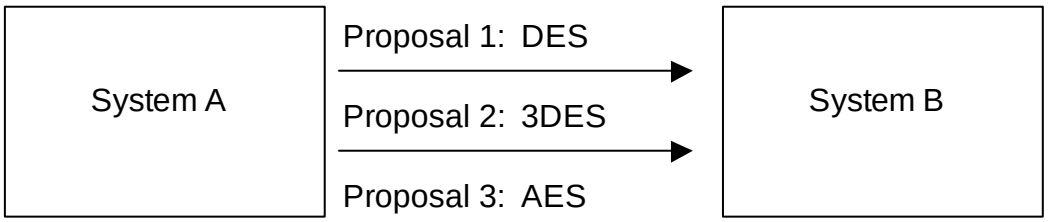

Figure 3-1: IKE Negotiation Proposals

There is a major difference between the requirements of IPSEC and SCSP-SP as compared with other protocols requiring key exchanges. IPSEC and SCPS-SP require the establishment of security associations, whereas the other protocols do not.

Therefore, it would appear to make the most sense for the space community to either adopt IKE as it presently exists, or develop a minimal profile for its use in a space communications environment while maintaining interoperability with the rest of the Internet ground infrastructure.

3.2 IKE ADOPTION BY THE SPACE COMMUNITY

In spite of its overhead, IKE appears to be the right answer for the space community. The means by which the overhead may be minimized will be discussed in this subsection.

IKE (and the base ISAKMP protocol) can operate in several phases/modes. (ISAKMP uses the term 'phases', whereas Oakley uses the term 'modes.') The most secure manner in which to operate IKE requires two phases:

a) a first phase (Phase I) to establish a secure channel between the communicating entities; and

b) a second phase (Phase II) to exchange security parameters used to establish security associations.

IKE does not assume that there are any other existing security mechanisms over which to perform a secure exchange of parameters (e.g., an existing IPSEC secure channel or an SSL secure channel). This is accomplished via the Phase I exchange using either 'main mode' or 'aggressive mode'. As a result there is an overhead cost to use IKE, because a secure channel must first be established before security association parameters may be exchanged.

However, the ISAKMP authors understood that there might be situations when it is imperative that a secure security association exchange occurs quickly. As a result, IKE/ISAKMP allows the establishment and caching of multiple security associations under a single ISAKMP secure channel.

In addition, to save both time and bandwidth, but at the loss of some security, IKE/ISAKMP defines an 'aggressive exchange.' Whereas main mode requires six message exchanges to establish a security association, aggressive mode requires only three. An aggressive mode exchange allows ISAKMP security associations, key exchanges, and authentication payloads to be transmitted together in a single ISAKMP message. This mode reduces the number of round-trips required to establish a security association and key exchange. This is good for typical bandwidth-constrained space communications, but the reduction in overhead results in the loss of identity protection. In IKE/ISAKMP's usual mode of operation, identities are exchanged only after a common shared secret key has been used to establish a secure communications channel. In this way the identity exchange is protected. However, when using an aggressive exchange, there is no established secure communications channel to protect the identity exchanges. Nevertheless, the aggressive exchange attempts to establish all security-relevant information in a single exchange. The definition of the aggressive

exchange also allows only a single proposal and a single transform to be negotiated (i.e., no choices are allowed).

At first it would appear that the IKE/ISAKMP aggressive exchange is the answer to all of the space community's problems. It reduces the number of round trips and the payload overhead required to establish a security association, since it does not allow more than one proposal to be negotiated. However, its use also reduces the generality of the protocol, and there is a loss of authenticated identity.

But, despite the loss of authenticated identity and the ability to send multiple proposals, the security associations and key exchanges would still be interoperable with the ground-based Internet. This means that there appears to be a way to implement an existing Internet standard in a space communications environment while still preserving bandwidth and maintaining compatibility with the ground.

Although the results of this analysis seem to indicate that IKE, using aggressive exchange, should be adopted by the space community for key exchange, testing should first be performed to ensure that this is the correct answer. A test bed to demonstrate and test IKE needs to be established. IKE/ISAKMP servers should be set up running in an aggressive exchange manner. Measurements should be taken showing the overhead and latency of a non-aggressive-mode exchange versus an aggressive exchange. The differences in bandwidth utilization and round trips would be analyzed to determine the best approach for use of IKE/ISAKMP in the space community.

3.3 STRAWMAN IKE PROFILE

As was previously stated, one way in which IKE could successfully be employed in the space community would be through an agreed upon minimalist profile. This would be a specified subset of IKE that would remain compatible with other IKE implementations but use less bandwidth and offer less overhead than a non-minimal subset. As a result, a strawman IKE profile for space environments can take the following appearance for Phase I negotiations:

a) Use aggressive mode rather than main mode if overhead and the number of round-trips are of concern (when faced with limited bandwidth links or limited contact times).

b) Use only triple DES (3DES) as the proposed encryption algorithm (in the future this should be changed to the Advanced Encryption Standard [AES] as it finds its way into the mainstream).

c) Use either MD5 or SHA-1 as the proposed hash algorithm.

d) Set the IKE security association lifetime to a long period (e.g., hours, days) in order to re-use the existing IKE SA for other security association negotiations. The actual length of the SA lifetime should be dictated by local security policies.

4 CONCLUSIONS

The descriptions in the previous sections have illustrated that there has already been a great deal of work done to create various security and key management protocols in several different standards bodies.

The IPSEC protocols are currently being rolled out into everyday use in products such as Virtual Private Network (VPN) devices (in both client software as well as gateway servers), firewalls, routers, and embedded into operating systems (e.g., Windows 2000/XP, Solaris, Linux, and xBSD).

An important point that must be remembered is that the IPSEC protocols and SCPS-SP affect *every* IP packet transmitted and received. They cause additional overhead to be added to each packet (IPSEC ESP adds a minimum of 10 bytes of overhead per IP packet). SCPS-SP was developed to minimize the amount of per-packet overhead for space communications systems—reduced to a minimum of 2 bytes per packet which is an 80% reduction in overhead resulting from security. However, because of the need to reduce the per-packet overhead for space communications, we have ended up with two, non-interoperable security protocols. These different security protocols can be made to interoperate by creating a translating gateway between SCPS and Internet protocols, which includes translation between IPSEC and SP. As was previously explained, the course of action is to extend the existing SCPS gateway to add IPSEC ESP to the 'Internet-side' of the gateway.

While IPSEC ESP and SCPS-SP cause overhead to be added on a per-packet basis, a typical key management/exchange protocol does not. There are some key management/exchange protocols that do cause an increase in per-packet overhead (e.g., SKIP), but the Internet community required that their key management standard operate 'out-of-band (that is, not play an active role in each packet transmitted and received). IKE/ISAKMP is such an 'out-of-band' key management protocol. It operates as an application (peer-to-peer applications) that performs security parameter negotiations, including key exchanges, and then drops out of the connection flow.

As a result, IKE/ISAKMP can be run occasionally to set up security associations and not 'cost' additional overhead on a per-packet basis. While IKE/ISAKMP typically requires several round trips to establish a security association and perform key exchanges, there is a mode that provides a means to reduce the number of round trips required to perform a negotiation (i.e., aggressive exchange). In the space communications environment aggressive exchange appears to make good sense, since it eliminates additional round trips and reduces the uncertainty of the negotiation process (in the sense that aggressive exchanges do not allow the negotiation of multiple security proposals). However, this conclusion would have to be proven by performing tests.

IKE/ISAKMP certainly has more overhead in its baseline protocol than would be normally desirable for space communications (e.g., 16 bytes of initiator and responder cookies). However, this is not per-packet overhead on all network traffic; rather, it is just security association establishment, or re-keying.

Using an existing Internet standard for both space- and ground-based systems outweighs the benefits that would be achieved by developing an entirely new protocol. Rather than spend the time and effort (and money) to develop a new, non-interoperable protocol, it could be argued that it makes more sense to use IKE/ISAKMP, with aggressive exchanges, with its reduced overhead, since it is used only on an occasional basis as compared to continuous network traffic. IKE/ISAKMP could also be run in a mode whereby multiple security associations are negotiated which are then cached for future connections, thereby saving even more overhead while still using a standard, off-the-shelf protocol.

ANNEX A

ABBREVIATIONS AND ACRONYMS

AH	Authentication Header
AIST	Advanced Information Systems Technology
ANSI	American National Standards Institute
CCSDS	Consultative Committee for Space Data Systems
DoD	US Department of Defense
ESP	Encapsulating Security Payload
FTP	File Transfer Protocol
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IP	Internet Protocol
IPSEC	Internet Protocol Security
ISAKMP	Internet Security Association and Key Management Protocol
ISO	International Organization for Standardization
MD5	Message Digest 5
NGSI	Next Generation Space Internet
NSA	National Security Administration
NTP	Network Time Protocol
PBK	Purpose Built Keys
SA	Security Association
SCPS	Space Communications Protocol Specification
SCPS-FP	Space Communications Protocol Specification-File Transfer Protocol
SCPS-NP	Space Communications Protocol Specification-Network Protocol

SCPS-SP	Space Communications Protocol Specification-Security Protocol
SCPS-TP	Space Communications Protocol Specification-Transport Protocol
SDNS	Secure Data Network System
SHA-1	Secure Hashing Algorithm-1
SKIP	Simple Key Management for Internet Protocol
STRV	Space Technology Research Vehicle
TCP	Transmission Control Protocol
TEK	Traffic Encryption Key
VPN	Virtual Private Network

ANNEX B

INFORMATIVE REFERENCES

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

[2] Atkinson, R. *IP Authentication Header*. RFC 1826, August 1995.

[3] Atkinson, R. *Encapsulating Security Payload (ESP)*. RFC 1827, August 1995.

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

[5] *Report on the Workshop on Key Management Using Public Key Cryptography*. NIST, http://www.nist.gov/kms, February 10-11, 2000.

[6] Rahikka, D. *NIST-KMS Key Management Standard: User Perspectives/Requirements: Wireless Applications*. (Presentation slides.) February 10, 2000.

[7] Krawczyk, H., *SKEME: A Versatile Secure Key Exchange Mechanism for Internet*. IBM T.J. Watson Research Center, 1996.

[8] Korhonen, M. *IPv6 Key Management*. Helsinki University of Technology, http://www.tml.hut.fi/Opinnot/Tik-110.551/1196/keymgmt.htm

[9] Summary of ANSI X9.44 Public Key Cryptography for the Financial Services Industry: *Key Establishment Using Factoring Based Public Key Cryptography*.

[10] Summary of ANSI X9.63 Public Key Cryptography for the Financial Services Industry: *Key Agreement and Key Transport Using Elliptic Curve Cryptography*.

[11] Thomas, M., and Oran, D., *Home Agent Cookies for Binding Updates*. draft-thomas-mobileip-ha-cookies-00.txt, March 2001.

[12] Nikander, P., and Perkins, C. *Binding Authentication Key Establishment Protocol for Mobile IPv6*. draft-perkins-bake-00.txt, April 2001.

[13] Bradner, S., Mankin, A., and Schiller, J. *A Framework for Purpose Built Keys (PBK)*. draft-bradner-pbk-frame-00.txt, February 2001.

[14] Piper, D., The Internet IP Security Domain of Interpretation for ISAKMP. RFC 2407, November 1998.