(U) TECHNICAL REQUIREMENTS DOCUMENT

FOR THE

PROGRAMMABLE OBJECTIVE ENCRYPTION TECHNOLOGIES (POET)

ADVANCED CRYPTOGRAPHIC MODULE (ACM)

(U) Revision History

Rev.	Date	Description of Change	Affected Pages
0.1	10/13/05	Initial Draft	All
0.2	11/14/05	Second Draft	All
0.3	11/17/05	Updated cover sheet Â“Document Distribution RestrictionsÂ”	Cover Page

(U) TABLE OF CONTENTS

Section Page

(U) List of Figures

Section Page

(U) List of Tables

Section Page

1.
2. (U) Introduction

1.
    1.
    2. (U) Scope

(U) The Programmable Objective Encryption Technologies (POET) program will design and develop an Advanced Cryptographic Module (ACM) capable of meeting the communications security needs outlined below. This document provides the technical requirements of the POET ACM.

(U) Future terminal architectures will require enhanced high-speed cryptographic capabilities in order to support the higher data rates and increased information security (INFOSEC) demands imposed by modern waveforms and ever-increasing security regulations.

(U) These future terminals are envisioned to be a family of multi-band, multi-mode, transportable, modular, flexible, Software Communications Architecture (SCA) compliant, satellite and Line of Sight (LOS) systems supporting simultaneous communications between ground, airborne and surface/subsurface units for a range of missions across all services in tactical operational environments.

(U) These terminals will be capable of operating with the military and commercial satellites, envisioned to be operational in the 2010 and beyond timeframe, utilizing X, Ku, Ka, and Q-band frequencies (above 2 GHz). These systems include, but are not limited to, the Wideband Gapfiller System (WGS), Advanced Extremely High Frequency (AEHF), Global Broadcast System (GBS) and Transformational Satellite Communications System (TSAT) constellations. Commercial satellite support will include both Ku and Ka-band systems. The terminals will also provide high-capacity line-of-sight (LOS) communications utilizing Common Data Link (CDL) / Networked Common Data Link (N-CDL) links or future variations thereof.

(U) Because the family of terminal systems will operate with many legacy waveforms currently used by military and civilian agencies and incorporate new waveforms as they are developed, the cryptographic components of the terminal system will need to be programmable and scaleable to meet specific user operational needs. The desire is that the cryptographic capability provides flexibility through an open system architecture that enables technology insertion (via re-

programmability or other means). The terminal system will be capable of high data throughput rates per Radio Frequency (RF) channel; incremental RF channel expansion; high levels of reliability, availability, and maintainability; technological enhancement; and commercial support service compatibility.

(U) The emphasis of the initial POET development will be to mitigate the risks to the terminal developments by establishing a flexible and scalable cryptographic capability for which the long lead time development and certification efforts are already complete. The emphasis is not to develop a particular Â“one size fits allÂ” cryptographic product, but rather a capability that can be reused, scaled, and/or repackaged to satisfy the particular constraints of the different terminal developments. Achieving NSA certification of the POET Engineering Development Models (EDMs) is considered an important aspect of mitigating the risks to the terminal developments.

1.
2.
3. (U) System Overview

(U) POET ACM flexibility is important due to the wide range of applications envisioned for these future terminals, which may differ significantly in their mission applications and their size weight and power (SWAP) requirements. In turn, the performance capabilities and the SWAP requirements to be imposed on the ACM may vary significantly. For example, a man-pack embedment would likely trade-off some multi-band/multi-channel capability and data rate capacity in return for a more stringent SWAP. In this case, an ACM might be required to support two RF channels at lower data rates. Fixed terminals will likely require the maximum functionality and capabilities and ACM embedments will be capable of utilizing the increased SWAP available. Therefore, flexibility, modularity, and scalability of design and architecture are necessary for the ACM.

(U) It is also anticipated that the ACM will be used in terminals with different INFOSEC architectures. Some terminals are expected to implement a Â“RED coreÂ” terminal architecture while other terminals are expected to implement a Â“REDÂ” terminal architecture. A RED core terminal is defined as a terminal that may internally process classified information as required by one or more waveforms, but for which all external interfaces are unclassified. External interfaces include baseband Input/Output (I/O) interfaces, locally connected operator interfaces, and RF interfaces. A RED core terminal is capable of unattended operation in the ground tactical environment. A RED terminal is defined as a terminal that may internally process classified information and may also support classified external interfaces. A RED terminal is not capable of unattended operation in the ground tactical environment.

(U) A notional Â“RED coreÂ” terminal architecture with embedded ACM is depicted in Figure 1-1. A RED core terminal with an embedded ACM will:

-
- (U) Support only BLACK baseband traffic interfaces. All Â“RedÂ” user data will be encrypted prior to entering the terminal.

- (U) Provide the requisite Transmission Security (TRANSEC) functionality for communications with RF frequencies above 2 GHz.
- (U) Provide the requisite Communications Security (COMSEC) functionality for communications with above 2 GHz systems.
- (U) Incorporate an embedded Type 1 High Assurance Internet Protocol Encryptor (HAIPE) capability within the INFOSEC boundary of the terminal in order to support terminal remote control/management requirements.

Figure 1-1. (U) Notional RED Core Terminal Block Diagram

UNCLASSIFIED//FOUO

UNCLASSIFIED//FOUO

(U) A notional Â“REDÂ” terminal architecture with embedded ACM is depicted in Figure 1-2. A RED terminal with an embedded ACM will:

-
- (U) Support BLACK and/or RED baseband traffic interfaces. All Â“RedÂ” user data will be encrypted within the terminal.

- (U) Provide the requisite TRANSEC functionality for communications with RF frequencies above 2 GHz.
- (U) Provide the requisite COMSEC functionality for communications with above 2 GHz systems.
- (U) Incorporate an embedded Type 1 HAIPE capability within the INFOSEC boundary of the terminal in order to support requirements for remote control/management of the terminal.

Figure 1-2. (U) Notional Red Terminal Block Diagram

UNCLASSIFIED//FOUO

UNCLASSIFIED//FOUO

1.

3.

4. (U) Definition of Terms

(U) The following paragraphs provide definitions for terminology that is used in the remainder of this TRD. These terms are defined here to promote uniform understanding of the requirements for the ACM.

(U) Encryption Â– The process of converting Plaintext into Ciphertext by means of a code or cryptographic system.

(U) Decryption Â– The process of converting Ciphertext into Plaintext by means of a code or cryptographic system.

(U) Plaintext Â– Unencrypted information.

(U) Ciphertext Â– Encrypted information.

(U) Cryptographic Algorithm Â– Well-defined procedure or sequence of rules or steps, or a series of mathematical equations used to describe cryptographic processes such as encryption/decryption, key generation, authentication, signatures, etc.

(U) RED interface Â– An interface on which plaintext may enter or emerge from the ACM.

(U) BLACK interface Â– An interface on which unclassified plaintext and/or ciphertext may enter or emerge from the ACM.

(U) RED Â– Pertaining to plaintext.

(U) BLACK Â– Pertaining to unclassified and/or ciphertext.

(U) RED Core Terminal Â– A terminal that may internally process classified information as required by one or more waveforms/applications, but for which all external interfaces are unclassified. A RED Core terminal implies a terminal with the following attributes:

-
- (U) No RED user data is exposed at any of the terminalÂ's interfaces. All RED user data is externally converted to BLACK data prior to being presented to any terminal interface.
- (U) No RED user data is displayed on the terminal itself.
- (U) All keys and cryptographic algorithms are stored in BLACK form, except when needed for processing within an approved cryptographic boundary within the terminal and only if using an approved embedment with a certified cryptographic module to provide the required information assurances.
- (U) When the terminal is placed into a specific non-operational mode, its classification sensitivity is unclassified CCI.
- (U) When the terminal is operating, its classification sensitivity depends on the mission profile and operating mode of the terminal.

(U) RED Terminal Â– A terminal that may internally process classified information and may also support classified external interfaces.

(U) COMSEC Â– Acronym for Â“Communications SecurityÂ”; generally includes security mechanisms that provide high assurance that information is protected while in transit or at rest. The process for COMSEC protection of information typically involves encryption of plaintext from the RED side of the cryptographic module to ciphertext on the BLACK side. Decryption of the ciphertext by intended recipients would be realized from BLACK side of a cryptographic module. The original plaintext would be presented on the RED side of the cryptographic module.

(U) TRANSEC Â– Acronym for Â“Transmission SecurityÂ”; generally includes measures to protect electronic emanations and RF transmissions. TRANSEC may provide anti-jam, Low Probability of Intercept (LPI), Low Probability of Detection (LPD), and Low Probability of Exploitation (LPE).

(U) TRANSEC Key Stream Â– A pseudo-random bit stream used in conjunction with TRANSEC functions/processing. TRANSEC key streams are typically consumed within a modem/signal processing system in support of TRANSEC protected transmissions.

(U) Cover Â– Application of a pseudo-random key stream to BLACK information. Cover is often used to prevent traffic-flow analysis by an adversary and/or to prevent other adversary attacks. Cover is sometimes referred to as Â“TRANSEC encryptionÂ”.

(U) Decover Â– Application of a pseudo-random key stream to BLACK information. Decover removes the application of cover. Decover is sometimes referred to as Â“TRANSEC decryptionÂ”.

(U) Bulk Encryption Â– Refers to the encryption function applied to a serial data stream. Bulk encryption may be a second encryption that is applied to data that is separately encrypted by another process.

(U) Bulk Decryption Â– Refers to the decryption function that corresponds to a bulk encryption.

(U) HAIPE Encrypt Â– A COMSEC function that is used for providing High Assurance Type 1 protection for internet protocol data between two endpoints.

(U) HAIPE Decrypt Â– Refers to the decryption function that corresponds to HAIPE encryption.

(U) Waveform COMSEC Â– Encryption/decryption functions that are required processing in support of a particular waveform.

(U) Baseband COMSEC Â– Encryption/decryption of data that is received/transmitted at the ingress/egress of a host terminal interface to a user.

2.
3. (U) Applicable Documents

1.
    1.
    2. (U) Program-Specific Documents

Table 1. (U) Program-Specific Documents

UNCLASSIFIED//FOUO (TO BE SUPPLIED)

Doc ID	Name	Classification
CR-1021	CR2P Performance Work Statement (PWS) 05 April 2005	None
CDRL A001	HC3 Study Modular Architecture Report, Raytheon Company, July 2004	None
CDRL A005-2	HC3 Study Security Architecture Report, Raytheon Company, 29 March 2005	None
CDRL A001	HC3 Study Modular Architecture Report, The Boeing Company, 01 October 2004	None
CDRL A005	HC3 Study Security Architecture Report, The Boeing Company, 01 August 2005	None

UNCLASSIFIED//FOUO

1.
    2.
    3. (U) Government Documents

Table 2. (U) Government Documents

UNCLASSIFIED//FOUO(TO BE SUPPLIED)

Doc ID	Name	Classification
	Terminal Information Assurance Reference Architecture, Version 4 +, Sept. 2005	(U//FOUO)
	Advanced EHF (AEHF) System Program Alternate Key Management Plan (AKMP) , 6 February 2004	

	Advanced Extremely High Frequency (AEHF) Key Management Plan (KMP), 30 September 2004	
	AKMP Advanced Extremely High Frequency (AEHF) Key Management System Description (KMSD), Version 0.75, 20 October 2004	
	TSAT Key Management Architecture V2.0, October 2005	
JROCM 134-01	Capstone Requirements Document Global Information Grid	
DoD Instruction 8500.2	Information Assurance (IA) Implementation	
DoD Directive 8500.1	Information Assurance (IA) Directive	
NSA 03-01A	Cryptographic Interoperability Specification for Link Encryptor Family (LEF) Equipment Version 2.0	(U//FOUO)
JTRS 5000-SEC	Security Supplement to the Software Communications Architecture Specification V2.2.1	(U)
EKMS 308 Rev. D	EKMS Data Tagging and Delivery Standard,	(U//FOUO)
EKMS 217 Rev G	EMKS Benign Techniques Specification	(U//FOUO)
EKMS 317	Generic Fill Format Specification, EKMS Phase 4 Baseline	(U//FOUO)
	Programmability Design Guidance for Crypto Modernization, Version 4.0, 17 November 2003	(U//FOUO)
	Using CMS to Protect Firmware Packages: Algorithms for use with Type 1 Cryptographic Modules, Draft 0.4 September 2003	(U//FOUO)
	Trust Anchor Management Protocol (TAMP), Draft 0.8, 13 October 2003	(U//FOUO)
	Trust Anchor Management Protocol (TAMP): Algorithms for use with Type 1 Cryptographic Modules, Draft 0.0, August 2003	(U//FOUO)
KMI 3001	Electronic Serial Number Standard, Draft 0.7, 23 October 2003	(U//FOUO)
	Certificate and CRL Profiles for Use by End Crypto Units (ECUs) Version 0.92, 14 October 2003	(U//FOUO)

1.
    1.
        1.
        2. (U) Specifications, Standards, or Handbooks

Table 3. (U) Specification, Standards, or Handbooks

UNCLASSIFIED//FOUO (TO BE SUPPLIED)

Doc ID	Name	Classification
FIPS Pub 186	Digital Signature Standard, 19 May 1994	None
FIPS Pub 180-1	Secure Hash Standard, 17 April 1995	None
FIPS Pub 180-2	Secure Hash Standard, 1 August 2002	None

UNCLASSIFIED//FOUO

1.
    3.
    4. (U) Other

1.
    1.
        1.
        2. (U) Standards

Table 4. (U) Standards

UNCLASSIFIED//FOUO (TO BE SUPPLIED)

Doc ID	Name	Classification
RFC4109	Algorithms for Internet Key Exchange version 1, P. Hoffman, May 2005	None
RFC4108	Using Cryptographic Message Syntax (CMS) to Protect Firmware Packages , Rich Housley, August 2005	None
ANSI X9.62-1998	Public Key Cryptography For the Financial Services Industry: The Elliptical Curve Digital Signature Standard (ECDSA), January 7, 1999	None
RFC3369	Cryptographic Message Syntax (CMS), August 2002	None

UNCLASSIFIED//FOUO

3.
4. (U) Technical Requirements

1.
    1.
    2. (U) Notional Advanced-High-Speed-Embeddable Cryptographic Module (ACM)

(U) Figure 3-1 is a notional functional block diagram of the ACM, which will serve as the basis for the discussions and the requirements in this document. The ACM is an embeddable, reprogrammable cryptographic module which provides cryptographic services to its host (terminal or platform).

(U) The ACM should be designed for reuse within a variety of host platforms having differing performance and SWAP requirements. The initial ACM design will provide the functionality and performance required for a core set of cryptographic services demanded by a broad range of host platforms. The ACM architecture should allow for modular design and expansion capabilities to support candidate platforms and applications as they evolve.

Figure 3-1. (U) Notional Advanced Cryptographic Module (ACM)

UNCLASSIFIED//FOUO

1.
    1.
        1.
        2. (U) ACM Capabilities

(U) The ACM provides the following major capabilities:

-
- A single cryptographic capability that supports:
    1.
    2. Simultaneous, high-speed key stream generation for independent waveforms
    3. COMSEC support concurrent with high-speed key stream generation
    4. COMSEC support for network management and remote control using HAIPE IS V3.x [TBD] or IPSEC as required
    5. Control/status bypass for terminal management
    6. High Assurance Guard for Ciphertext bypass
- Scalability Throughput and Power
- Multiple Single Security Levels

- Bulk Encryption for resource management and control information for some waveforms
- Programmable Algorithm & Cryptographic Modernization Initiative compliance
- Remote Management
- Unattended Operation
- Key Management
- Algorithm Interoperability
- SCA Security Supplement compliance
- Access Control with use of Token
- Anti-tamper and Zeroization
- Non Type 1 support

1.
    2.
    3. Requirements Organization

(U) The ACM requirements are primarily categorized into the following areas:

-
- Functional
- External Interfaces
- Processor Capacity
- Memory Capacity
- Internal Battery
- Power
- Physical
- Environmental
- Thermal
- Maintainability
- Reliability
- Interchangeability
- Workmanship
- Documentation
- Logistics

1.
    3.
    4. (U) Functional

(U) The ACM functional requirements can be categorized based on following capabilities:

-
- General Capabilities
- Waveform Specific Capabilities
- Levels of Security and Classification
- Remote Management
- Unattended Operation

- Cryptographic Modernization Initiative
- SCA Security Compliance
- Cryptographic Services
- Cryptographic Control and Status
- Clock Management
- Built-In Test and Health Status
- Alarm
- Tamper
- Interoperability

1.
    1.
        1.
        2. (U) Cryptographic Services

(U) The ACM is a set of hardware, software, and firmware that provides cryptographic services to a host terminal. The ACM will provide cryptographic services to the host via a well defined set of primitives/functions. Each primitive/function will provide an implementation of a cryptographic process such as encryption/decryption, key stream generation, authentication, etc.

(U) The ACM must support TRANSEC key stream generation, cover/decover key stream generation, bulk encryption/decryption, waveform COMSEC encryption/decryption, baseband COMSEC encryption/decryption, bypass processing, and ancillary key management, key agreement, digital signature, and utility services that are provided for the host. These general cryptographic services provided by the ACM are discussed in more detail in the subsections that follow.

(U) The different types of cryptographic services are presented separately so that it is clear what types of functionality the ACM must support. This does not imply that there must be some difference in implementation among the different cryptographic services. For example, the same algorithm and implementation could apply for bulk encryption/decryption and waveform COMSEC encryption/decryption. In general, it is desired for the cryptographic services provided by the ACM to be as flexible as possible in order to accommodate application in different types of terminals and evolvability within the terminals.

(U) In the requirements that follow, some of the algorithms identified may be in the process of TBD, but are necessary for terminal level interoperability with existing systems. These algorithms will no longer be used within the ACM when there is no longer a requirement for interoperability with existing systems.

1.
    1.
        1.
            1.

2. Identification and Authentication (I&A)

(U) It is expected that the ACM ICD will identify management commands that require identified and authenticated entities to accomplish; however, the ACM will not store or manage identity or role information related to the entities authorized to execute those commands. It is anticipated the ACM host will accomplish entity I&A and role validation. Subsequently, if an entity requires access to ACM stored information to accomplish specific activities including adding or deleting ACM stored data, accessing audit information, effecting changes to ACM internal configuration then the host will verify I&A and assigned role then pass the validated request to the ACM over the command interface. The ACM will act on the command IAW the applicable security policy. At a minimum, the ACM acting on the management command shall generate an audit entry that includes entity identity, time, role, requested service, and summarization of request result.

(U) The ACM shall support the following minimum roles, security administrator, system maintainer, and user.

(U) Security administrator: At a minimum and as allowed by security policy, the Security Administrator role shall be able to query ACM status, accomplish key management tasks such as request an inventory of stored keys, delete keys, or request keys be added to the key store using ACM key management primitives, command a load of updated algorithms to the protected store or delete stored algorithm images from the protected stored, or access all audit entries. Note, it is recommended that the host establish high robustness, multifactor, I&A processes for the Security Administrator.

(U) System Maintainer. At a minimum and as allowed by security policy, the System Maintainer role shall be able to query the ACM for information related to ACM heath and performance including he results of the last full BIT, command the ACM to accomplish a BIT or specific performance tests the ACM may be designed to accomplish, or access health or performance audit records. Note, it is recommended that the host establish high robustness I&A processes for the System Maintainer.

(U) User: Consistent with security policy, the user shall be able to request ACM status (is it up or down). Note, ACM status requests may cause the ACM to accomplish performance testing; however, the user will only be provided summarized test results.

1.
    1.
        1.
            2.
            3. Key Stream Generation

(U) The ACM must support key stream generation for both TRANSEC key streams and cover/decover key streams. These functions are required for some of the waveforms that must be supported by future terminals. In addition to these waveforms, additional waveforms could subsequently be specified at the terminal level that also require key stream generation. This could occur, for example, if a waveform is added to the list of waveforms that must be supported by a particular terminal or if a new type of terminal is defined that will be supported by POET. It is expected that the ACM implementation will provide flexibility to accommodate such changes, if necessary, as the terminal programs mature.

(U) The ACM shall be able to operate with the following cryptographic TRANSEC and cover/decover algorithms:

1. AES Counter (CTR) mode

2. AES Galois Counter (GCM) mode

3. AES Electronic Code Book (ECB) mode

4. AES Cipher Block Chaining (CBC) mode

5. MEDLEY mode(s) in accordance with the classified supplement of this document

6. SHILLELAGH TBD mode

7. BATON TBD mode

8. KEESEE TBD mode

9. SAVILLE TBD mode

< Are these modes classified? If so use the same statement as for MEDLEY>

1.
    1.
        1.
            1.
                1.
                2. TRANSEC Key Stream Generation

(U) The ACM must support TRANSEC key stream generation for the Milstar, AEHF, TSAT, and MIL-STD-188-EEE waveforms. The following requirements are applicable to TRANSEC key stream generation. A TRANSEC key stream may be a continuous stream of bits, a periodic sequence, or a waveform specific request/reply function between the host and the ACM.

(U) The ACM [CR0130] shall support TRANSEC key stream generation for AEHF as defined in Appendix A, Table A-1 (U) AEHF Algorithms and concurrent with the other cryptographic services identified as simultaneous.

Note: The requirements for TRANSEC key stream generation for the AEHF system are a superset of the requirements for TRANSEC key stream generation for the Milstar system.

(U) The ACM shall support TRANSEC key stream generation for TSAT as defined in Appendix A, Table A-5 (U) TSAT Algorithms and concurrent with the other cryptographic services identified as simultaneous.

(U) The ACM shall support cover/decover key stream generation for MIL-188-EEE as defined in Appendix A, Table A-3 (U) MIL-188-EEE Algorithms and concurrent with the other cryptographic services identified as simultaneous.

(U) The ACM shall provide the capability to output TRANSEC key streams to either a BLACK side ACM interface or a RED side ACM interface based on the ACM security policy.

Note: This provides flexibility to the host. Existing hosts process TRANSEC key streams on the BLACK side, but some hosts may process TRANSEC key streams on the RED side.

(U) The ACM shall be capable of simultaneously providing TRANSEC key stream and cover/decover key stream generation functions for up to four independent configurable waveforms having a maximum aggregate rate of (2 Gbps, threshold; 10 Gbps, objective).

<Recommend replacing this with a requirement that captures the worst case simultaneous waveforms based on the processing requirements rolled up in appendix A>

(U) The ACM shall provide at least eight (8) independent TRANSEC Key Stream cryptographic channels for up to four waveforms.

<Recommend replacing this with a requirement that captures the worst case simultaneous waveforms based on the processing requirements rolled up in appendix A>

(U) When applicable (e.g., some algorithms require TOD for crypto sync), the ACM shall use time-of-day (TOD) functions for key stream generation.

(U) When applicable, the ACM shall synchronize clock for key stream generation.

(U) The ACM shall define primitives required to facilitate the high speed TRANSEC functions required for the terminal waveforms.

1.
    1.
        1.
            1.

2.
3. Cover/Decover Key Stream Generation

(U) The ACM must support cover/decover key stream generation for the Milstar, AEHF, TSAT, and MIL-STD-188-EEE waveforms. The following requirements are applicable to cover/decover key stream generation. A cover/decover key stream may be a continuous stream of bits, a periodic sequence, or a waveform specific request/reply function between the host and the ACM.

(U) The ACM shall support cover/decover key stream generation for AEHF as defined in Appendix A, Table A-1 (U) AEHF Algorithms and concurrent with the other cryptographic services identified as simultaneous.

Note: The requirements for cover/decover key stream generation for the AEHF system are a superset of the requirements for TRANSEC key stream generation for the Milstar system.

(U) The ACM shall support cover/decover key stream generation for TSAT as defined in Appendix A, Table A-5 (U) TSAT Algorithms and concurrent with the other cryptographic services identified as simultaneous.

(U) The ACM shall support cover/decover key stream generation for MIL-188-EEE as defined in Appendix A, Table A-3 (U) MIL-188-EEE Algorithms and concurrent with the other cryptographic services identified as simultaneous.

(U) The ACM shall provide the capability to output cover/decover key streams to either a BLACK side ACM interface or a RED side ACM interface based on the ACM security policy.

Note: This provides flexibility to the host. Existing hosts process cover/decover key streams on the BLACK side, but some hosts may process cover/decover key streams on the RED side.

(U) The ACM shall provide at least four independent cover/decover key stream generation cryptographic channels for up to four waveforms.

<Recommend replacing this with a requirement that captures the worst case simultaneous waveforms based on the processing requirements rolled up in appendix A>

(U) The ACM shall be capable of supporting multiple cover/decover key stream generation functions with different cryptographic algorithms and keys over the same TRANSEC Key stream generation cryptographic channel in accordance with the security policy.

(U) When applicable, the ACM shall use time-of-day (TOD) functions for TRANSEC key stream generation.

(U) When applicable, the ACM shall synchronize clock for TRANSEC key stream generation.

1.
    1.
        1.
            3.
            4. Bulk Encryption/Decryption

(U) The Â“Bulk Encrypt/DecryptÂ” function provides a COMSEC function which is necessary when plaintext control data is multiplexed with ciphertext data. The ACM must support bulk encryption/decryption operations for the CDL and MIL-188-165 waveforms and may also be required for operation over commercial SATCOM systems. In addition to these requirements, additional waveforms could subsequently be specified at the terminal level that also require bulk encryption/decryption.

(U) The ACM shall support bulk encryption/decryption for MIL-188-165 as defined in Appendix A, Table A-2 (U) MIL-188-165 Algorithms and concurrent with the other cryptographic services identified as simultaneous.

(U) The ACM shall perform legacy CDL encryption/decryption for all CDL control information along with the user data.

(U) The ACM shall support bulk encryption/decryption for CDL as defined in Appendix A, Table A-4 (U) CDL Algorithms and concurrent with the other cryptographic services identified as simultaneous.

1.
    1.
        1.
            4.
            5. Waveform COMSEC Encryption/Decryption

(U) The ACM must support the COMSEC encryption/decryption processing that is required by the AEHF, TSAT, MIL-188-165, and MIL-188-165 waveforms. In addition to these waveforms, additional waveforms could subsequently be specified at the terminal level that also require COMSEC encryption/decryption functions. It is expected that the waveform COMSEC encryption/decryption requirements specified here will provide ample flexibility for such changes, if necessary, as the terminal programs mature.

(U) The ACM shall support waveform COMSEC encryption/decryption using the following Suite A algorithms: MEDLEY, SHILLELAGH, BATON.

(U) The ACM shall support waveform COMSEC encryption/decryption using the following Suite B algorithms: AES.

(U) The ACM shall support waveform COMSEC encryption/decryption for AEHF as defined in Appendix A, Table A-1 (U) AEHF Algorithms and concurrent with the other cryptographic services identified as simultaneous.

(U) The ACM shall support waveform COMSEC encryption/decryption for TSAT as defined in Appendix A, Table A-5 (U) TSAT Algorithms and concurrent with the other cryptographic services identified as simultaneous.

(U) The ACM shall support waveform COMSEC encryption/decryption for MIL-188-165 as defined in Appendix A, Table A-2 (U) MIL-188-165 Algorithms and concurrent with the other cryptographic services identified as simultaneous.

(U) The ACM shall support waveform COMSEC encryption/decryption for MIL-188-EEE as defined in Appendix A, Table A-3 (U) MIL-188-EEE Algorithms and concurrent with the other cryptographic services identified as simultaneous.

(U) The ACM shall provide the capability to receive waveform message to be encrypted from either a BLACK side ACM interface or a RED side ACM interface based on the ACM security policy.

Note: This provides flexibility to the host. For example, waveform processing functions and information may be within the BLACK side host.

1.
    1.
        1.
            5.
            6. Baseband COMSEC Encryption/Decryption

(U) The ACM must support the COMSEC processing that is required for terminals that support encryption/decryption of end user data that is received on a baseband interface. These requirements are not specified as simultaneous with the waveform processing requirements outlined in the previous sections because there may be a functional separation at the terminal level between waveform cryptographic processing and cryptographic processing for end user data. For example, physically separate instances of the ACM could be used within different components of the terminal (e.g., signal processing system module versus baseband I/O module). It may also be necessary to use physically separate instances of the ACM in order to avoid multiple levels of processing for waveforms and end user information at the terminal level. Even though the following baseband COMSEC encryption/decryption requirements are not specified as simultaneous with the waveform processing requirements, it is still desirable for the ACM to provide the maximum capability. An affordable and certifiable implementation that supports simultaneous baseband COMSEC encryption/decryption and waveform cryptographic processing is preferable as long as the existing threshold requirements are met.

(U) The ACM shall support baseband COMSEC encryption/decryption using the following Suite A algorithms: MEDLEY (TBD), SHILLELAGH (TBD), BATON (TBD), SAVILLE (TBD), WALBURN (TBD).

(U) The ACM shall support baseband COMSEC encryption/decryption as defined in Appendix A, Table A-10 (U) Baseband COMSEC.

1.
    1.
        1.
            6.
            7. HAIPE Encryption/Decryption

(U) The terminals will utilize HAIPIS compliant encryption/decryption for control, monitoring, and management of RED side terminal functionality from a remote management site. The threshold requirements that follow are specified based on the encryption/decryption rate that must be supported for control, monitoring, and management. It is expected that this threshold provides ample margin for control type traffic for different terminal types. It is desirable that a much higher data rate be supported for HAIPIS compliant encryption/decryption of baseband traffic because this makes the ACM more generally capable and useful for different terminal/military applications. The objective requirements that follow capture the desired data rate for HAIPIS compliant encryption/decryption as a COMSEC baseband function. It is important to note that the objective requirements are not specified as simultaneous with the threshold waveform processing requirements.

The ACM shall provide HAIPE cryptographic services compliant with the core and extension HAIPE IS Version 3.x [TBD] requirements.

Note: The requirements traceability matrix provided in Appendix B indicates the allocation of requirements between the ACM and the host.

(U) The ACM shall support HAIPE IS v3.x compliant encryption/decryption as defined in Appendix A, Table A-6 (U) HAIPE Algorithms.

(U) The ACM shall support HAIPE IS v3.x compliant encryption/decryption as defined in Appendix A, Table A-6 (U) HAIPE Algorithms for as many as 4 simultaneous interfaces between the host and remote management workstations.

(U) The ACM should support HAIPE IS v3.x compliant encryption/decryption for the objective data rates defined in Appendix A, Table A-6 (U) HAIPE Algorithms.

(U) The ACM shall provide HAIPE IS V3.x encryption/decryption for IPv4.

Note: This may be unnecessary if IPv4 is no longer required to be supported during the time period that the terminals will be fielded.

(U) The ACM shall provide HAIPE IS V3.x encryption/decryption for IPv6.

(U) The ACM should support bulk encryption/decryption for CDL as defined in Appendix A, Table A-4 (U) CDL Algorithms and concurrent with the other cryptographic services identified as simultaneous simultaneously with HAIPE IS v3.x compliant encryption/decryption for the objective data rates defined in Appendix A, Table A-6 (U) HAIPE Algorithms.

Note: This would allow a terminal that must simultaneously perform CDL bulk encryption/decryption and end-to-end packet encryption/decryption using the same ACM.

(U) The ACM shall provide HAIPE cryptographic services compliant with the High Assurance Internet Protocol Encryptor Interoperability Specification Guide, Traffic Protection Â– Suite A Cryptography, Version 3.0.1, 30 September 2005.

(U) The ACM shall provide HAIPE cryptographic services compliant with the High Assurance Internet Protocol Encryptor Interoperability Specification Guide, Traffic Protection Â– Suite B Cryptography, Version 3.0.1, 30 September 2005.

(U) The ACM shall provide HAIPE cryptographic services compliant with the High Assurance Internet Protocol Encryptor Interoperability Specification Guide, Traffic Protection Â– Legacy Cryptography, Version 3.0.1, 30 September 2005.

Note: The legacy crypto extension specification will be removed in HAIPIS version 4.0.0. This requirement may be contingent on the required interoperability at the time that the ACM will be used.

(U) The ACM shall provide the maximum throughput for each of HAIPE encryption algorithm per cryptographic channel as identified in APPENDIX A, Table A-6 (U) HAIPE Algorithms.

(U) The ACM shall support the Cryptographic Suite B for HAIPE IS V3.x [TBD] Suite B extension.

1.
    1.
        1.
            7.
            8. Non Type 1 Services

(U) Terminals will support both Type 1 and non Type 1 cryptographic processing. For example, TSAT terminals will communicate with the TSAT network management system using a medium robustness confidentiality mechanism. TSAT terminals will utilize Medium Assurance DoD PKI

device certificates to perform key agreement and negotiate shared session keys with the TSAT network management system. TSAT terminals will also use medium robustness confidentiality mechanisms to secure the network control plane connection to a TSAT payload. Additionally, terminals are expected to utilize DoD PKI certificates for identification and authentication of their own operators and administrators.

A particular terminal integration could choose to implement the required medium robustness confidentiality mechanisms using a separate cryptographic product. There are several advantages, however, to using the ACM to support both Type 1 and non Type 1 cryptographic processing. These advantages are include: common and very secure storage for private keys; no need to build/buy and NIAP certify an additional product; and existing algorithm overlap between Suite B and NIAP

The following requirements describe the capabilities the ACM must provide to support interoperability with the DoD PKI system and non type 1 cryptographic processing using DoD PKI certificates or other non Type 1 key material. In some cases, an algorithm listed as a non Type 1 algorithm may be the same algorithm as a Suite B algorithm. A single ACM implementation of such algorithms may be possible, but differences in key management are expected.

1.
    1.
        1.
            1.
                1.
                2. Key Pair Management

(U) A public/private key pair shall be generated within the ACM and the public keys output to the DOD PKI for publication. Additionally, the ACM must accept and manage X.509v3 certificates and their corresponding keys over the instruction interface to the host in support of DOD PKI interoperable confidentiality, integrity, and authentication services. The following are requirements for DoD PKI key pair management functions that the ACM must support.

(U) The ACM shall generate Public/Private key pairs when commanded by the red or black side Cryptographic Control Interface.

(U) After generation of a public/private key pair, the ACM shall retain the Private key and wrap it for storage within the cryptographic boundary.

(U) After generation of a public/private key pair, the ACM shall export the Public key over the Cryptographic Control Interface.

(U) The ACM shall support the loading of DoD PKI Certificates and their corresponding public keys over the Cryptographic Control Interface.

(U) The ACM shall implement processes necessary to validate offered certificates.

Note. Validation may range from offering the credential particulars, e.g., serial number, valid from / to dates, etc., to an external, terminal resident process that proxies the validation to incorporating an OCSP or other lightweight validation protocol client and/or a full certificate validation process capable of downloading and scanning a CRL. At a minimum, validation will include verifying offered certificate root is valid and known and verifying certificate has reached its valid-from date and not exceeded its valid-to date. The desired maximum is verifying root, date, and certificate revocation status.

1.
    1.
        1.
            1.
                1.
                2.
                3. Non Type 1 IPSec

(U) TSAT terminals will use non Type 1 Internet Protocol Security (IPSec) to communicate with the TSAT network management system and to secure the network control plane connection to a TSAT payload and possible future network management systems. IPSec will provide security services including access control, connectionless integrity, data origin authentication, detection and rejection of replays (a form of partial sequence integrity), confidentiality, and limited traffic flow confidentiality.

(U) The IPSec series of protocols makes use of various cryptographic algorithms in order to provide security services. The IPSec, Internet Key Exchange (IKE), and IKEv2 protocols rely on security algorithms to provide privacy and authentication between the initiator and responder. Terminals will use the Encapsulating Security Payload (ESP) and may use the Authentication Header (AH) to protect data being sent over an IPSec Security Association (SA). IKE provides a mechanism to negotiate which algorithms should be used in any given association, and two IPSec systems cannot interoperate unless they share common algorithms. The ACM therefore needs to support the current set of mandatory IPSec algorithms as well as algorithms that may be promoted to mandatory in the future.

(U) From the perspective of encryption/decryption algorithms, the algorithm that is used to support non Type 1 IPSec may be identical to a Type 1 Suite B algorithm. A single ACM implementation of such algorithms may be possible, but differences in key management must be addressed. The following requirements relevant to non Type 1 encryption/decryption may be redundant with other requirements in this TRD for Type 1 services, but are repeated here as non Type 1 requirements so that the context and intended applications are clear.

(U) The ACM shall support TripleDES-CBC in accordance with RFC 2451 for use with non Type 1 IPSec.

(U) The ACM should support AES-CBC in accordance with RFC 3602 for use with non Type 1 IPSec.

Note: All AES key sizes should be supported.

(U) The ACM should support AES-CTR in accordance with RFC 3686 for use with non Type 1 IPSec.

Note: All AES key sizes should be supported.

(U) The ACM should support AES-GCM in accordance with TBD for use with non Type 1 IPSec.

Note: All AES key sizes should be supported.

(U) The ACM should support AES-XCBC-MAC-96 in accordance with RFC 3566 for use with non Type 1 IPSec.

Note: All AES key sizes should be supported.

(U) The ACM shall support HMAC-SHA1-96 in accordance with RFC 3566 for use with non Type 1 IPSec.

(U) The ACM shall support IPSec compliant encryption/decryption as defined in Appendix A, Table A-7 (U) IPSec Algorithms.

1.
    1.
        1.
            1.
                3.
                4. Other Encryption/Decryption

Support for IPSec is one application of non Type 1 encryption/decryption that is known to be required for TSAT terminals. The following are additional encryption/decryption capabilities that are specified in order for the ACM to provide a more generally useful non Type 1 cryptographic capability. These functions are objectives and are believed to not significantly increase the complexity of the ACM, but would provide capabilities for additional waveforms/applications within the terminal.

(U) The ACM should support Rivest, Shamir, and Adelman (RSA) compliant encryption/decryption in accordance with TBD.

1.

1.
    1.
        1.
            4.
            5. Authentication Processing

(U) The ACM shall provide generation of a cryptographic signature for the RSA with signatures method of authentication.

(U) The ACM shall provide cryptographic signature verification for the RSA with signatures method of authentication.

(U) The ACM shall allow for configuration of shared secret authentication, in lieu of RSA with signatures method of authentication.

1.
    1.
        1.
            8.
            9. Integrity Services

The ACM must support algorithms to verify the integrity of ACM software/firmware loads, signed ACM security policies, and terminal software/firmware loads. Integrity services are also required for HAIPE and IPSec interoperability. The following requirements summarize the integrity functions that the ACM must support.

(U) The ACM shall support the hash algorithms defined in Federal Information Processing Standards Publication 180-2, Â“Secure Hash Standard (SHS)Â”, 1 August 2002.

(U) The ACM shall support the HMAC SHA1 96 integrity algorithm in accordance with RFC 2404.

(U) The ACM shall support the AES PRF 128 integrity algorithm in accordance with TBD.

(U) The ACM shall return results of integrity processing to the host.

1.
    1.
        1.
            9.
            10. Digital Signature Services

The ACM must provide digital signature services, including generation and validation, to the host for both Type 1 and non Type 1 digital signature algorithms. The following requirements summarize the digital signature functions that the ACM must support.

(U) The ACM shall support the digital signature algorithms defined in Document: Federal Information Processing Standards Publication 186-3, Â“Digital Signature StandardÂ”, 27 January 2000, Change Notices (October 2001).

(U) The ACM shall support the digital signature algorithms defined in ANSI X9.31.

(U) The ACM shall support the digital signature algorithms defined in ANSI X9.62.

(U) The ACM shall support digital signature verification for ACM software/firmware in accordance with Â“Standard for Signing and Obtaining a Hashword for a Software Package to Support INFOSEC Applications (for Signature Verification Only)Â”, KM-TG-0002-96 Rev 9, April 1997.

(U) The ACM shall return results of cryptographic authentication processing to the host.

1.
    1.
        1.
            10.
            11. Host Utility Services

(U) The ACM will be a trusted (high assurance) component within a terminal and may be the only high assurance component within a terminal. The ACM is expected to provide cryptographic utility services for ensuring that other entities within the terminal are running valid software/firmware. The following are the requirements applicable to utility services provided by the ACM to the host. An underlying assumption is that terminal software/firmware is downloaded as a package that is encrypted and signed by a designated authority.

(U) The ACM shall provide the capability to encrypt terminal software for data at rest storage using Suite A.

(U) The ACM shall provide the capability to encrypt terminal software for data at rest storage using Suite B.

(U) The ACM shall encrypt a TBD MB terminal software file within TBD seconds.

(U) The ACM shall provide the capability to decrypt and integrity check terminal software taken from data at rest using Suite A.

(U) The ACM shall provide the capability to decrypt and integrity check terminal software taken from data at rest using Suite B.

(U) The ACM shall decrypt and integrity check a TBD MB terminal software file within TBD seconds.

(U) The ACM shall provide the capability to decrypt and integrity check terminal software updates loaded during Â“on airÂ” terminal operation.

(U) The ACM shall provide file encryption/decryption service using a unique File Encryption Key (FEK) for non cryptographic software/firmware images.

<Encryption/decryption already covered; do we need separate requirements for file system encryption versus file encryption? Is there a difference?>

1.
    1.
        1.
            11.
            12. Other Services

Requirements for other cryptographic services provided by the ACM are included below.

(U) When configured for Suite A operation, the ACM shall provide the capability to decrypt Type 1 cryptographic algorithms using the JOSEKI-1 algorithm.

(U) When configured for Suite B operation, the ACM shall provide the capability to decrypt Type 1 cryptographic algorithms using the TBD algorithm.

1.
    1.
        2.
        3. Key Management

The ACM must support key management functionality for both Type 1 (Suite A and Suite B) and non Type 1 cryptographic algorithms. ACM key management requirements are complicated by several factors.

First, the future Key Management Infrastructure (KMI) will replace the existing Electronic Key Management System (EKMS) sometime during the ACM development or the follow on terminal development and integration efforts. The ACM must therefore include the functionality that is required to be a Â“KMI enabledÂ” cryptographic product. The KMI is an evolutionary system

that will field as capabilities in phased increments, and the KMI delivery schedules are likely to be fluid. For these reasons, it is expected that the ACM will need to support key fill from both the EKMS and KMI systems. There may also be terminal level reasons for this (e.g., a terminal waveform that can receive key material only from EKMS).

Second, the current goal is to develop the ACM as a fully releasable cryptographic product that can be integrated, if necessary, into a foreign partner or Homeland Defense (HLD) terminal system. It is expected that the ACM will be able to be configured as either a fully releasable cryptographic product (i.e., only Suite B) or as a non releasable cryptographic product (i.e., inclusion of Suite A). This implies that the ACM must support both the Suite A and Suite B key management operations.

Third, the goal of supporting both Type 1 and non Type 1 cryptographic services in the ACM further complicates key management. Different processes will be used to deliver Type 1 key material to the ACM compared to non Type 1 key material. The ACM must therefore support non Type 1 key management mechanisms in addition to what is required for Suite A and Suite B.

Key management functions include key agreement and key exchange, key update, key fill, key identification, key accounting, key storage and retention, and zeroization. In some cases, particular key management requirements are specified by the HAIPIS and LEF cryptographic family specifications. Key management requirements are specified in the sections that follow.

1.
    1.
        1.
            1.
            2. Key Agreement and Key Exchange

The ACM must support key agreement and key exchange functionality for both Type 1 (Suite A and Suite B) and non Type 1 cryptographic algorithms.

1.
    1.
        1.
            1.
                1.
                2. Type 1

The ACM must support the following key agreement and key exchange requirements for Type 1 key material.

(U) The ACM shall support the FIREFLY exchange in accordance with EKMS 322B in order to generate Suite A encryption keys (either for traffic or key encryption).

(U) The ACM shall perform Enhanced FIREFLY key calculation in accordance with EKMS 322 and the classified appendix to the LEF IS.

(U) The ACM shall support the Menezes-Qu-Vanstone (MQV) exchange in order to generate Suite B encryption keys.

1.
    1.
        1.
            1.
                2.
                3. Non Type 1

The ACM must support the following key agreement and key exchange requirements for Non Type 1 key material.

(U) The ACM shall support the calculation of shared secret keys using Diffie-Hellman MOD Group 2 in accordance with TBD.

(U) The ACM shall support the calculation of shared secret keys using Diffie-Hellman MOD Group 14 in accordance with TBD.

(U) The ACM shall support the calculation of shared secret keys using Diffie-Hellman Elliptic Curves in accordance with TBD.

1.
    1.
        1.
            2.
            3. Key Fill

(U) ACM key material can be filled either directly over a fill port interface to the ACM or from the host-to-ACM instruction interface. The ACM must support both RED fill and benign/BLACK fill of key material. Key material received in this manner can include a seed, operational, or one-time-use FIREFLY vector set or traditional key material. Delivery may be in RED form, BLACK form (using a pre-placed key encryption key), or utilizing benign techniques. The following requirements apply to kill fill in general. Subsequent sections capture the requirements specific to RED key fill and benign/BLACK key fill.

(U) The ACM shall provide a DS-101 key fill interface to support key fill.

Note: The ACM key fill interface may or may not be extended to the platformÂ's exterior for a particular terminal/platform integration.

(U) The ACM shall support the DS-101 protocol for key fill in accordance with EKMS 308.

(U) The ACM shall provide positive confirmation following each successful key fill.

<Positive confirmation to whom?>

(U) The ACM shall provide indication in the event of key fill failure.

<Indication to whom?>

(U) The ACM shall provide the capability to accept key and control/status information from the DS-101 fill interface.

(U) The ACM shall provide the capability to send control/status information to the DS-101 fill interface.

(U) The ACM shall support accepting key loads in accordance with EKMS 308.

1.
    1.
        1.
            1.
                1.
                2. RED Fill

The ACM must support RED fill of key material. Even if RED key fill is not required for operational systems (e.g., AEHF), the ACM must support RED fill for terminal development, testing, and integration efforts. Without this capability, a full key management system must be present at the terminal developerÂ's facility in order to load key material into the ACM (including test key material). The intent is to avoid this.

(U) The ACM shall accept RED key fill using the DS-101 protocol.

Note: It is assumed the RED key fill is only via the DS-101 interface and is not applicable to the host-to-ACM instruction interface.

(U) The ACM shall support the capability to disable support for RED key fill through software configuration.

(U) The ACM shall accept RED fill of keys with DS-100-1 key tag information in accordance with EKMS 308.

(U) The ACM shall accept RED fill of keys without DS-100-1 key tag information.

(U) The ACM shall support the loading of DoD PKI Certificates over the host instruction interface.

(U) The ACM should support the loading of DoD PKI Certificates over the DS-101 key fill interface.

(U) The ACM shall support RED fill of non Type 1 keys over the host instruction interface.

<Symmetric keys? Asymmetric keys? In accordance with what standard?>

(U) The ACM should support RED fill of non Type 1 keys over the DS-101 key fill interface.

<Symmetric keys? Asymmetric keys? In accordance with what standard?>

(U) The ACM shall cryptographically validate the integrity of the data of RED messages.

(U) The ACM shall cryptographically validate the integrity of RED key material.

1.
    1.
        1.
            1.
                2.
                3. Benign Fill

(U) The ACM must support benign fill of key material in support of the AEHF, TSAT, and TBD waveforms. Benign fill may also be applicable to the keys for other waveforms and/or for HAIPE or baseband COMSEC functions. It is also expected that the KMI system will send key material to the ACM using some form of benign fill.

(U) All the keys filled to an operational terminal in the field are expected to be in benign or BLACK form. However, to support benign/BLACK key fill, some unencrypted key material (trust anchors certificates, KEK, FF or EFF seed key material, etc.) must first be loaded. It is expected that such unencrypted key material will be loaded in a trusted facility. Subsequent to initialization, key material within the ACM may be protected using a CIK/CRK like mechanism.

(U) The ACM shall accept benign key fill using the DS-101 protocol.

(U) The ACM shall accept benign fill of keys without DS-100-1 key tag information.

Note: Some AEHF keys are benign filled that do not have a DS-100-1 tag.

(U) The ACM shall accept BLACK fill of keys without DS-100-1 key tag information.

Note: AEHF BLACK keys are not filled with a DS-100-1 tag.

(U) The ACM shall accept benign fill messages from the host instruction interface.

(U) The ACM shall support benign fill capability (in accordance with, EKMS 217, EKMS 308, EKMS 317 and EKMS 322).

(U) The ACM shall support benign fill capability in accordance with EKMS 217, EKMS 308, EKMS 317, and EKMS 322.

(U) The ACM shall support EKMS Benign Techniques in accordance with EKMS 217.

(U) The ACM shall generate keys via Firefly processing for Benign Fill.

(U) The ACM shall accept FIREFLY vector sets formatted in accordance with EKMS 322.

(U) The ACM shall accept Enhanced FIREFLY vector sets formatted in accordance with EKMS 322.

(U) The ACM should accept MAYFLY key material in accordance with TBD.

Note: Support for MAYFLY is expected in order for the ACM to be KMI compatible.

(U) The ACM should support benign fill of key material via MAYFLY processing.

Note: Support for MAYFLY is expected in order for the ACM to be KMI compatible.

(U) The ACM shall accept keys in both EKMS 317 generic fill format and DS-100-1 tagged key data formats.

(U) For AEHF, the ACM shall provide the capability to accept, decrypt, and store keys up to SECRET using Suite A benign fill.

(U) For AEHF, the ACM shall provide the capability to accept, decrypt, and store keys up to SECRET using Suite A benign fill during Â“on airÂ” terminal operation.

(U) For TSAT, the ACM shall provide the capability to accept, decrypt, and store keys up to SECRET using Suite A benign fill or Suite B benign fill.

(U) For TSAT, the ACM shall provide the capability to accept, decrypt, and store keys up to SECRET using Suite A benign fill or Suite B benign fill during Â“on airÂ” terminal operation.

(U) For CDL, the ACM shall provide the capability to handle, manage and store keys up to TOP SECRET/SECRET COMPARTMENTED INFORMATION (TS/SCI) using Suite A benign fill.

(U) For CDL, the ACM shall provide the capability to handle, manage and store keys up to TOP SECRET/SECRET COMPARTMENTED INFORMATION (TS/SCI) using Suite A benign fill during Â“on airÂ” terminal operation.

1.
    1.
        1.
            1.
                3.
                4. BLACK Fill

The ACM must support BLACK fill of key material in support of the AEHF and TBD waveform. BLACK fill may also be applicable to the keys for other waveforms. BLACK fill is distinguished from benign fill because BLACK fill implies that a key is encrypted by another existing key and benign fill implies that a key is encrypted by a key that was negotiated via a benign key agreement algorithm.

(U) All the keys filled to an operational terminal in the field are expected to be in benign or BLACK form. However, to support benign/BLACK key fill, some unencrypted key material (trust anchors certificates, KEK, FF or EFF seed key material, etc.) must first be loaded. It is expected that the user will accomplish RED key fill at a trusted facility. Subsequent to initialization, key material within the ACM may be protected using a CIK/CRK like mechanism.

(U) The ACM shall accept BLACK key fill using the DS-101 protocol.

(U) The ACM shall accept BLACK fill of keys with DS-100-1 key tag information in accordance with EKMS 308.

(U) The ACM shall accept BLACK fill of keys without DS-100-1 key tag information.

Note: AEHF BLACK keys are not filled with a DS-100-1 tag.

(U) The ACM shall accept BLACK keys from the host instruction interface.

(U) The ACM shall support the use of host selectable Key Encryption Keys (KEK) to support Black key fill.

(U) The ACM should automatically determine the correct KEK to use during a BLACK key fill.

(U) The ACM shall support BLACK key fill operations in accordance with EKMS 217 (EKMS Benign Techniques Specification).

(U) For AEHF, the ACM shall provide the capability to accept, decrypt, and store keys up to SECRET using Suite A BLACK fill.

(U) For AEHF, the ACM shall provide the capability to accept, decrypt, and store keys up to SECRET using Suite A BLACK fill during Â“on airÂ” terminal operation.

(U) For TSAT, the ACM shall provide the capability to accept, decrypt, and store keys up to SECRET using Suite A BLACK fill or Suite B BLACK fill.

(U) For TSAT, the ACM shall provide the capability to accept, decrypt, and store keys up to SECRET using Suite A BLACK fill or Suite B BLACK fill during Â“on airÂ” terminal operation.

(U) For CDL, the ACM shall provide the capability to handle, manage and store keys up to TOP SECRET/SECRET COMPARTMENTED INFORMATION (TS/SCI) using Suite A BLACK fill.

(U) For CDL, the ACM shall provide the capability to handle, manage and store keys up to TOP SECRET/SECRET COMPARTMENTED INFORMATION (TS/SCI) using Suite A BLACK fill during Â“on airÂ” terminal operation.

(U) The ACM shall provide the capability to accept, process, and store unencrypted Suite A key material on a fill port interface to support crypto initialization.

(U) The ACM shall provide the capability to accept, process, and store unencrypted Suite B key material on a fill port interface to support crypto initialization.

(U) The ACM shall cryptographically validate the integrity of the data of BLACK messages.

(U) The ACM shall cryptographically validate the integrity of BLACK key material.

(U) The ACM shall accept BLACK key material encrypted in ACCORDION version 3.0 in accordance with TBD.

(U) For AEHF, the ACM shall accept BLACK key material encrypted in ACCORDION version 1.3 in accordance with TBD.

1.
    1.
        1.
            3.
            4. Key Identification

The ACM must be able to identify keys and must be able to process information such as key classification from the key tag for a key. The ACM must also be able to provide key tagging information to the host. Also, in order to provide flexibility and the capability of supporting different waveforms with different key tagging standards, the ACM should provide the capability for the host to append key tagging information to a key.

(U) The ACM shall provide an interface through which information (e.g., update count, key tag, description) about one or more keys may be requested and provided.

(U) The ACM shall identify keys using the DS-100-1 key tag standard in accordance with EKMS 308.

(U) The ACM shall identify keys using the AEHF key tag format in accordance with the AEHF key tag specification.

(U) The ACM shall be capable of binding key tagging and ID information to a key.

(U) The ACM should provide the capability for the host to append information to the DS-100-1 key tagging information.

(U) The ACM should provide the capability to incorporate key tag format changes that may be defined by updates to EKMS 308 or other TBD specifications.

(U) The ACM shall provide capability to output key tagging and ID information to the host.

1.
    1.
        1.
            4.
            5. Key Allocation and Usage

The ACM must enforce some security policies for keys based upon the identification information that is bound to the key.

(U) The ACM shall instantiate the use of keys according to their tag or ID information.

(U) The ACM shall enforce the security policy that specifies the classification level of each channel based on the classification of the RED Interface of the cryptographic channel.

(U) The ACM shall ensure that keys are instantiated for a specific application. For the purposes of this requirement, a Â“specific applicationÂ” of a key is:

1. Using the specified classification key

2. Using the specified key type (e.g., TEK, KEK, TSK, etc)

3. Using a key with the specified algorithm (e.g., a KEESEE key with a KEESEE algorithm)

4. Using the key for the specified compartment (e.g., US, NATO, Allied)

5. Using key with a specified waveform

6. Using key with specified instance of the waveform

<Recommend splitting these out to individual requirements MOVE TO POLICY SECTION>

(U) The ACM shall test the keys for integrity at instantiation.

1.
    1.
        1.
            5.
            6. Key Accounting and Audit

In general, the ACM is not required to maintain accountability and audit records for key material as it is loaded, utilized, and destroyed (TBD). The following requirements do apply.

(U) The ACM shall maintain the identification of cryptographic key holdings correlated with waveform, algorithm, and key data.

(U) The ACM shall maintain the identification of keys currently loaded.

1.
    1.
        1.
            6.
            7. Key Storage

The following requirements apply to storage of key material within the ACM.

(U) The ACM shall provide a minimum storage capacity of TBD MB of working memory for key material including the information that is bound to the key (i.e. key tag, effective date, etc).

Note: Key material includes TEKs, KEKs, TSKs, FEKs, FIREFLY Vectors and associated information, public/private key pairs, and key splits.

(U) The ACM shall store persistent keys in BLACK form.

(U) The ACM shall provide a minimum storage capacity of TBD MB of persistent storage for key material including the information that is bound to the key (i.e. key tag, effective date, etc).

Note: Key material includes TEKs, KEKs, TSKs, FEKs, FIREFLY Vectors and associated information, public/private key pairs, and key splits.

(U) The ACM shall bind keys to their respective identification information in storage.

(U) When configured for Suite A operation, the ACM shall encrypt keying material for storage using ACCORDION.

(U) When configured for Suite B operation, the ACM shall encrypt keying material for storage using AES Key Wrap.

(U) The ACM shall encrypt updated keys for storage.

(U) The ACM shall provide the capability to store DoD PKI private keys.

(U) The ACM shall protect keys stored within ACM.

1.
    1.
        1.
            7.
            8. Key Update

The ACM must support key update functions for certain algorithms. The following requirements apply to ACM key update functions.

(U) The ACM shall perform deterministic update of a TEK used to protect MEDLEY encrypted traffic using ACCORDION 3.0 in accordance with R21-TECH-03-02, Â“NSA MEDLEY Implementation Standard: An ACCORDION MEDLEYÂ”, 7 Feb 2002.

(U) The ACM shall perform deterministic update of a TEK used to protect BATON encrypted traffic using ACCORDION 1.3 in accordance with KM-TG-0001-87, Â“ACCORDION 1.3Â”, 30 Oct 1987.

(U) The ACM shall perform deterministic update of a TEK using ACCORDION 3.0.

(U) The ACM shall perform deterministic update of a Suite B TEK using the algorithm specified in R21-TECH-02-05, Â“R21 Information Technical Report dated 10 January 2005: A Key Update Function Based on the AES Key WrapÂ”, 10 Jan 2005.

(U) The ACM shall provide a key update function, as applicable, for each supported cryptographic algorithm.

(U) When configured for Suite A operation, the ACM shall provide a key update using ACCORDION for US only missions.

1.
    1.
        1.
            8.
            9. Key Rollover

Support for HAIPE requires the ACM to provide the capability to do a rollover from one key variable to another. The following requirements apply to ACM key rollover functions.

(U) The ACM shall provide a function for the host to command a rollover from one key to another.

(U) The ACM shall support a rollover command from the host.

(U) The ACM shall support rollover as specified by the performance documents of the waveform being supported.

1.
    1.
        1.
            9.
            10. Key Zeroization

The ACM must support different types of key zeroization. Zeroization requirements are discussed in the sections that follow.

1.
    1.
        1.
            1.
                1.
                2. Selective zeroization

Selective zeroization is a process through which the host can selectively zeroize one or more keys within the ACM. Selective zeroization applies to both Â“workingÂ” key material and key material in persistent storage.

(U) The ACM shall provide for the selective zeroization of Type 1 keys when commanded on the Cryptographic Control Interface.

(U) The ACM shall provide for the selective zeroization of private keys when commanded on the Cryptographic Control Interface.

(U) The ACM shall provide for the selective zeroization of trust anchors and/or public keys when commanded on the Cryptographic Control Interface.

(U) The ACM shall provide for the selective zeroization of key split material when commanded on the Cryptographic Control Interface.

1.
    1.
        1.
            1.
                2.
                3. Recoverable zeroization

Recoverable zeroization is a process through which all application key material within the ACM is zeroized, but ACM specific key material is left intact. ACM specific key material is intended to mean key material that would be used to fill new application key material into the ACM. An ACM does not have to be reinitialized after a recoverable zeroization has been done

1.
    1.
        1.
            1.
                3.
                4. Destructive zeroization

Destructive zeroization is a process through which the ACM is completely zeroized and all key material within the ACM is rendered unrecoverable. An ACM must be reinitialized after a destructive zeroization has been done.

(U) The ACM shall provide a message to the host that the ACM is in the destructive zeroize state.

(U) The ACM shall accept a hardware discrete destructive Zeroize signal.

(U) The ACM shall, upon detection of the assertion of the destructive Zeroize hardware discrete line signal, zeroize the RED keys and internal active key splits.

(U) The ACM destructive zeroization function shall function in all states and modes, with or without prime power or terminal battery backup power.

1.
    1.
        1.
            1.
                4.
                5. OTAZ

Over-the-Air Zeroization (OTAZ) messages may or may not be processed directly by the ACM. Instead, an OTAZ command may first be sent to the host system. After authenticating the OTAZ message, the host would Â“actionÂ” the OTAZ message by executing the appropriate zeroization commands (e.g., selective, recoverable, or destructive zeroization) using its control interface with the ACM. Future systems, or possibly the future KMI, may send OTAZ type messages directly to the appropriate ACM that are not processed/interpreted first by the host system. It is desirable for the ACM design and functionality to be able to be extended to allow direct processing of OTAZ messages. The following objective requirements summarize the flexibility that is desired for OTAZ functionality within the ACM.

(U) The ACM should receive, cryptographically authenticate, and process and respond to Over the Air Zeroize (OTAZ) messages.

Note: This does not imply that the ACM has an Â“over-the-airÂ” interface. The term OTAZ is used to indicate a remote zeroization command. A remote zeroization command is considered to be any zeroization command that originates from a source other than the host.

(U) The ACM should cryptographically authenticate remote zeroization commands prior to execution in accordance with the ACM security policy.

(U) The ACM should provide the capability to zeroize all RED keys and internal active key splits, in accordance with the ACM security policy, when an authenticated OTAZ message is received.

(U) The ACM should provide the capability to zeroize the RED keys and other key material associated with a specific waveform/application.

Note: This allows for the capability to action an authenticated OTAZ message that is limited to a specific waveform or application.

1.
    1.
        1.
            10.
            11. OTAR

Over-the-Air Rekey (OTAR) messages may or may not be processed directly by the ACM. Instead, an OTAR message may first be sent to the host system. In this case, the host would format the OTAR message as required for the ACM and load the key(s) received via the OTAR using its control interface with the ACM. Future systems/waveforms may require OTAR processing within the ACM itself. It is desirable for the ACM design and functionality to be able to be extended to allow direct processing of OTAR messages. The following requirements summarize the flexibility that is desired for OTAR functionality within the ACM.

(U) The ACM shall support Over-the-Air-Rekey (OTAR), as required, by the configured waveforms/applications.

(U) The ACM should provide the capability to cryptographically authenticate OTAR messages.

(U) The ACM should be capable of providing notification that an OTAR initiated transaction has occurred.

(U) The ACM should provide the capability to cryptographically authenticate Over-the-Air-Transfer (OTAT) messages as required by the configured waveforms for waveform specific keys/data.

1.
    1.
        3.
        4. Cryptographic Modernization

The ACM must satisfy the requirements of the Cryptographic Modernization Initiative (CMI) in accordance with NSA policy number 3-9, Â“Cryptographic Modernization Initiative Requirements for Type 1 Cryptographic ProductsÂ”. Some of the CMI requirements, such as the requirement to implement Suite A and/or Suite B modernization algorithms, are discussed elsewhere in this TRD. This section organizes the additional CMI requirements that are not discussed elsewhere.

1.
    1.
        1.
            1.

2. Configurability

The ACM must support configurable capabilities in order to be useful for different terminal/platform types. Some platforms are not required to provide high or multi-channel throughput capability, but low power consumption is critical. Ideally, the ACM would be configurable simply via software/firmware. This is not the definition of Â“configurabilityÂ”, however, because it is recognized that additional components may be required for the full ACM capability and fewer components may be required for a low power ACM capability.

(U) The ACM should be reconfigurable over the complete range of ACM capabilites via software/firmware load.

(U) The ACM should be configurable for different levels of aggregate throughput from 100 Kbps to 10 Gbps (objective). Move to worst case section and make part of intro.

(U) Different configurations of the ACM to support varying levels of capability and/or power consumption shall share a common control and management scheme.

<I think this HAS to be a threshold versus a Â“shouldÂ”>

(U) Different configurations of the ACM to support varying levels of capability and/or power consumption shall implement a common design.

1.
    1.
        1.
            2.
            3. Programmability

The ACM must be programmable in a tactical environment. The ACM must be capable of changing crypto algorithms and keys based on the mission to be supported (e.g., Coalition interoperable vs. US only). Programmability includes general requirements related to the architecture of the ACM and functional requirements for cryptographic software/firmware loading, identification, storage, accounting, and zeroization. Programmability requirements are specified in the sections that follow.

1.
    1.
        1.
            1.
                1.
                2. Programmable Architecture

The following requirements are applicable to the general architecture of the ACM. The overall ACM architecture is an important consideration in the programmability of the device. It is expected that the ACM will be implemented using General Purpose Processor (GPP) and/or Field Programmable Gate Array (FPGA) devices. These devices are inherently reprogrammable, offer implementation flexibility, and provide lower cost, schedule, and capability risks for development than an Application Specific Integrated Circuit (ASIC). This TRD does not abrogate the use of an ASIC for the implementation of the ACM as long as all requirements can be satisfied. In order to support CMI goals, the ACM should be completely reprogrammable in order to support algorithms and cryptographic services required by future waveforms. A GPP/FPGA implementation allows for new and updated algorithms to be installed without hardware upgrade, and may be preferably evaluated.

(U) If the ACM implementation utilizes FPGA technology, in whole or in part, the following requirements are applicable:

(U//FOUO) The classified FGPA images shall be saved in BLACK form when ACM is not operational.

(U//FOUO) The classified FPGA images shall be decrypted and loaded only when ACM is operational.

(U) If the ACM implementation utilizes ASIC technology, in whole or in part, the following requirements are applicable:

(U//FOUO) The ASIC shall be fabricated in a DoD approved Trusted Foundry.

(U//FOUO) An ASIC implementing classified algorithm shall require the use of NSA approved method(s) that allows the ASIC to be handled as a COMSEC Controlled Item (CCI).

1.
    1.
        1.
            1.
                2.
                3. Cryptographic Software/Firmware Loading

The following requirements are applicable to the functionality that the ACM must provide in order to support (re)programmability and field upgrade of the ACM software/firmware. In the requirements that follow, the implementation of a Type 1 cryptographic algorithm is assumed to be part of the ACMÂ's software/firmware that is encrypted and signed by NSA and not a standalone software/firmware package. It is assumed that NSA requires that all security-related software/firmware components that will be executed on the ACM must be bundled together in a

single package and evaluated as a whole. Under these assumptions, a set of algorithms would be packaged together based on typical mission requirements (e.g., US only versus foreign interoperable). These assumptions should not preclude novel alternative implementations that are advantageous and capable of NSA certification.

(U) The ACM shall provide field upgrade and reprogramming of the cryptographic software/firmware.

(U) The ACM shall support the use of Trust Anchors for the purpose of providing signature verification for software/firmware downloads.

(U) The ACM shall only accept cryptographic software/firmware packages signed by NSA.

(U) The ACM should be capable of supporting the cryptographic algorithms for the evolving requirements for upgrading of software/firmware using the Cryptographic Message Syntax (CMS) for protecting firmware packages for Type1 Cryptographic Modules (Objective).

(U) The ACM should be capable of supporting the cryptographic algorithms for the evolving standards of the Trust Anchor Management Protocol for use with Type 1 Cryptographic Modules (Objective).

(U) The ACM shall perform authentication and integrity checks of received cryptographic software/firmware packages.

(U) The ACM shall report the result of a failed integrity check for received cryptographic software/firmware as an auditable event.

(U) The ACM shall be capable of being configured to not accept a version of a cryptographic software/firmware package that is older than the currently installed version.

(U) The ACM shall maintain the type and version of the cryptographic software/firmware package(s) that is (are) loaded.

(U) The ACM shall provide the capability to accept and store Suite A or Suite B cryptographic algorithms up to TOP SECRET using over-the-network algorithm update during Â“on airÂ” terminal operation

(U) The ACM shall be capable of being software/firmware upgraded and reconfigured without the need to be returned to the factory, depot, or a trusted facility.

(U) The ACM shall be delivered complete with all test equipment, harnesses and software tools needed for Depot support.

1.
    1.
        1.

1.
3.
4. Cryptographic Algorithm Identification

The host and the ACM must be able to identify the algorithms that are implemented within the ACM software/firmware. The following requirements apply to algorithm identification.

(U) The ACM shall provide an interface through which algorithm information (e.g., supported algorithms, supported modes, version, etc.) may be requested and provided.

(U) The ACM should identify algorithms in accordance with TBD.

1.
1.
1.
1.
4.
5. Cryptographic Software/Firmware Storage

The following requirements are applicable to storage of cryptographic software/firmware within the ACM.

(U) The ACM shall encrypt cryptographic software/firmware for storage.

Note: It is assumed that the algorithm that is used to encrypt cryptographic software/firmware for storage can be different than the algorithm that is used to encrypt cryptographic software/firmware for distribution. It is desirable for cryptographic software/firmware to be encrypted for distribution using a mechanism that does not require management of individual decryption symmetric key segments for each ACM.

(U) The ACM shall decrypt stored cryptographic software/firmware.

Note: It is assumed that the algorithm that is used to decrypt stored cryptographic software/firmware can be different than the algorithm that is used to encrypt cryptographic software/firmware for distribution. It is desirable for cryptographic software/firmware to be encrypted for distribution using a mechanism that does not require management of individual decryption symmetric key segments for each ACM.

(U) The ACM shall perform integrity checks on Security Critical Packages and files prior to instantiation or use.

(U) The ACM shall have the capability to store a minimum of (TBD) different Cryptographic software/firmware images in BLACK form.

<ShouldnÂ't this be specÂ'd as a storage size instead of a number of packages?>

(U) Algorithm information (e.g., algorithm type, supported modes, version, etc.) shall be bound to the algorithm in storage.

(U) The ACM shall have the capability to maintain at least two versions for each cryptographic software/firmware image.

<This one needs rationale>

(U) The ACM shall have the capability to rollback the version of the cryptographic software/firmware image.

1.
    1.
        1.
            1.
                5.
                6. Cryptographic Software/Firmware Erasure

The following requirements are applicable to erasure/zeroization of cryptographic software/firmware within the ACM.

(U) The ACM shall have the capability to overwrite the previous version of the stored cryptographic software/firmware when a new version of the software/firmware is downloaded and after its authenticity and integrity has been verified.

1.
    1.
        1.
            3.
            4. Releasability

(U//FOUO) The ACM shall be capable of supporting a programmable Positive Access Control (PAC) ing and privileges to support foreign releasability.

(U) The ACM shall support Type 1 Suite B algorithms required for foreign interoperability, when required.

1. 1. 1. 4. 5. Cryptographic Family Interoperability

1. 1. 1. 1. 1. 1. 2. HAIPIS

<Recommend that we need to get specific about which HAIPIS extension specifications are required and which are not (if any are not)>

The ACM shall provide HAIPE cryptographic services in accordance with appendix B.

Note: The requirements traceability matrix in Appendix B indicates the allocation of requirements between the ACM and the host.

1. 1. 1. 1. 2. 3. LEF

(U) The ACM shall support the cryptographic requirements for Link Encryptor Family LEF 2.0.

The ACM shall provide LEF cryptographic services in accordance with appendix C.

Note: The requirements traceability matrix in Appendix C indicates the allocation of requirements between the ACM and the host.

1. 1. 4. 5. Software Communications Architecture (SCA)

(U) The host terminals will be SCA compliant terminals. The goals established by the SCA are to increase software portability, increase software reuse and increase use of COTS software. This typically applies to both the RED side and BLACK side of the host terminal. The SCA uses the Common Object Request Broker Architecture (CORBA), which is considered to provide a low level of assurance, in order to provide an environment for software portability and reuse. While the ACM may use a non SCA compliant internal implementation in order to provide the necessary high assurance, it must provide an SCA compliant host interface on both the RED and BLACK sides.

(U) The SCA includes a security supplement that establishes security requirements for SCA compliant systems. The SCA security supplement includes both system level (terminal level) requirements and requirements on the cryptographic subsystem. Because the host terminals will be SCA compliant terminals, the ACM must satisfy the requirements from the SCA security supplement that are allocated to the cryptographic subsystem.

(U) [CR0000] The ACM shall support the requirements listed in section 4.2 of the Security Supplement to the SCA Specification, JTRS-5000SEC, rev 3.0.

(U) [CR0000] The ACM shall provide a native and/or software adaptor interface that implements the SCA Security API Service Definition, as defined in Attachment 1 to JTRS-5000SEC, rev 3.0.

(U) [CR0000] The SCA interface to the ACM shall support all ACM functions otherwise available through any other interfaces.

(U) [CR0000] ACM functions invoked through the SCA interface will satisfy the performance requirements (e.g., encrypt/decrypt rates, latency, etc.) specified elsewhere in this document.

1.
    1.
        5.
        6. Lifecycle

The following requirements are applicable to the life cycle processes for the ACM and for the cryptographic services supported by the ACM. Elements of the ACM lifecycle include ACM startup and ACM shutdown. Elements of the lifecycle for cryptographic services include algorithm initialization, cryptographic channel instantiation, cryptographic channel runtime, and cryptographic channel termination.

1.
    1.
        1.
            1.
            2. ACM Startup

(U) The ACM shall perform startup in accordance with the tailored Unified INFOSEC Criteria (UIC) for POET.

(U) Startup refers to the boot and function initialization processes of the ACM. The following requirements apply to the ACM startup process.

(U) The ACM shall provide services to decrypt, verify, install, and execute its run-time application.

(U) The ACM run-time application shall not execute if integrity verification fails.

(U) The ACM shall provide notification to the host when it is operationally ready to provide services to the host.

(U) Notification shall include a status output indicating that boot and test were completed without error or indicate that an error occurred and provide error information.

(U) The ACM shall be operational within 5 seconds after power on.

(U) The ACM shall be designed to restart properly independent of the previous shutdown (orderly shutdown or unscheduled shutdown).

1.
    1.
        1.
            2.
            3. ACM Shutdown

(U) The ACM shall perform shutdown in accordance with the tailored Unified INFOSEC Criteria (UIC) for POET.

(U) Shutdown refers to the process of executing a shutdown operation for the ACM. Shutdown could be a scheduled (orderly) shutdown or an unscheduled shutdown (e.g., removal of power). The following requirements apply to the ACM shutdown process.

(U) (U) The ACM shall be capable of shutting down and reverting to a CCI state within TBD seconds after a receipt of a shutdown command.

1.
    1.
        1.
            3.

4. Algorithm Initialization

(U) Algorithm initialization refers to the process of starting an algorithm to provide a cryptographic service to the host. The following requirements apply to the algorithm initialization process.

(U) The ACM shall provide the capability to instantiate cryptographic algorithms with different modes/options supported by the algorithm.

(U) The ACM shall decrypt and instantiate algorithms prior to use.

(U) The ACM shall test instantiated cryptographic algorithms for correct function prior to operational use.

1.
    1.
        1.
            4.
            5. Cryptographic Channel Instantiation

(U) In the following requirements, a cryptographic channel is defined as a cryptographic service provided to the host system that is associated with a particular key. A cryptographic channel may be used to support TRANSEC key stream generation, COMSEC encryption/decryption, or any of the other cryptographic services supported by the ACM. The ACM performs cryptographic operations on a cryptographic channel as allowed by the ACM security policy. The following requirements apply to cryptographic channel instantiation.

(U) The ACM shall perform cryptographic channel instatiation in accordance with the tailored Unified INFOSEC Criteria (UIC) for POET.

(U) The ACM shall verify against the resident security policy that there are no policy violations prior to cryptographic channel instantiation.

(U) The ACM shall instantiate cryptographic functions for a cryptographic channel within TBD seconds of the request to instantiate the channel.

1.
    1.
        1.
            5.
            6. Cryptographic Channel Run-time

(U) The following requirements apply to cryptographic channels during runtime processing on the cryptographic channel.

(U) The ACM shall provide the ability to perform periodic tests to assure that instantiated ACM operations on a cryptographic channel are maintained.

(U) The ACM shall provide cryptographic channel status outputs on request.

(U) The ACM shall provide the capability to change a cryptographic channel or the cryptographic channel operating parameters without affecting the operation of other cryptographic channels.

(U) The ACM shall provide the capability to turn on/off on a cryptographic channel without affecting the operation of other cryptographic channels.

1.
    1.
        1.
            6.
            7. Cryptographic Channel Termination

(U) The following requirements apply to cryptographic channel termination. Termination includes the cases of routine termination of a cryptographic channel and abnormal termination.

(U) In the event of internal failure, the ACM shall terminate the cryptographic channel in accordance with the tailored UIC for POET.

(U) The ACM shall provide abnormal cryptographic channel failure status as audit event to the host.

(U) In the event of abnormal termination of a cryptographic channel, the ACM shall be capable of supporting the operation of the remaining cryptographic channels.

1.
    1.
        6.
        7. Levels of Security and Classification

(U) The following requirements clarify the different levels of security that must be supported by the ACM and the requirements for supporting different levels of security concurrently.

(U) The ACM shall support cryptographic operations using unclassified, SECRET, or TOP SECRET key material.

(U) The ACM should support cryptographic operations using unclassified, CONFIDENTIAL, SECRET, or TOP SECRET key material.

(U) The ACM should support cryptographic operations using SECRET compartmented key material.

(U) The ACM should support cryptographic operations using TOP SECRET compartmented key material.

(U) The ACM shall support unclassified and SECRET level cryptographic services simultaneously.

(U) The ACM should support unclassified, CONFIDENTIAL, SECRET, and TOP SECRET level cryptographic services simultaneously.

(U) The ACM should support collateral and compartmented cryptographic services simultaneously.

(U) The ACM shall support TOP SECRET/Sensitive Compartmented Information (TS/SCI) cryptographic services in a system-high configuration.

1.
    1.
        7.
        8. ACM Security Policy

(U) A security policy is a set of rules which are tested for compliance and then enforced by a given security function. Security policies are expected to be different for different waveforms and/or applications that the crypto must support. Security policies are expected to be created by the terminal vendor. TBD may be used as the format for the security policy definition.

(U) The ACM shall protect the security policy rules for cryptographic bypass within the ACM.

(U) The ACM shall accept waveform security policy elements from the Cryptographic Control Interface prior to cryptographic channel instantiation.

(U) The ACM shall authenticate the cryptographic signature of each security policy.

(U) The ACM shall provide security policy status upon authenticated request.

(U) The ACM shall authenticate the signature of each security policy prior to use.

(U) The ACM shall enforce the internal security policy for security critical items prior to reporting status.

(U) The ACM shall report any attempted violation of the security policies as audit events.

1.
    1.
        8.
        9. Bypass Processing

(U) The following requirements are for high assurance cryptographic bypass processing that must be supported by the ACM. Cryptographic bypass processing includes both cryptographic channel bypass and control/status bypass. There is also a waveform specific bypass function that must be supported for the CDL waveform for terminals with a RED core architecture. There may also be waveform specific bypass functions that must be supported for other waveforms. This highlights the fact that the ACM bypass policy must be flexible as well as secure and capable of NSA certification.

1.
    1.
        1.
            1.
            2. Cryptographic Channel Bypass

The following requirements are applicable to cryptographic channel bypass. Cryptographic channel bypass is defined as the bypassing of information from the plaintext to the ciphertext side of the ACM for information that will eventually egress the terminal on an external interface.

(U) The ACM shall perform traffic header bypass for packet encryption/decryption cryptographic channels in accordance with the cryptographic channel bypass security policy for the cryptographic channel.

(U) The ACM shall perform waveform specific bypass functions in accordance with the cryptographic channel bypass security policy for the waveform.

(U) The ACM shall have a mechanism to terminate communication on the cryptographic channel if the bypass policy is violated.

(U) If policy violation exceeds a policy-determined threshold, the ACM shall support a mechanism to terminate the channel, resulting in termination of the operation of the affected channel.

(U) The ACM bypass mechanism shall block messages that violate the bypass security policy.

1.
    1.
        1.
            2.
            3. Control/Status Bypass

The following requirements are applicable to control/status bypass. Control/status bypass is defined as the bypassing of information from the plaintext to the ciphertext side of the ACM for internal control/status information that will not egress the terminal on an external interface. A certain amount of control/status bypass is expected for any terminal architecture that includes both RED side and BLACK side processing functions.

(U) The ACM shall support control/status bypass in accordance with the control/status bypass security policy.

(U) The ACM control/status bypass mechanism shall be non-bypassable.

(U) The ACM control/status bypass mechanism shall always be invoked.

(U) The ACM bypass mechanism shall block messages that violate the bypass security policy.

(U) The ACM control/status bypass mechanism algorithm shall check for valid format of bypass messages.

(U) The ACM control/status bypass mechanism algorithm shall check for valid length of bypass message.

(U) The ACM control/status bypass mechanism algorithm shall check for valid frequency of bypass messages for a given bypass mechanism.

1.
    1.
        1.
            3.
            4. CDL Bypass

(U//FOUO) A waveform specific bypass function must be supported for the CDL waveform for terminals with a RED core architecture. Legacy CDL communications require link control information to be exchanged between the two communicating systems via encrypted transmissions in order to securely aid in signal acquisition as well as to maintain directional links

with moving platforms. On the legacy CDL waveforms, link control information is exchanged by multiplexing it together with the end-to-end traffic on the CDL link and then bulk encrypting the multiplexed signal. The end-to-end traffic may or may not be encrypted separately before bulk encryption. A RED core terminal can support CDL links where the end-to-end traffic is separately encrypted. After bulk decryption, the end-to-end traffic can be bypassed to the BLACK side of the terminal for transmission to the next hop on the end-to-end path. This is a type of bypass function that must be supported by the ACM. This type of bypass is discussed separately because it is specific to the CDL waveform and because of the high data rates that are involved. Below are requirements for the CDL specific bypass.

(U) The ACM shall provide a security guard function restricting CDL plaintext control information to the RED side.

(U) The ACM shall provide a high speed control bypass function commensurate with the <maximum> CDL waveform data rate to allow for the transfer of Bulk decrypted data from RED side to the Black side.

1.
    1.
        1.
            4.
            5. Network Management Bypass

(U//FOUO) The TSAT waveform, and potentially all network centric waveforms, may require BLACK side terminal functionality to be managed via a Type 1 security association (TBD). Since management commands/messages would terminate on the RED side of the terminal, there must be some RED-to-BLACK and BLACK-to-RED bypass in order to support this type of management. This type of bypass is discussed separately, but may only be a particular case of control/status bypass. The fact that network management bypass is discussed separately should not be interpreted to mean that this must be a unique bypass mechanism in addition to control/status bypass. Below are requirements for network management bypass.

(U) The ACM shall support bypass of network management data from the RED side to the BLACK side in accordance with the bypass security policy.

(U) The ACM shall support bypass of network management data from the BLACK side to the RED side in accordance with the bypass security policy.

1.
    1.
        9.
        10. Unattended Operation

(U) The ACM must support terminals that operate unattended in the ground tactical and airborne environments. Unattended operation requires that the ACM be activated using a physical token and/or pass phrase and remain activated when the token is removed. Unattended operation also requires a robust anti-tamper mechanism and ability to perform remote zeroization of the ACM.

(U) The ACM shall be capable of supporting the use of CRK over the host interface for initial setup.

(U) The ACM shall be capable of supporting the use of host provided CRK equivalent variables for initial setup

(U) The ACM shall be capable of supporting unattended operation after the initial setup.

(U) The ACM shall allow configuration through a remote operator interface.

(U) The ACM shall provide a cryptographically authenticated mechanism for loading the requisite key information for remote re-activation without the use of a CRK.

(U) The ACM will be used in ground terminals/platforms that operate unattended at the unclassified or SECRET level. POET will not support unattended ground terminals/platforms at the TOP SECRET level.

1.
    1.
        10.
        11. Unclassified Handling

(U) The ACM shall provide the capability to be rendered unclassified for storage/shipping.

(U) The ACM shall support the means to disable classified processing capabilities (i.e., transition to an Unclassified Controlled Cryptographic Item (CCI)).

(U) The ACM shall support the recovery from Unclassified CCI to Classified Operation state.

(U) The ACM shall require the CRK to be physically attached to the CRK interface.

(U) The ACM shall have exclusive access to the physical CRK, when present.

1.
    1.
        11.
        12. Tamper

(U) The ACM shall support tamper in accordance with the tailored Unified INFOSEC Criteria (UIC) for POET.

(U) The ACM shall be provided with field tamper recovery capability.

(U) ACM anti-tamper capabilities shall be available in the absence of external power.

(U) ACM zeroization capabilities shall be available in the absence of external power.

1.
    1.
        12.
        13. Identification and Authentication (I&A)

1.
    1.
        13.
        14. Audit

The ACM shall provide audit capabilities in accordance with the tailored Unified INFOSEC Criteria (UIC) for POET.

(U) The ACM shall be capable of reporting audit events over the Cryptographic Control Interface.

(U) The ACM shall support TBD of internal storage for audit events.

(U) The ACM control/status bypass mechanism shall output all violations of bypass security policy as audit events.

(U) The ACM should support recording of audit messages as required by KMI.

(U) The ACM should support reporting of audit messages, as required by KMI, via the fill port interface or the host instruction interface.

1.
    1.
        14.
        15. Alarm

The ACM shall provide alarm capabilities in accordance with the tailored Unified INFOSEC Criteria (UIC) for POET.

(U) The ACM shall be capable of detecting faults and provide alarm events.

(U) The ACM shall be capable of supporting segregation of major and minor alarms events.

(U) The ACM shall be capable of recovery from compromise events (e.g. loss of key, equipment failure).

(U) The ACM shall default to a secure state in the event of a failure.

1.
    1.
        15.
        16. Built-In Test (BIT) and Health Status

The ACM shall provide BIT and health status capabilities in accordance with the tailored Unified INFOSEC Criteria (UIC) for POET.

1.
    1.
        1.
            1.
            2. BIT

(U) The ACM BIT function can be categorized as follows:

-
- Power-on BIT (PBIT)
- Continuous BIT
- Initiated BIT

(U) The ACM BIT function shall detect no less than 95 percent of all faults, by failure rate.

1.
    1.
        1.
            1.
                1.
                2. Power-on BIT

(U) The ACM shall perform PBIT upon reset or power-up.

(U) The ACM PBIT shall include test for each type (volatile, non volatile) of memory.

(U) The ACM PBIT shall include tests for all processors and FPGAs.

1.
    1.
        1.
            1.
                2.
                3. Continuous BIT

(U) The ACM CBIT function shall detect no less than 95 percent of all tested fault conditions by failure rate.

(U) The ACM CBIT shall monitor the battery voltage of the internal battery.

1.
    1.
        1.
            1.
                3.
                4. Initiated BIT (IBIT)

(U) The ACM shall support IBIT function in an offline state.

(U) The ACM shall have the capability to perform IBIT when requested from the Cryptographic Control Interface.

(U) The ACM shall complete IBIT within TBD seconds after IBIT has been initiated.

(U) Upon request, the ACM shall perform additional diagnostics during the terminal initialization sequence. These diagnostics will include known answer tests (to verify the encrypt/decrypt path) and tests that verify the correct operation of the cryptographic bypass.

1.
    1.
        1.
            2.
            3. ACM Health Status

(U) The ACM shall provide a Health discrete signal and BIT status messages for reporting health status.

(U) The ACM shall provide the results of the diagnostic tests.

(U) The ACM shall provide the overall POET ACM version identifier.

1.
    4.
    5. External Interfaces

(U) Industry standards shall be used, where possible, for external interfaces.

(U) The ACM shall accept a PPS (pulse-per-second) clock.

(U) The ACM shall include a DS-101 Cryptographic fill port interface IAW EKMS-308.

(U) The ACM shall accept a RESET signal

(U) The ACM shall accept a Tamper signal

(U) The ACM shall provide an interface to support connection to a CRK interface.

(U) The ACM shall accept a destructive Zeroize hardware signal.

(U) The ACM shall provide a Health signal.

(U) The ACM shall receive RED power from the host.

(U) The ACM shall receive BLACK power from the host.

(U) The ACM shall accept the Time of the Day.

.

(U) The ACM shall accept the Date Time from the host in a TBS format.

1.
    5.
    6. Design margins

(U) The ACM shall provide TBD design margin reserve capacity to accommodate new cryptographic algorithms and services for alternate/future waveforms.

1.

1.
    1.
    2. Processor Capacity

(U) The ACM shall have 50 percent of processor capacity reserve required for the worst case scenario, when it is configured with cryptographic channels for TRANSEC channels for four waveforms, Bulk Encryption/Decryption for CDL waveform and supporting multiple HAIPE tunnels for NMS.

1.
    1.
        2.
        3. Memory Capacity

(U) The ACM shall have 100 percent memory capacity reserve required for the worst case scenario, when it is configured with cryptographic channels for TRANSEC channels for four waveforms, Bulk Encryption/Decryption for CDL waveform and supporting multiple HAIPE tunnels for NMS.

1.
    6.
    7. Power

1.
    1.
        1.
        2. General

The following requirements are general requirements related to the power required to operate the ACM within a terminal/platform.

(U) Some platforms are not required to provide high or multi-channel throughput capability but low power consumption is critical. Reasonable means to conserve power shall be employed, including reducing ACM power consumption when high performance is not needed.

(U) The ACM shall scale its power requirements in relation to the configured throughput and channels supported, lower throughput using less power.

(U) Reasonable means to conserve power shall be employed, including reducing ACM power consumption when high performance is not needed.

(U) The POET ACM shall be configurable for the number of independent channels supported from one to four. The POET ACM shall scale its power requirements in relation to the configured throughput and channels supported, lower throughput using less power.

(U) The ACM shall receive a RED external power of TBD VDC.

(U) The ACM shall receive BLACK external power of TBD VDC.

(U) The ACM shall consume power less than TBD watts.

(U) The ACM shall locally generate any additional internal voltages required.

(U) The ACM power consumption shall be defined in its interface control document so that the host terminal can provide sustaining voltage in the event of an external un-commanded power interruption. The ACM full load operating conditions shall be used to derive this parameter. Given the ACM scalability requirements, several sets of power interface specifications may be necessary to characterize performance.

1.
    1.
        2.
        3. Battery

(U) The ACM shall include a hold-up battery that will detect the absence of external power and transition the device into its cryptographic safe mode as specified in the UIC. It is desired that this battery be capable of being recharged from the host when external power is available in order to maximize field service life. The ACM shall provide an output status to the terminal when its battery reaches 85 percent of its useful life. In the absence of external power under nominal room ambient temperature conditions the ACM battery shall have a minimum installed shelf life of not less than 10 years to provide UIC specified capabilities. The ACM battery must deliver required UIC capability over the full range of temperature and shock exposure but the shelf life under these extreme conditions shall be reduced to no less than three years.

(U) The ACM shall have the capability to detect the absence of external power and transition without power loss to use the ACM internal battery supply.

(U) When external power is present, the ACM shall use the external power in place of the internal battery.

(U) The ACM shall have the capability to report the status of the internal battery.

(U) The ACM shall provide internal battery with a life of at least TBD years.

1.

7.
8. Physical

(U) The ACM shall not exceed a width of TBD inches, a length of TBD inches and a height of TBD inches.

(U) The ACM shall have a maximum weight of TBD pounds.

1.
8.
9. Environmental

(U) The ACM shall operate at a temperature range of between TBDÂ°C to TBD Â°C at sea level.

(U) The ACM shall operate at altitudes up to TBD feet above sea level.

(U) The ACM shall operate when subject to random vibration IAW TBD.

(U) The ACM shall operate when subject to operational shock IAW TBD.

(U) The ACM shall meet the EMI/EMC requirements IAW TBD.

(U) The ACM shall operate when subjected to a relative humidity of TBD% to TBD%.

(U) The ACM shall survive exposure to temperature ranging from TBDÂ°C to TBDÂ°C while non-operational.

(U) The ACM shall survive exposure to altitudes up to TBD feet above sea level while non-operational.

(U) The ACM shall meet the requirements for thermal design IAW TBD.

(U) The ACM design shall not rely on cooling air coming into contact with internal electrical parts, circuitry, or connectors.

(U) The ACM shall not require or use internal electrical heaters, fans, blowers, or similar devices unless specifically approved by the Government.

1.
9.
10. Maintainability

(U) The ACM shall be designed such that preventive maintenance is not required other than battery replacement.

1.

10.
11. Reliability

(U) The ACM shall have a Mean Time between Failures (MTBF) of TBD hours.

1.

11.
12. Interchangeability

(U) The ACM as part of the TERMINAL INFOSEC module shall be fully interchangeable, both mechanically and electrically IAW TBD.

1.

12.
13. Workmanship

(U) The ACM workmanship shall be IAW TBD.

1.

13.
14. Documentation

(U) The ACM implementation shall document the software interface/primitives as part of the ACM Embedment Manual.

(U) The ACM primitives for a specific implementation shall be documented in the ACM Embedment Manual.

(U) The ACM implementation shall document the physical interfaces as part of the ACM Embedment Manual.

(U) The ACM implementation shall document the physical interface signaling including data rate, signal structure including duty cycle and shape, trigger [edge or signal level], levels and current, allowable jitter or skew, and timing relationships to other ACM interface signals as part of the ACM Embedment Manual.

1.

14.

15. IA Standards and Certification

(U) The ACM incorporates both NSA approved Type 1 and NIST approved Non Type 1 cryptographic algorithms. The ACM is required to have UIC Certification. The NIST approved Non Type 1 cryptographic algorithms are subject to FIPS-140-2 requirements. The FIPS-140-2 certification process is for use by commercial cryptographic modules.

(U) The ACM may be subject to both UIC and FIPS certification. The UIC Certification process involves Cryptographic Verification (CV) and Security Verification (SV) tests for the ACM. The UIC Certification provides a much higher assurance than the FIPS 140 certification. If the NIST approved algorithms are verified as part of the CV and SV tests for UIC Certification, the FIPS Certification can probably be avoided.

(U) The ACM shall be developed in accordance with (IAW) the IA Â“Defense in DepthÂ” standards by CJSCI 6510.01C, DoD Directive 8500.1 Information Assurance and DoD Directive 8500.2 Information Assurance Implementation.

(U//FOUO) The ACM shall meet the requirements of the Tailored System Security Requirements (SSR) and Fail Safe Design Analysis (FSDA) requirements from NSA.

(U//FOUO) The ACM security design shall be documented and submitted to NSA for approval IAW the Contract Data Requirements List (CDRL) identified in the Telecommunications Security Requirements Document (TSRD).

The ACM design shall meet the information assurance requirements described in the following sections and must satisfy the security and certifiability of the cryptographic application by the National Security Agency (NSA). The principal sections of Interim Information Assurance Directorate (IAD) Procedure No. 01-02 that must be addressed during the design include:

14.a(4) Proper Classification

14.b Development Requirements-Building Assurance Levels 1, 2, and 3

14.b(1)(a) Cleared Personnel

14.b(1)(b) Cleared Facilities

14.b(1)(c) Established Development Methodology and Standards

14.b(2) High Level System Requirements Review

14.b(8)(c)(1) Informal Review-Requirements

14.b(8)(c)(3) Informal Review-Code Review

The ACM shall be designed such that follow-on product delivery is capable of achieving Type 1 certification by NSA. Certification is a process involving meeting the security requirements provided by NSA as well as evolution of the technology and policy requirements as defined by the process documented in the User Partnership Agreement (UPA) between the Army and NSA. This process defines both the policy, technical capability, and testing needed for certification for a particular use.

The ACM will comply with the security and information assurance (IA) requirements provided in a document by the NSA as part of the User Partnership Agreement (UPA) for NSA cryptographic certification.

ACM security and IA requirements will be developed in accordance with tailored Unified INFOSEC Criteria (UIC), and documented in a Theory of Design and Operation (TDO).

The ACM shall include security mechanisms sufficient to satisfy the NSA requirements allocated to the ACM. The NSA will provide a document containing these requirements as part of the UPA for NSA cryptographic certification.

The ACM will demonstrate compliance with any additional security or IA requirements imposed outside of the TDO by ACM Type 1 certification with the NSA.

The ACM shall protect sensitive software and firmware within the ACM tamper boundary.

The ACM shall support Type-1 security classification up to TS/SCI for a single compartment.

The ACM shall include means of enforcing the least privilege principle. The least privilege principal requires that each subject be granted the most restrictive set of privileges needed for the performance of authorized tasks.

The ACM shall include services that allows the terminal to enforce that the system manager is authenticated before the terminal accepts its commands.

The ACM shall audit security relevant events. Security relevant events will be identified in the appropriate security document.

The ACM shall provide audit data to host upon request or when the memory is full.

The ACM shall perform cryptographic testing in order to become operational. The ACM will not transition to the operational state until the ACM passes all security-relevant tests.

The operational ACM shall provide the capability to automatically perform specified periodic cryptographic testing. The ACM must pass all security-relevant tests in order to remain operational.

(U) NOTES

1.
2. (U) Definitions

1.
    2.
    3. (U) Acronyms

AEHF Advanced Extremely High Frequency

AES Advanced Encryption Standard

API Application Program Interface

ASIC Application Specific Integrated Circuit

ATOW Acquisition Tracking Orderwire

BER Bit Error Ratio/Rate

BF Benign Fill

BIT Built in Test

BLOS Beyond Line Of Sight

CBC Cipher Block Chaining

CCI COMSEC Controlled Item

CDL Common Data Link

CDRL Contract Data Requirements List

CIK Crytpo Ignition Key

COI Communities Of Interest

COMSEC Communications Security

CORBA Common Object Request Broker Architecture

COTH Communication on the Halt

COTM Communications On The Move

COTQH Communications On The Quick Halt

COTS Commercial Off The Shelf

CRK Cryptographic Recovery Key

CSS Crypto Sub System

DoD Department of Defense

DoDAF Department of Defense Architecture Framework

DoS Denial of Service

EC-MQV Elliptic Curve Menezes-Qu-Vanstone

ECU End Cryptographic Unit

EFF Enhanced Firefly

EHF Extremely High Frequency

EKMS Electronic Key Management System

EMC Electromagnetic Compatibility

EMI Electromagnetic Interference

FCAPS Fault, Configuration, Account, Performance, Security

FCC Federal Communications Commission

FF Firefly

FPGA Field Programmable Gate Array

FSDA Fail Safe Design Assurance

GCM Galois Counter Mode

GBS Global Broadcast System

GIG Global Information Grid

GPP General Purpose Processor

GPS Global Positioning System

HAIPE High Assurance Internet Protocol Encryptor

HAIPE IS HAIPE Interoperability Specification

HC3 High Capacity Communications Capability

HCLOS High Capacity Line of Site

HDR High Data Rate

HF High Frequency

IA Information Assurance

I&A Identification and Authorization

IAW In Accordance With

IKE Internet Key Exchange

INFOSEC Information Security

IO Input Output

IP Internet Protocol

IPSec Internet Protocol Security

IPv4 Internet Protocol, version 4

IPv6 Internet Protocol, version 6

JTRS Joint Tactical Radio System

KB Kilo Byte

KEK Key Encryption Key

LDR Low Data Rate

LOS Line of Sight

MDR Medium Data Rate

MTBF Mean Time Between Failures

MTTR Mean Time To Repair

N-CDL Networked Common Data Link

NDL Network Data Link

NIST National Institute of Standards and Technology

NSA National Security Agency

OTAR Over The Air Rekey

OTAT Over The Air Transfer

OTAZ Over The Air Zeroize

OTNK Over The Network Keying

PKI Public Key Infrastructure

PPK Pre-Placed Key

RSA Rivest, Shamir, and Adelman

SA Security Association

SATCOM Satellite Communications

SBU Sensitive But Unclassified

SCA Software Communications Architecture

SNMP Simple Network Management Protocol

SW Software

SWAP Size, Weight and Power

TBD To Be Determined

TDN Tactical Data Network

TOD Time of Day

TRANSEC Transmission Security

TSAT Transformation Satellite Communications

TSRD Telecommunications Security Requirements Document

UAV Unmanned Aerial Vehicle

UIC Unified INFOSEC Criteria

VDC Voltage Direct Current

VHF Very High Frequency

WGS Wideband Gapfiller System

WIN-T Warfighter Information Network-Tactical

XCP eXplicit Control Protocol

XDR eXtended Data Rate

XML eXtensible Markup Language

A.

B. APPENDIX A: POET ACM REQUIRED ALGORITHMS

(U) POET ACM must provide the traffic algorithms that are required to support each of the terminal waveforms/applications. Some applications require more than one crypto algorithm to be running simultaneously. The algorithms are organized by application and the maximum data rate per algorithm per application is listed. The max data rates listed are draft and need to be confirmed. Each algorithm is also given an algorithm ID which is used to show simultaneity within and among applications. Required simultaneity among waveforms will be established by the terminal programs and is not shown.

(U) Type, S = Simplex; F = Full Duplex

(U) Highlighting shows areas that are TBD

(U) Note that no design or future proofing margins are shown

Where references are cited for the mode(s) that must be supported for a given algorithm, assume that all modes in the references must be supported unless otherwise indicated.

Table A-1 (U) AEHF Algorithms

Algorithm(s)	Function	Alg ID	Mode(s)	Max Rate	Type	Simultaneous with	Notes	Reference Document(s)
MEDLEY	Uplink TRANSEC stream generation	4a	Per references	1.024 Mbps	S	4b, 4c, 4d, 4e	[1]	
MEDLEY	Downlink TRANSEC stream generation	4b	Per references	1.024 Mbps	S	4a, 4c, 4d, 4e	[1]	
MEDLEY	XC2 encryption	4c	Per references	19.2 Kbps	S	4a, 4b, 4d, 4e	[2]	
MEDLEY	XC3 decryption	4d	Per references	38.4 Kbps	S	4a, 4b, 4d, 4e	[3]	
MEDLEY	XC3/ATOW decover stream generation	4e	Per references	96 Kbps	S	4a, 4b, 4c, 4d	[1], [4]	
SHILLELAGH	Uplink TRANSEC stream generation	4f	Per references	1.024 Mbps	S	4g, 4h, 4i, 4j	[1]	
SHILLELAGH	Downlink TRANSEC stream generation	4g	Per references	1.024 Mbps	S	4f, 4h, 4i, 4j	[1]	
SHILLELAGH	XC2 encryption	4h	Per references	19.2 Kbps	S	4f, 4g, 4i, 4j	[2]	
SHILLELAGH	XC3 decryption	4i	Per references	38.4 Kbps	S	4f, 4g, 4h, 4j	[3]	
SHILLELAGH	XC3/ATOW decover stream generation	4j	Per references	96 Kbps	S	4f, 4g, 4h, 4i	[1], [4]	

BATON	Uplink TRANSEC stream generation	4k	Per references	1.024 Mbps	S	4l, 4m, 4n, 4o	[1]	
BATON	Downlink TRANSEC stream generation	4l	Per references	1.024 Mbps	S	4k, 4m, 4n, 4o	[1]	
BATON	XC2 encryption	4m	Per references	19.2 Kbps	S	4k, 4l, 4n, 4o	[2]	
BATON	XC3 decryption	4n	Per references	38.4 Kbps	S	4k, 4l, 4m, 4o	[3]	
BATON	XC3/ATOW decover stream generation	4o	Per references	96 Kbps	S	4k, 4l, 4m, 4n	[1], [4]	
KEESEE	Uplink TRANSEC stream generation	4p	Per references	1.024 Mbps	S	4q	[1]	
KEESEE	Downlink TRANSEC stream generation	4q	Per references	1.024 Mbps	S	4p	[1]	

Notes

1.
2. Max rate is aggregate. Crypto must support outputting key stream block every TOD.
3. Max assumes one XC2 per frame plus sending OTADD.
4. Max assumes 3 XC3s per frame plus receiving OTADD.
5. Max assumes 10 XC3s or ATOWs per frame.

Table A-2 (U) MIL-188-165 Algorithms

Algorithm(s)	Function	Alg ID	Mode(s)	Max Rate	Type	Simultaneous with	Notes	Reference Document(s)
WALBURN	TRANSEC Bulk	6a	Per references	51.84 Mbps	F	6b	[1]	

	Encryption							
AES	IMPCS encryption / decryption	6b	ECB Mode	153.6 Kbps	F	6a	[2]	

Notes

1.
2. Max rate assumes that communication over WGS satellites in RF bypass mode is only used for CDL. The algorithms and maximum rate are therefore already covered by the CDL table.
3. Assume 256 bit key size.

Table A-3 (U) MIL-188-EEE Algorithms

Algorithm(s)	Function	Alg ID	Mode(s)	Max Rate	Type	Simultaneous with	Notes	Reference Document(s)
AES	Transmission Link Cover	7a	CBC Mode	24.576 Mbps	F		[1], [2]	

Notes

1.
2. Max rate is the maximum data rate for MIL-188-EEE link cover.
3. Assume 256 bit key size.

Table A-4 (U) CDL Algorithms

Algorithm(s)	Function	Alg ID	Mode(s)	Max Rate	Type	Simultaneous with	
CDL 1	Encrypt outbound	1a	Per references	10.71 Mbps (T) 548 Mbps (O)	S		[

CDL 1	Decrypt inbound	1a	Per references	274 Mbps (T) 548 Mbps (O)	S		
CDL 2	Encrypt outbound	1b	Per references	10.71 Mbps (T) 548 Mbps (O)	S		
CDL 2	Decrypt inbound	1b	Per references	274 Mbps (T) 548 Mbps (O)	S		

Notes

1.
2. Objective max rate is a full duplex 548 Mbps link.
3. See reference documents for required CDL algorithms.

Table A-5 (U) TSAT Algorithms

Algorithm(s)	Function	Alg ID	Mode(s)	Max Rate	Type	Simultaneous with	Notes	Reference Document(s)
MEDLEY	Uplink TRANSEC stream generation	5a	Per references	1.024 Mbps	S	[5b or 5q], 5c, 5d, 5e, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [2], [3]	
MEDLEY	Downlink TRANSEC stream generation	5b	Per references	1.024 Mbps	S	5a, 5c, 5d, 5e, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [2]	
MEDLEY	XC2 encryption	5c	Per references	19.2 Kbps	S	5a, [5b or 5q], 5d, 5e, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [4]	
MEDLEY	XC3 decryption	5d	Per references	38.4 Kbps	S	5a, [5b or 5q], 5c, 5e,	[1], [5]	

						[2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u		
MEDLEY	XC3/ATOW decover stream generation	5e	Per references	96 Kbps	S	5a, [5b or 5q], 5c, 5d, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [2], [6]	
SHILLELAGH	Uplink TRANSEC stream generation	5f	Per references	1.024 Mbps	S	[5g or 5q], 5h, 5i, 5j, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [2], [3]	
SHILLELAGH	Downlink TRANSEC stream generation	5g	Per references	1.024 Mbps	S	5f, 5h, 5i, 5j, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [2]	
SHILLELAGH	XC2 encryption	5h	Per references	19.2 Kbps	S	5f, [5g or 5q], 5i, 5j, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [4]	
SHILLELAGH	XC3 decryption	5i	Per references	38.4 Kbps	S	5f, [5g or 5q], 5h, 5j, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [5]	
SHILLELAGH	XC3/ATOW decover stream generation	5j	Per references	96 Kbps	S	5f, [5g or 5q], 5h, 5i, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [2], [6]	
BATON	Uplink TRANSEC stream generation	5k	Per references	1.024 Mbps	S	[5l or 5q], 5m, 5n, 5o, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [2], [3]	

BATON	Downlink TRANSEC stream generation	5l	Per references	1.024 Mbps	S	5k, 5m, 5n, 5o, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [2]	
BATON	XC2 encryption	5m	Per references	19.2 Kbps	S	5k, [5l or 5q], 5n, 5o, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [4]	
BATON	XC3 decryption	5n	Per references	38.4 Kbps	S	5k, [5l or 5q], 5m, 5o, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [5]	
BATON	XC3/ATOW decover stream generation	5o	Per references	96 Kbps	S	5k, [5l or 5q], 5m, 5n, [2a or 2b or 2c], 2d, 3a, 3b, 5p, 5r, 5s, 5t, 5u	[1], [2], [6]	
AES	Uplink TRANSEC stream generation	5p	TBD	5.12 Mbps	S	[5a or 5f or 5k], [5b or 5g or 5l or 5q], [5c/e/f or 5h/i/j or 5m/n/o], [2a or 2b or 2c], 2d, 3a, 3b, 5r, 5s, 5t, 5u	[2], [7], [8]	
AES	Downlink TRANSEC stream generation	5q	TBD	1.024 Mbps	S	[5a or 5f or 5k], 5p, [5c/e/f or 5h/i/j or 5m/n/o], [2a or 2b or 2c], 2d, 3a, 3b, 5r, 5s, 5t, 5u	[2], [8]	
AES	Permutation Table Decryption	5r	TBD	230.4 Kbps	S	[5a or 5f or 5k], [5b or 5g or 5l or 5q], 5p, [5c/e/f or	[8], [9]	

						5h/i/j or 5m/n/o], [2a or 2b or 2c], 2d, 3a, 3b, 5s, 5t, 5u		
AES	TCRC Message Authentication	5s	TBD	25.6 Kbps	S	[5a or 5f or 5k], [5b or 5g or 5l or 5q], 5p, [5c/e/f or 5h/i/j or 5m/n/o], [2a or 2b or 2c], 2d, 3a, 3b, 5t, 5u	[8], [10]	
AES	XDR+ Uplink Cover	5t	Counter Mode	147.456 Mbps	S	[5a or 5f or 5k], [5b or 5g or 5l or 5q], 5p, [5c/e/f or 5h/i/j or 5m/n/o], [2a or 2b or 2c], 2d, 3a, 3b, 5s, 5u	[8], [11]	
AES	XDR+ Downlink Cover	5u	Counter Mode	442.368 Mbps	S	[5a or 5f or 5k], [5b or 5g or 5l or 5q], 5p, [5c/e/f or 5h/i/j or 5m/n/o], [2a or 2b or 2c], 2d, 3a, 3b, 5s, 5t	[8], [12]	
AES (TBD)	Ka AISR Uplink Cover	5v	Counter Mode (TBD)	320 Mbps	S	5w	[8], [13], [14], [15], [16]	
AES (TBD)	SONET overhead encryption / decryption	5w	See notes	15.552 Mbps	F	5v	[8], [15], [17], [18], [19],	

							[20]	

Notes

1.
2. Needed for backward compatibility with AEHF.
3. Max rate is aggregate. Crypto must support outputting key stream block every TOD.
4. Simultaneity with 5p is worst case if TSAT goes with direct orthogonalization approach.
5. Max assumes one XC2 per frame plus sending OTADD.
6. Max assumes 3 XC3s per frame plus receiving OTADD.
7. Max assumes 10 XC3s or ATOWs per frame.
8. TSAT uplink TRANSEC is still being defined. The max rate assumes simultaneous generation of 5 different keys (worst case for direct orthogonalization approach).
9. Assume 256 bit key size.
10. Max rate assumes (25x9x1024) bit permutation table. Some margin is inherent since terminal should only get 1 column of table, not 25.
11. TSAT link layer security mechanisms are still being defined. Max rate assumes 1 TCRC message per frame with an encrypted, 512 bit authentication tag.
12. Max rate is the maximum expanded cover rate for the XDR+ uplink.
13. Max rate is the maximum expanded cover rate for the XDR+ downlink.
14. Any requirement for cover for TSAT Ka AISR links is TBD.
15. There are no implementation details for cover for TSAT Ka AISR links. The assumption is that Ka AISR cover is implemented similar to XDR+ cover (e.g., same algorithm, same mode, and use of a cover expansion function).
16. Simultaneity column assumes that a terminal is not running XDR/XDR+ and SONET (Ka AISR or lasercomm) concurrently.
17. Max rate is the maximum expanded cover rate for the Ka AISR links.
18. Max rate assumes encryption/decryption for SONET section, line, and path overhead for AISR links (Ka AISR or lasercomm). Any requirement for encryption/decryption of the SONET overhead is TBD.
19. Max rate is worst case. Note that path overhead is probably a separate E2E encryption.
20. SONET overhead is only full duplex for lasercomm. Ka AISR is simplex.
21. Must support all modes defined in NIST 800-38A (TBD).

Table A-6 (U) HAIPE Algorithms

Algorithm(s)	Function	Alg ID	Mode(s)	Max Rate	Type	Simultaneous with	Notes	Reference Document(s)
MEDLEY	Classified level terminal	2a	Per references	200 Kbps	F	2d	[1], [2]	

	management/control							
BATON	Classified level terminal management or control	2b	Per references	200 Kbps	F	2d	[1], [2]	
AES	Classified level terminal management or control	2c	See notes	200 Kbps	F	2d	[1], [2], [3], [5]	
WEASEL	Classified level terminal management or control	2d	Per references	25.6 Kbps	F	[2a or 2b or 2c]	[4]	

Notes

1.
2. Max rate assumes HAIPE used only for terminal net management and remote control. Baseband HAIPE rates are specified separately.
3. Max rate taken from Raytheon HC3 study security architecture report.
4. Assume 256 bit key size.
5. Max rate computed assuming 200 Kbps aggregate stream, median packet size of 500 bytes, and 512 bit hash length.
6. Must support all modes defined in NIST 800-38A (TBD).

Table A-7 (U) IPSec Algorithms

Algorithm(s)	Function	Alg ID	Mode(s)	Max Rate	Type	Simultaneous with	Notes	Reference Document(s)
AES	Router-to-router control plane (terminal to TSAT payload)	3a	See notes	500 Kbps	F		[1], [2], [5]	
AES	TMOS-to-terminal network management	3b	See notes	200 Kbps	F		[2], [3], [5]	
Any FIPS approved (TBD)	Router-to-router control plane (terminal to	3c	See notes	500 Kbps	F		[2], [4], [5]	

	baseband router)							
Any FIPS approved (TBD)	Other NMS-to-terminal network management	3d	See notes	200 Kbps	F		[2], [4], [5]	

Notes

1.
2. Max rate estimated based on router convergence after TSAT logon. Assumes 150 route updates, max 4096 bytes per update, and 10s for convergence. All are conservative to estimate worst case data rate.
3. Assume 256 bit key size.
4. Max rate taken from Raytheon HC3 study security architecture report.
5. Uses same max rate as estimated in [1] because it is a similar application.
6. Must support all modes defined in NIST 800-38A (TBD).

(U) The POET ACM must support the following algorithms for key management. The key management algorithms are assumed to be simultaneous with any waveform, but do not have a maximum data rate.

Table A-8 (U) Key Management Algorithms

Algorithm	Function	Alg ID	Performance	Notes	Reference Document(s)
FIREFLY v 9	Benign key fill	20a	TBD	[1], [2]	
FIREFLY v 17	ECU-to-ECU key agreement	20b	TBD	[5]	
Enhanced FIREFLY	Benign key fill, benign algorithm fill (TBD)	20c	TBD	[1], [2], [3]	
MAYFLY	Benign key fill, benign algorithm fill (TBD)	20d	TBD	[3]	
ACCORDION v 1.3	BLACK fill, File encryption/decryption (TBD)	20e	TBD	[1], [4]	
ACCORDION v 3.0	BLACK fill, File encryption/decryption (TBD)	20f	TBD	[3]	
FFC DH	Key agreement	20g	TBD	[6]	
ECC CDH	Key agreement	20h	TBD	[6]	
FFC MQV	Key agreement	20i	TBD	[6], [7]	
ECC MQV	Key agreement	20j	TBD	[6], [7]	

AES Key Wrap	BLACK fill, File encryption/decryption (TBD)	20k	TBD	[7], [8]	

Notes

1.
2. Required for EKMS compatibility.
3. Required for HAIPE interoperability.
4. Required for KMI compatibility.
5. Required for AEHF key management.
6. Assume that this is required for ECU-to-ECU FIREFLY key agreement (TBD).
7. Required for IPSec interoperability.
8. Required for foreign releasable KMI compatibility.
9. Assume 256 bit key size.

(U) The POET ACM must support the following algorithms for general use. The key management algorithms are assumed to be simultaneous with any waveform, but do not have a maximum data rate.

Table A-9 (U) Terminal Utility Algorithms

Algorithm(s)	Function	Alg ID	Mode(s)	Max Rate	Type	Simultaneous with	Notes	Reference Document(s)
AES Key Wrap	Terminal File Encryption/Decryption	10a	N/A	200 Mbs	S		[1], [2]	
ACCORDION	Terminal File Encryption/Decryption	10b	N/A	200 Mbs	S		[1], [2]	

Notes

1.
2. Max rate assumes 100 Mb file encrypted or decrypted in 0.5s (TBD).
3. Assume 256 bit key size.

Table A-10 (U) Baseband COMSEC

Algorithm(s)	Function	Alg	Mode(s)	Max	Type	Simultaneous	Notes	Reference

		ID		Rate		with		Document(s)
Per references	Baseband HAIPE	11a	Per references					

(U) The POET ACM must support the following asymmetric key encryption algorithms.

Table A-11 (U) Asymmetric Encryption Algorithms

Algorithm	Function	Alg ID	Performance	Notes	Reference Document(s)
RSA	Asymmetric encryption/decryption	40a	TBD	[2], [3]	
Type 1 Asymmetric (TBD)	Benign key fill, benign data fill (TBD)	40b	TBD	[1], [4]	

Notes

1.
2. Required for KMI compatibility (TBD). Assume that a Type 1 asymmetric algorithm will be available in the KMI CI-3 timeframe.
3. Required for IPSec interoperability.
4. Required for DoD PKI compatibility.
5. A schedule for the LANTERN algorithm is not yet available.

(U) The POET ACM must support the following algorithms.

Table A-12 (U) Other Algorithms

Algorithm	Function	Alg ID	Performance	Notes	Reference Document(s)

JOSEKI	Crypto firmware and/or algorithm decryption	50a	20 Mbs	[1]	

Notes

1.
2. Performance is tied to terminal startup timeline. Performance is derived as decrypting a 20 Mb file in 0.5 s.

B.

C. APPENDIX B: HAIPE IS REQUIREMENT ALLOCATION

(U) The following table presents the allocation of the HAIPE IS 3.0 requirements to the POET ACM, portions of the platform other than the ACM, or to both. Notes are provided as necessary to provide additional detail or justification. This table is provided for reference purposed only.

Table B-1 (U) HAIPE IS REQUIREMENT ALLOCATION

Requirement ID	Requirement Type	Requirement Text	Requirement Source	Platform Allocation	Notes
TP.1	Threshold	HAIPEs shall pre-configure the deviceVersionTable as follows: dvName is "TrafficProtection", and dvVersion is "3.0.0".	Traffic Protection Core	Terminal	
TP.ESP.1	Threshold	HAIPEs shall set the Source Address field in the CT IP header of all outbound packets with the value specified in the corresponding Security Association Database entry.	Traffic Protection Core	Terminal	
TP.ESP.2	Threshold	HAIPEs shall set the Destination Address field in the CT IP header of all outbound packets with the value specified in the corresponding Security Association Database entry.	Traffic Protection Core	Terminal	
TP.ESP.3	Threshold	HAIPEs shall set the DSCP bits in the CT IP header of outbound packets to an all ZERO value by default on a per SA basis.	Traffic Protection Core	Terminal	
TP.ESP.4	Threshold	HAIPEs shall support the capability to define a set of administrator-specified DSCP values that if present are	Traffic Protection Core	Terminal	

		copied from PT IP headers to the CT IP headers of outbound packets on a per SA basis.			
TP.ESP.5	Threshold	HAIPEs shall support the copying of administrator-specified DSCP values from the PT IP headers to the CT IP headers of outbound packets on a per SA basis.	Traffic Protection Core	Terminal	
TP.ESP.6	Threshold	HAIPEs shall set the CT ECN bits to 10 for all packets, except when in admission control processing mode.	Traffic Protection Core	Terminal	
TP.ESP.7	Threshold	HAIPEs shall have the capability of enabling and disabling an ECN processing mode for congestion notification on a per SA basis.	Traffic Protection Core	Terminal	
TP.ESP.8	Threshold	On a SA where ECN processing is enabled for congestion notification, HAIPEs shall set the ECN bits in the post-decapsulation PT IP header as specified in Table - TP.ESP.8.	Traffic Protection Core	Terminal	
TP.ESP.9	Threshold	HAIPE ECN Congestion Notification processing shall default to OFF.	Traffic Protection Core	Terminal	
TP.ESP.10	Threshold	HAIPEs shall have the capability of ENABLING and DISABLING an ECN processing mode for admission control.	Traffic Protection Core	Terminal	
TP.ESP.11	Threshold	On a SA where ECN processing is enabled for admission control, HAIPEs shall support the copying of ECN values from the PT IP headers to the CT IP headers of outbound packets.	Traffic Protection Core	Terminal	
TP.ESP.12	Threshold	When ECN processing is enabled for admission control, HAIPEs shall support the copying of ECN values from the CT IP headers to the PT IP headers of inbound packets on a per SA basis.	Traffic Protection Core	Terminal	
TP.ESP.13	Threshold	HAIPE ECN Admission Control processing shall default to OFF.	Traffic Protection Core	Terminal	
TP.ESP.14	Threshold	HAIPE shall set the CT header Flow Label on outbound packets to all ZERO's by default on a per SA basis.	Traffic Protection Core	Terminal	
TP.ESP.15	Objective	HAIPEs shall support configuration of the default Flow Label value on outgoing CT packets on a per SA basis.	Traffic Protection Core	Terminal	
TP.ESP.16	Threshold	HAIPE shall support enabling/disabling IPv6 Flow Label processing on a per SA basis.	Traffic Protection Core	Terminal	
TP.ESP.17	Threshold	HAIPEs shall support configuration of the Flow Label masking value on a per SA basis.	Traffic Protection Core	Terminal	
TP.ESP.18	Threshold	When Flow Label processing is enabled, HAIPEs shall perform a bitwise AND	Traffic Protection Core	Terminal	

		function of the Flow Label masking value with the incoming PT packet Flow Label value and insert the result into the CT Header Flow Label.			
TP.ESP.19	Threshold	HAIPE shall set the TTL field in the CT IP Header for outbound packets on a per-interface basis to a constant value.	Traffic Protection Core	Terminal	
TP.ESP.20	Objective	HAIPEs shall make the TTL value in the CT IP Header for outbound packets configurable by the operator on a per-interface basis.	Traffic Protection Core	Terminal	
TP.ESP.21	Threshold	HAIPE shall set the IPv6 Hop Limit in the CT IP Header for outbound packets to a constant value on a per-interface basis.	Traffic Protection Core	Terminal	
TP.ESP.22	Objective	HAIPEs shall make the CT IPv6 Hop Limit constant value operator configurable on a per-interface basis.	Traffic Protection Core	Terminal	
TP.ESP.23	Threshold	HAIPEs shall be capable of configuring the system's treatment of the DF bit (set, clear, copy from encapsulated header) from the PT IP header to the CT IP header on a per SA basis.	Traffic Protection Core	Terminal	
TP.ESP.24	Threshold	The receiving HAIPE shall ignore the contents of the padding octets.	Traffic Protection Core	Terminal	
TP.ESP.25	Threshold	HAIPEs shall initialize the sliding window counter to zero, dropping any packet with a Sequence Number of ZERO.	Traffic Protection Core	Terminal	
TP.ESP.26	Threshold	HAIPEs shall verify that a received packet contains a sequence number that is not a duplicate for the life of the SA.	Traffic Protection Core	Terminal	
TP.ESP.27	Threshold	HAIPEs shall allow for the CT fixed packet length option.	Traffic Protection Core	Terminal	
TP.ESP.28	Threshold	HAIPEs shall allow for the CT fixed packet length option to be configurable.	Traffic Protection Core	Terminal	
TP.ESP.29	Threshold	HAIPEs shall have the CT fixed packet length option default to OFF.	Traffic Protection Core	Terminal	
TP.ESP.30	Threshold	HAIPEs shall pad packets arriving on the PT interface to create packets, that when combined with cryptographic and ESP overhead, equal the configured CT Fixed Packet Length.	Traffic Protection Core	Terminal	
TP.ESP.31	Objective	If the Fixed Packet Length Option is enabled and the DF bit is cleared, HAIPEs shall fragment PT packets that after ESP processing would create CT packets exceeding the fixed CT packet length configured, with the last fragment being padded as necessary.	Traffic Protection Core	Terminal	

TP.ESP.32	Objective	If the Fixed Packet Length Option is enabled and the DF bit is set, HAIPEs shall discard PT packets that, after ESP processing, would result in CT packets greater than the CT Fixed packet length and generate an ICMP destination unreachable (Code 4) message.	Traffic Protection Core	Terminal	
TP.ESP.33	Objective	HAIPEs shall support fragmentation of PT IPv4 packets to produce fixed length CT packets.	Traffic Protection Core	Terminal	
TP.ESP.34	Objective	HAIPEs shall be able to encapsulate and encrypt a packet concatenated with a variable number of TFC pad octets. ("Variable" means zero to the maximum number such that the resulting ESP packet does not exceed the link MTU).	Traffic Protection Core	Terminal POET ACM	Assuming that the to be encrypted datagram is a multiple of the encrypt/decrypt block size.
TP.ESP.35	Objective	HAIPEs shall allow a user to enable a mode of operation where the HAIPE encapsulates and encrypts user packets concatenated with a variable number of TFC pad octets.	Traffic Protection Core	Terminal POET ACM	Assuming that the to be encrypted datagram is a multiple of the encrypt/decrypt block size.
TP.ESP.36	Threshold	HAIPEs shall have the CT variable length masking option default to OFF.	Traffic Protection Core	Terminal	
TP.ESP.37	Threshold	HAIPEs shall be able to decrypt and properly decapsulate a packet that contains a user packet concatenated with an a priori unknown -- variable -- number of TFC pad octets.	Traffic Protection Core	Terminal POET ACM	Assuming that the to be decrypted datagram is a multiple of the encrypt/decrypt block size.
TP.ESP.38	Objective	HAIPEs shall discard received PT packets that are destined to be tunneled if the PT packet's IPv6 header has a Hop Limit value (prior to decrementing) of ZERO.	Traffic Protection Core	Terminal	
TP.ESP.39	Objective	HAIPEs shall discard received PT packets that are destined to be tunneled if the PT packet's IPv4 header has a TTL value (prior to decrementing) of ZERO.	Traffic Protection Core	Terminal	
TP.ESP.ESPV3.1	Threshold	HAIPE shall not use SPI values 0-255 for manual SAs, or during a Key Agreement Exchange for an automated SA.	Traffic Protection Core	Terminal	
TP.ESP.ESPV3.2	Threshold	HAIPEs shall reset the Sequence Number such that when the SPI changes due to a PPK/Update or Active/Changeover, the first packet transmitted under the new SPI contains a Sequence Number of one.	Traffic Protection Core	Terminal	
TP.ESP.ESPV3.3	Threshold	HAIPEs shall set the value 59 in the Next header field for dummy packets.	Traffic Protection Core	Terminal	
TP.ESP.ESPV3.4	Threshold	HAIPEs shall discard all packets received with the	Traffic Protection Core	Terminal	

		value of 59 specified in the Next Header field without indicating an error.			
TP.ESP.ESPV3.TNDMD.1	Threshold	HAIPEs shall format ESPv3 packets as specified in Section 3.1 of RFC 2406 for Tunnel Mode.	Traffic Protection Core	Terminal	
TP.ESP.ESPV3.TNDMD.2	Threshold	When configured for Tunnel Mode, HAIPEs shall format the ESPv3 packet as specified in Table - TP.ESP.ESPV3.TNDMD.2.	Traffic Protection Core	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.ESP.ESPV3.TNDMD.3	Threshold	HAIPEs shall encrypt all information from the Original IP Packet to the Next Header fields inclusive.	Traffic Protection Core	Terminal POET ACM	The network interfaces are responsible for preparing the data in such a way that the appropriate data is encrypted correctly.
TP.ESP.ESPV3.TNDMD.4	Threshold	HAIPEs shall add padding (Traffic Flow Confidentiality, then Cryptographic) starting from the least significant bit (LSb) of the PT packet.	Traffic Protection Core	Terminal	
TP.IKE.LEG.1	Threshold	HAIPEs shall populate PT IP Header source and destination addresses in IKE Message 5 with the HAIPE PT interface IP addresses or any internal PT IP Addresses.	Traffic Protection Core	Terminal	
TP.IKE.LEG.2	Threshold	HAIPEs shall send a Delete Payload containing the SPI(s) to be deleted to the peer HAIPE on the receipt of a CT packet with an unknown SPI after re-establishing an SA with that peer HAIPE using HAIPE's orphan SA recovery.	Traffic Protection Core	Terminal	
TP.IKE.LEG.3	Threshold	When generating an IKE exchange in response to receipt of an ESP packet with an unknown SPI, HAIPEs shall populate the IKE Message 5 PT IPv4 or IPv6 destination address, if unknown, with all ZEROs.	Traffic Protection Core	Terminal	
TP.IKE.LEG.4	Threshold	Upon receipt of an IKE message 5 with a PT destination IP address of all ZERO's, HAIPEs shall respond with an IKE Message 6.	Traffic Protection Core	Terminal	
TP.IKE.LEG.5	Threshold	HAIPEs shall format the Delete Payload as detailed in RFC 2408, Section 3.15.	Traffic Protection Core	Terminal	
TP.IKE.LEG.6	Threshold	HAIPEs shall perform Delete Payload processing as detailed in RFC 2408, Section 5.15 and as modified herein.	Traffic Protection Core	Terminal POET ACM	The requirements in this document are satisfied by both the POET ACM and the remaining components of the terminal.

TP.IKE.LEG.7	Threshold	HAIPEs shall populate the Delete Payload Security Parameter Index(es) (SPI) field with the unknown SPI initiating orphan SA recovery and the SPI assigned in IKE Message 1 of the IKE Exchange in progress.	Traffic Protection Core	Terminal	
TP.IKE.LEG.8	Threshold	HAIPEs initiating an orphan SA recovery shall tear down the SA established to send a Delete Payload after receipt of IKE Message 6.	Traffic Protection Core	Terminal POET ACM	SA termination requires key deletion.
TP.IKE.LEG.9	Threshold	If the security levels exchanged in the ID Payload do not match and/or are outside of the overlapping range of initiator and responder classification inherent in the negotiated vector set, the HAIPE shall delete the TEK resulting from a Key Agreement Exchange and abort the IKE exchange.	Traffic Protection Core	Terminal POET ACM	
TP.IKE.LEG.10	Threshold	HAIPEs shall negotiate the Encryption Algorithm in IKE Messages 1 and 2 independent of setting the Proposal Payload Transform ID.	Traffic Protection Core	Terminal POET ACM	The negotiated algorithm is used during the generation of the pairwise TEK.
TP.IKE.LEG.11	Threshold	HAIPEs shall not send Lifetime Type and duration Security Association Attribute Types.	Traffic Protection Core	Terminal	
TP.IKE.LEG.12	Threshold	HAIPEs offering more than one Universal/Edition shall do so in the form of separate Transform Payloads.	Traffic Protection Core	Terminal	
TP.IKE.LEG.13	Threshold	HAIPEs shall allow for the exchange of a minimum of 6 Universals, and a minimum of 2 Editions per Universal (i.e., Current and Next) during IKE Messages 1-2.	Traffic Protection Core	Terminal	
TP.IKE.LEG.14	Threshold	HAIPEs shall pass the Most Significant (MS) credential (CC1) in the KE payload, followed immediately by the CC2 credential.	Traffic Protection Core	Terminal POET ACM	
TP.IKE.LEG.15	Threshold	HAIPEs negotiating ESPv3 and EKL shall pass the Most Significant (MS) credential (CC1) concatenated with the CC2 credential, concatenated with the EKL of the negotiated vector set.	Traffic Protection Core	Terminal POET ACM	
TP.IKE.LEG.16	Threshold	HAIPEs shall format the Exclusion Key List Tag as shown in Table TP.IKE.LEG.16.	Traffic Protection Core	Terminal	
TP.IKE.LEG.17	Threshold	HAIPEs shall code the Exclusion Key List Tag as shown in Table TP.IKE.LEG.17.	Traffic Protection Core	Terminal	
TP.IKE.LEG.18	Threshold	HAIPEs shall pass the EK Tag information using the following EKL format. [Key Type-Global][Number of Tag Parameters][Global Global Tag][Key Type-Local][Number of Tag Parameters][Local Tag.	Traffic Protection Core	Terminal	

		Local Tag][Key Type-Privacy][Number of Tag Parameters][Privacy Tag...Privacy Tag][Key Type-Separation][Number of Tag Parameters][Separation Tag Separation Tag].			
TP.IKE.LEG.19	Threshold	HAIPEs shall support receipt of Transform Payloads (detailed as Suites) detailed in Table - TP.IKE.LEG.19 in IKE Message 1.	Traffic Protection Core	Terminal POET ACM	Table defines required encryption algorithm, integrity mode, hash algorithm, update method, authentication method, and crypto block size.
TP.IKE.LEG.20	Threshold	HAIPEs shall code EKL Tag Type 00000 (DS 100) as shown in Table - TP.IKE.LEG.20.	Traffic Protection Core	Terminal POET ACM	Short title information MAY have to be provided by the POET ACM.
TP.IKE.LEG.21	Threshold	HAIPEs shall offer a Transform Payload including the EKL Class "supported" for Global, Local, Privacy or Separation EKs and another lower prioritized Transform Payload with identical SA Data Attributes excluding the EKL Class for Privacy and Separation EK, for each Universal ID offered in a Key Agreement Exchange with an associated EKL as shown in Table TP.IKE.LEG.21.	Traffic Protection Core	Terminal	
TP.IKE.LEG.22	Threshold	HAIPEs shall offer "Current" EKL for Key Agreement Exchanges initiated before Active/Changeover.	Traffic Protection Core	Terminal	
TP.IKE.LEG.23	Threshold	HAIPEs shall offer "Next" EKL for Key Agreement Exchanges initiated after Active/Changeover.	Traffic Protection Core	Terminal	
TP.IKE.LEG.24	Threshold	HAIPEs shall support receipt of "Next" EKL at 2305 GMT at Active/Changeover.	Traffic Protection Core	Terminal	
TP.IKE.LEG.25	Threshold	HAIPEs shall support receipt of "Current" EKL until 0055 GMT at Active/Changeover.	Traffic Protection Core	Terminal	
TP.IKE.LEG.26	Threshold	HAIPEs shall perform EKL processing as detailed in "Exclusion Key and its Application to Foreign Interoperability" dated June 7, 2002.	Traffic Protection Core	Terminal POET ACM	
TP.IKE.LEG.27	Threshold	HAIPEs shall populate the ID Payload Data field (MSb to LSb) with the Classification Attribute Type/ Attribute Value, followed by the CKL Attribute Type/Attribute Value based on the negotiated vector set in IKE Messages 1-2, followed with the Connection Type Attribute Type (if supported)/Attribute Value.	Traffic Protection Core	Terminal	

TP.IKE.LEG.28	Threshold	HAIPEs shall code the Universal ID/Edition Attribute Type, Attribute Value Field (MSb-LSb) Universal ID (4 bytes, coded as 0000-9999 ASCII), Universal Edition (2 bytes, coded as 01-99 ASCII).	Traffic Protection Core	Terminal	
TP.IKE.LEG.29	Threshold	HAIPEs shall send all IKE Messages addressed to UDP Destination port 500 (decimal).	Traffic Protection Core	Terminal	
TP.IKE.LEG.30	Threshold	HAIPEs shall set the HAIPE IPv4 IHL field to 5 decimal.	Traffic Protection Core	Terminal	
TP.IKE.LEG.31	Threshold	HAIPEs shall include in IKE Message 1 UDP Payload, ISAKMP payloads as shown in Table - TP.IKE.LEG.31.	Traffic Protection Core	Terminal	
TP.IKE.LEG.32	Threshold	On receipt of an unfamiliar vendor ID, HAIPEs shall process the Vendor ID Payload, Next Payload, Reserved and Payload Length fields to determine the beginning of the next payload to process.	Traffic Protection Core	Terminal	
TP.IKE.LEG.33	Threshold	HAIPEs shall include in IKE Message 2 UDP Payload, ISAKMP payloads as shown in Table TP.IKE.LEG.33.	Traffic Protection Core	Terminal	
TP.IKE.LEG.34	Threshold	HAIPEs shall include in IKE Message 3 UDP Payload, ISAKMP payloads as shown in Table - TP.IKE.LEG.34.	Traffic Protection Core	Terminal POET ACM	Credentials are provided by the POET ACM.
TP.IKE.LEG.35	Threshold	HAIPEs shall include in IKE Message 4 UDP Payload, ISAKMP payloads as shown in Table TP.IKE.LEG.35.	Traffic Protection Core	Terminal POET ACM	Credentials are provided by the POET ACM.
TP.IKE.LEG.36	Threshold	HAIPEs shall protect IKE Messages 5 and 6 in the ESP Mode negotiated in IKE Messages 1-4.	Traffic Protection Core	Terminal POET ACM	
TP.IKE.LEG.37	Threshold	When GCM is negotiated as the Integrity Algorithm, HAIPEs shall default to ESPv3 Tunnel Mode in the ABSENCE of an Encapsulation Mode attribute in the Transform Payload.	Traffic Protection Core	Terminal	
TP.IKE.LEG.38	Threshold	HAIPEs shall include in IKE Message 5 UDP Payload, ISAKMP payloads as shown Table - TP.IKE.LEG.38.	Traffic Protection Core	Terminal	
TP.IKE.LEG.39	Threshold	HAIPEs shall set the Sequence Number to one (1) for IKE message 5 and 6, and then increment it for each packet sent on the SA. Note: Retransmitted IKE Messages will increment the Sequence Number.	Traffic Protection Core	Terminal	
TP.IKE.LEG.40	Threshold	HAIPEs shall not reject IKE Message 5 or 6 after performing anti-replay if the Sequence Number is greater than 1.	Traffic Protection Core	Terminal	
TP.IKE.LEG.41	Threshold	HAIPEs shall populate the HASH_I Payload with the negotiated Hash Algorithms output of the concatenation of A^ri Mod P, A^rr Mod P, CKY-	Traffic Protection Core	Terminal	

		I, CKY-R, SAi_b and Idii_b.			
TP.IKE.LEG.42	Threshold	HAIPEs shall validate the HASH_I Payload, Hash Data field and terminate the IKE exchange if invalid.	Traffic Protection Core	Terminal	
TP.IKE.LEG.43	Threshold	HAIPEs shall include in IKE Message 6 UDP Payload, ISAKMP payloads as shown Table - TP.IKE.LEG.43.	Traffic Protection Core	Terminal	
TP.IKE.LEG.44	Threshold	HAIPEs shall populate the HASH_R Payload with the negotiated Hash Algorithms output of the concatenation of A^rr Mod P, A^ri Mod P, CKY-R, CKY-I, SAi_b and IDir_b.	Traffic Protection Core	Terminal	
TP.IKE.LEG.45	Threshold	HAIPEs shall validate the HASH_R Payload, Hash Data field, and terminate the IKE exchange if invalid.	Traffic Protection Core	Terminal	
TP.IKE.LEG.46	Threshold	HAIPEs shall code and interpret the Security Association Data Attributes as specified in Section 3.3 of RFC 2408, and as detailed in Table - TP.IKE.LEG.46.	Traffic Protection Core	Terminal	
TP.PE.1	Threshold	HAIPEs shall support a mechanism for an operator to manually delete SAs by modifying the Security Association Database.	Traffic Protection Core	Terminal POET ACM	SA termination requires key deletion.
TP.PE.SAD.AUTO.1	Threshold	Upon successful completion of the Key Agreement Exchange, HAIPEs shall use the new TEK to protect traffic.	Traffic Protection Core	Terminal POET ACM	
TP.PE.SAD.AUTO.2	Threshold	HAIPEs shall support the assignment of a loaded asymmetric Key to be the Next key for a specific Current asymmetric key.	Traffic Protection Core	Terminal POET ACM	
TP.PE.SAD.AUTO.3	Threshold	Upon receipt of a Delete Payload, HAIPEs shall release the associated SAs if the identified SPIs are valid for the source HAIPE sending the Delete Payload. Note: Validity involves checking the SAD for an existing SA with the SPI, CT and PT HAIPE IP addresses of the HAIPE sending the Delete Payload.	Traffic Protection Core	Terminal POET ACM	SA termination requires key deletion.
TP.PE.SAD.MAN.1	Threshold	When performing Active/Changeover, HAIPEs shall use the "next" TEK starting at 0000 GMT to protect transmit traffic.	Traffic Protection Core	Terminal POET ACM	
TP.PE.SAD.MAN.2	Threshold	When performing Active/Changeover, HAIPEs shall accept received traffic encrypted with the "next" TEK starting at 2305 GMT on the last day of the previous calendar month.	Traffic Protection Core	Terminal POET ACM	
TP.PE.SAD.MAN.3	Threshold	When performing Active/Changeover, HAIPEs shall continue to accept received traffic encrypted with the "current" TEK until 0055 GMT on the first day of the	Traffic Protection Core	Terminal POET ACM	

		next calendar month.			
TP.PE.SAD.MAN.4	Threshold	When performing Active/Changeover, at 0055 GMT on the first day of the next calendar month, HAIPEs shall delete the "current" TEK.	Traffic Protection Core	Terminal POET ACM	
TP.PE.SAD.MAN.5	Threshold	When performing a Deterministic Update on a TEK, HAIPEs configured to use ESPv3 shall use the "next" TEK starting at 0000 GMT to protect transmit traffic.	Traffic Protection Core	Terminal POET ACM	
TP.PE.SAD.MAN.6	Threshold	When performing a Deterministic Update on a TEK, HAIPEs configured to use ESPv3 shall accept received traffic encrypted with the "next" TEK starting at 2305 GMT.	Traffic Protection Core	Terminal POET ACM	
TP.PE.SAD.MAN.7	Threshold	When performing a Deterministic Update on a TEK, HAIPEs configured to use ESPv3 shall continue to accept received traffic encrypted with the "current" TEK until 0055 GMT.	Traffic Protection Core	Terminal POET ACM	
TP.PE.SAD.MAN.8	Threshold	When performing a Deterministic Update on a TEK, HAIPEs configured to use ESPv3, at 0055 GMT, shall delete the "current" TEK.	Traffic Protection Core	Terminal POET ACM	
TP.PE.SAD.MAN.9	Threshold	HAIPEs shall support mapping inbound traffic to a SA using the longest match based on the 3-tuple (SPI, Destination IP Address, Source IP Address).	Traffic Protection Core	Terminal	
TP.PE.SAD.MAN.10	Threshold	HAIPEs shall not perform Deterministic Updates on Exclusion Keys.	Traffic Protection Core	Terminal	
TP.PE.SAD.MAN.11	Threshold	HAIPEs shall support creating SAs to encrypt IP user multicast traffic.	Traffic Protection Core	Terminal POET ACM	
TP.PE.SAD.MAN.12	Threshold	HAIPEs shall support configuration of current and next PPK SPIs for all SAs configured to use ESPv3.	Traffic Protection Core	Terminal	
TP.PE.SAD.MAN.13	Threshold	HAIPEs shall reject configuration of consecutive PPK SPIs that are the same, for all SAs configured to use ESPv3.	Traffic Protection Core	Terminal	
TP.PE.SAD.MAN.14	Threshold	HAIPEs shall reject configuration of identical values for an incoming/outgoing PPK SPI pair for the same unicast SA, for all SAs configured to use ESPv3.	Traffic Protection Core	Terminal	
TP.PE.SAD.MAN.15	Threshold	HAIPEs shall reject configuration of a SPI value that is already in use by another SA.	Traffic Protection Core	Terminal	
TP.PE.SAD.MAN.16	Threshold	HAIPEs shall support the assignment of a loaded symmetric key to be the Changeover key for a specific Active symmetric key.	Traffic Protection Core	Terminal POET ACM	

TP.PE.SAD.MAN.17	Objective	HAIPEs shall support the assignment of a sequence of eleven loaded symmetric keys to be the Changeover keys for a specific Active symmetric key.	Traffic Protection Core	Terminal POET ACM	
TP.PE.SAD.MAN.18	Threshold	HAIPEs shall provide the capability to configure the ESPv3 mode for Integrity Check Value of 96 bits.	Traffic Protection Core	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.PE.SAD.MAN.19	Threshold	HAIPEs shall provide the capability to configure the ESPv3 mode for Integrity Check Value of 0 bits.	Traffic Protection Core	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.PE.SAD.MAN.20	Objective	HAIPEs shall provide the capability to configure the ESPv3 mode for Integrity Check Value of 32 bits.	Traffic Protection Core	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.PE.SAD.MAN.21	Objective	HAIPEs shall provide the capability to configure the ESPv3 mode for Integrity Check Value of 64 bits.	Traffic Protection Core	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.PE.SAD.MAN.22	Objective	HAIPEs shall provide the capability to configure the ESPv3 mode for Integrity Check Value of 128 bits.	Traffic Protection Core	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.PE.SPD.1	Threshold	HAIPEs shall allow for the management of the Security Policy Database as specified in RFC 2401 and consistent with the SPD Selectors required herein.	Traffic Protection Core	Terminal	
TP.PE.SPD.2	Threshold	HAIPEs shall order the Security Policy Database so that lookup results are deterministic.	Traffic Protection Core	Terminal	
TP.PE.SPD.3	Threshold	HAIPEs shall discard packets which do not match a policy in the Security Policy Database.	Traffic Protection Core	Terminal	
TP.PE.SPD.4	Threshold	HAIPEs shall allow for the use of a Destination IPv4 Address as a Selector in the Security Policy Database.	Traffic Protection Core	Terminal	
TP.PE.SPD.5	Threshold	HAIPEs shall allow for the use of a Destination IPv6 Address as a Selector in the Security Policy Database.	Traffic Protection Core	Terminal	
TP.PE.SPD.6	Threshold	HAIPEs shall allow for the use of a Source IPv4 Address as a Selector in the Security Policy Database.	Traffic Protection Core	Terminal	
TP.PE.SPD.7	Threshold	HAIPEs shall allow for the use of a Source IPv6 Address as a Selector in the Security Policy Database.	Traffic Protection Core	Terminal	
TP.PE.SPD.8	Threshold	HAIPEs shall allow for the use of the ANY value with Security Policy Database Selectors.	Traffic Protection Core	Terminal	
TP.PE.SPD.9	Threshold	When a Security Policy Database entry is manually deleted, HAIPEs shall terminate all associated SAs.	Traffic Protection Core	Terminal POET ACM	SA termination requires key deletion.
TP.PE.SPD.10	Threshold	HAIPEs shall discard any IP packet received on the PT	Traffic Protection Core	Terminal	

		interface with a source address of the HAIPE PT IP interface or any internal PT IP address.			
TP.PE.SPD.11	Threshold	HAIPEs shall discard any IP packet received on the CT interface with a source address (pre-decryption) of its own CT interface.	Traffic Protection Core	Terminal	
TP.PE.SPD.12	Threshold	HAIPEs shall discard any IP packet received on the CT interface with an inner header (post-decryption) source address of its own PT IP interface or any internal PT IP address.	Traffic Protection Core	Terminal	
NET.IP.1	Threshold	HAIPEs shall generate the UDP checksum and populate it in the checksum field for all UDP packets it transmits.	Networking Core	Terminal	
NET.IP.2	Threshold	HAIPEs shall support simultaneously running IPv4 and IPv6 on both the Plaintext and Ciphertext traffic interfaces.	Networking Core	Terminal	
NET.IP.3	Threshold	HAIPEs shall pre-configure the deviceVersionTable as follows: dvName is "Networking", and dvVersion is "3.0.0".	Networking Core	Terminal	
NET.IP.4	Threshold	HAIPEs shall be capable of reassembling fragmented CT ESP packets into a single packet up to 1500 bytes.	Networking Core	Terminal	
NET.IP.5	Objective	HAIPEs shall be capable of reassembling fragmented CT ESP packets into a single packet up to 8192 bytes.	Networking Core	Terminal	
NET.IP.V4.1	Threshold	HAIPE interfaces running IPv4 shall support a minimum MTU size of 576 bytes.	Networking Core	Terminal	
NET.IP.V4.CT.ICMP.1	Objective	HAIPEs shall support PMTU Discovery as specified in RFC 2401 and as modified herein.	Networking Core	Terminal	
NET.IP.V4.CT.ICMP.2	Objective	HAIPEs shall support ICMP Destination Unreachable Code 4 messages on the CT interface for PMTU Discovery.	Networking Core	Terminal	
NET.IP.V4.CT.ICMP.3	Objective	HAIPEs shall support ICMP Informational and Error messages as specified in RFC 792.	Networking Core	Terminal	
NET.IP.V4.CT.ICMP.4	Threshold	HAIPEs shall have the capability on a per-interface basis to accept or discard ICMP messages directed to it from the CT network.	Networking Core	Terminal	
NET.IP.V4.CT.ICMP.5	Threshold	The HAIPE capability to accept or discard ICMP messages from the CT network shall be configurable.	Networking Core	Terminal	
NET.IP.V4.CT.ICMP.6	Threshold	The HAIPE capability to accept ICMP traffic on the CT interface shall default to OFF.	Networking Core	Terminal	
NET.IP.V4.CT.ICMP.7	Threshold	HAIPEs shall not send ICMP Error messages in response to received multicast packets.	Networking Core	Terminal	
NET.IP.V4.CT.ICMP.8	Objective	HAIPEs shall support manual configuration of the MTU.	Networking Core	Terminal	

NET.IP.V4.CT.ICMP.9	Threshold	HAIPEs shall report a MTU value to the PT link.	Networking Core	Terminal	
NET.IP.V4.CT.IGMP.1	Threshold	HAIPEs shall support each of the messages listed in Table - NET.IP.V4.CT.IGMP.1 as specified in RFCs 1112, 2236 and 3376 and as modified herein.	Networking Core	Terminal	
NET.IP.V4.CT.IGMP.2	Threshold	HAIPEs shall populate the IGMPv3 message Source IP address field with the HAIPE CT interface IP address.	Networking Core	Terminal	
NET.IP.V4.CT.IGMP.3	Threshold	HAIPEs shall populate the IGMPv3 message Auxiliary Data Length field to ZERO.	Networking Core	Terminal	
NET.IP.V4.CT.IGMP.4	Threshold	HAIPEs shall populate the IGMPv3 message Number of Sources field to ZERO.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.1	Threshold	HAIPEs shall allow the administrator to select the IGMP Host method or IGMP Bypass method of IGMP support.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.2	Threshold	The HAIPE method of IGMP support shall default to the IGMP Host Method.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.3	Threshold	When configured as an IGMP host, HAIPEs shall support acting as an IGMPv3 host on both the CT and PT interfaces as specified in RFCs 1112, 2236 and 3376.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.4	Threshold	When configured as an IGMP host, HAIPEs shall send an unsolicited IGMPv3 Report message to all configured PT and CT multicast groups.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.5	Threshold	When configured as an IGMP host, HAIPEs shall send an IGMPv3 Report message in response to IGMPv3 Query messages for all configured PT and CT multicast groups as specified in RFC 3376.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.6	Threshold	When configured for IGMP Bypass, upon receipt of an IGMP Report message on the PT interface containing a multicast address that the HAIPE is configured with, HAIPEs shall map the PT multicast address to a CT multicast address, and send an IGMPv3 Report message to the CT network.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.7	Objective	HAIPEs shall have the capability of mapping PT multicast addresses to CT multicast address "one to one" or "many to one".	Networking Core	Terminal	
NET.IP.V4.MSGBYP.8	Threshold	When configured for IGMP Bypass, upon receipt of an IGMPv3 Multicast Address Specific Query on the CT interface, HAIPEs shall map the CT multicast address to a PT multicast address, and send an IGMPv3 Multicast Address Specific Query to the PT link.	Networking Core	Terminal	

NET.IP.V4.MSGBYP.9	Threshold	When configured for IGMP Bypass and processing IGMP messages from PT to CT and CT to PT, HAIPEs shall restrict the Multicast Address Specific Query, Multicast Address field to belonging to the set of member multicast addresses associated with key material.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.10	Threshold	HAIPEs shall not bypass IGMPv3 Report messages with an address in the Source List of the Multicast Address Record field that would expose PT addresses to the CT network or CT addresses to the PT network.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.11	Threshold	HAIPEs shall include the IP Router Alert option and set the value field equal to ZERO on any IGMPv3 message sent from either the PT or CT network interface as specified in RFC 2113, Section 2.1.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.12	Threshold	HAIPEs shall not transmit IGMPv3 Group and Address Specific Query messages that contain CT or PT addresses to the opposite interface.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.13	Threshold	HAIPEs shall not transmit IGMPv3 Report messages that contain CT or PT addresses to the opposite interface.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.14	Threshold	HAIPEs shall set the Time To Live (TTL) to ONE (1) when bypassing IGMPv3 messages to the other interface for transmission.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.15	Threshold	HAIPEs shall set the Type Of Service (ToS) bits to 0xc0 when bypassing IGMP messages to the other interface for transmission.	Networking Core	Terminal	
NET.IP.V4.MSGBYP.16	Threshold	HAIPEs shall not bypass any ICMPv4 Type 0 (Echo Request), Type 5 (Redirect), or Type 8 (Echo Reply) messages from CT to PT or PT to CT.	Networking Core	Terminal	
NET.IP.V4.PT.ICMP.1	Objective	Upon receipt of a packet on the PT interface that would result in the generation of an ICMP message (e.g., Echo Reply) HAIPEs shall generate the appropriate IPv4 ICMP message from the PT interface as specified in RFC 792.	Networking Core	Terminal	PT error/statistics reporting MAY not be necessary for terminal (non-INE).
NET.IP.V4.PT.ICMP.2	Objective	HAIPEs shall discard any packet with the DF bit set that exceeds the PMTU value and send an ICMP destination unreachable (Code 4) message to the offending PT host unless the ICMPv4 message would require the PT host to set the MTU to less than 576 bytes.	Networking Core	Terminal	
NET.IP.V4.PT.ICMP.3	Objective	HAIPEs shall have the capability to support ICMP Informational and Error	Networking Core	Terminal	

		messages as specified in RFC 792.			
NET.IP.V4.PT.ICMP.4	Threshold	HAIPEs shall not send ICMP Error (with the exception of Destination Unreachable - Code 4) messages in response to received multicast packets.	Networking Core	Terminal	
NET.IP.V4.PT.ICMP.5	Objective	Upon discarding a PT packet due to its TTL value, HAIPEs shall generate and transmit an ICMP time exceeded error message (Type 11, code 0) to the originator of the discarded Unicast PT packet.	Networking Core	Terminal	PT error/statistics reporting MAY not be necessary for terminal (non-INE).
NET.IP.V4.PT.IGMP.1	Threshold	HAIPEs shall support IGMPv3 (host membership query and report) messages in Table - NET.IP.V4.PT.IGMP.1 as specified in RFCs 1112, 2236 and 3376 and as modified herein.	Networking Core	Terminal	
NET.IP.V4.PT.IGMP.2	Threshold	HAIPEs shall populate the IGMPv3 message source IP address field with the HAIPE PT IP address.	Networking Core	Terminal	
NET.IP.V4.PT.IGMP.3	Threshold	HAIPEs shall populate the IGMPv3 message Auxiliary Data field to ZERO.	Networking Core	Terminal	
NET.IP.V4.PT.IGMP.4	Threshold	HAIPEs shall populate the IGMPv3 message Number of Sources field to ZERO.	Networking Core	Terminal	
NET.IP.V6.1	Threshold	HAIPE interfaces running IPv6 shall support a minimum MTU size of 1280 bytes.	Networking Core	Terminal	
NET.IP.V6.CT.1	Threshold	HAIPEs shall act as an IPv6 host on the CT interface as specified in RFCs 2460-2463 and as modified herein.	Networking Core	Terminal	
NET.IP.V6.CT.EXHDR.1	Threshold	HAIPEs shall use the Fragment Extension header on CT Packets to fragment encrypted packets exceeding the CT MTU if the CT MTU minus the ESP and Cryptographic Overhead is less than 1280 bytes.	Networking Core	Terminal	
NET.IP.V6.CT.EXHDR.2	Threshold	HAIPEs shall format IPv6 Extension Headers as specified in RFC 2460 and as modified herein.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.1	Threshold	HAIPEs shall perform checksum calculation and populate the checksum field to the calculated value in generated ICMPv6 messages.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ADD.1	Threshold	The HAIPE CT Interface shall support at least 2 global Unicast addresses (Current and Deprecated).	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ADD.2	Objective	The HAIPE CT Interface shall support at least 4 global Unicast addresses (2x Current and 2x Deprecated).	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ADD.3	Threshold	The HAIPE CT interface shall support manual configuration of the global scoped address prefix and Interface Identifier.	Networking Core	Terminal	

NET.IP.V6.CT.ICMP.ADD.4	Objective	The HAIPE CT interface Address Configuration method (manual or automated) shall be selectable.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ADD.SFAC.1	Objective	If no Router Advertisement is received, HAIPEs shall attempt to use Stateful Address Autoconfiguration to obtain addresses and other configuration information for the CT interface as specified in RFC 3315.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ADD.SLAC.1	Threshold	HAIPEs shall support Stateless Address Autoconfiguration on the CT interface as specified in RFC 2462 and as modified herein.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ADD.SLAC.2	Objective	HAIPEs shall support Privacy Extensions for Stateless Address Autoconfiguration on the CT interface, as specified in RFC 3484.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ADD.SLAC.3	Objective	HAIPEs shall support at least two router address entries in the Default Router List.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ADD.SLAC.4	Threshold	Upon receipt on the CT interface of a valid (as specified in RFC 2461) Router Advertisement, containing a prefix not already in the HAIPE database, HAIPEs shall form a global-scoped address using the procedures defined in RFC 3513.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ADD.SLAC.5	Threshold	HAIPE Stateless Address Autoconfiguration on the CT interface shall default to OFF.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ADD.SLAC.6	Threshold	HAIPEs shall support Duplicate Address Detection on the CT interface.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ADD.SLAC.7	Threshold	HAIPE support for Duplicate Address Detection on the CT interface shall be configurable.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ADD.SLAC.8	Threshold	HAIPE Duplicate Address Detection on the CT interface shall default to OFF.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ADD.SLAC.9	Threshold	HAIPEs shall simultaneously enable or disable Duplicate Address Detection and Stateless Address Autoconfiguration on the CT interface.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.INFO.1	Objective	HAIPEs shall support the receipt of unencrypted ICMPv6 Informational and Error messages received on the CT interface.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.INFO.2	Threshold	HAIPEs shall code ICMPv6 Informational and Error messages on the CT interface using Type and Code as specified in RFC 2463.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.INFO.3	Threshold	HAIPEs capability to support ICMPv6 Informational and Error messages (as specified in RFC 2463 Sections 3 and 4) on the CT interface shall be configurable.	Networking Core	Terminal	

NET.IP.V6.CT.ICMP.INFO.4	Threshold	HAIPEs capability to support ICMPv6 Informational and Error messages (as specified in RFC 2463 Sections 3 and 4) on the CT interface shall default to OFF.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.INFO.5	Threshold	HAIPEs shall not send ICMPv6 Error messages in response to received multicast packets.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.MLD.1	Threshold	HAIPEs shall support each of the messages listed in Table NET.IP.V6.CT.ICMP.MLD.1 as specified in RFC 3810 and as modified herein.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.MLD.2	Threshold	HAIPEs shall populate the MLD message source IP address field with the HAIPE CT link local IP address.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.MLD.3	Threshold	HAIPEs shall populate the MLD message Number of Sources field to ZERO.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ND.1	Threshold	HAIPEs shall support Neighbor Discovery functionality for hosts as specified in RFC 2461.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ND.2	Objective	HAIPEs shall support Neighbor Unreachability Detection as specified in RFC 2461.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.ND.3	Objective	HAIPEs shall support processing received Redirect messages as specified in RFC 2461 and as modified herein.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.PMTU.1	Objective	HAIPEs shall support PMTU Discovery as specified in RFC 2401 and as modified herein.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.PMTU.2	Objective	When a HAIPE CT interface receives a Packet Too Big message, it shall reduce its estimate of the PMTU for the relevant path, based on the value of the MTU field in the message.	Networking Core	Terminal	
NET.IP.V6.CT.ICMP.PMTU.3	Objective	HAIPEs shall support manual configuration of MTU.	Networking Core	Terminal	
NET.IP.V6.MSGBYP.1	Threshold	HAIPEs shall not bypass ICMPv6 Type 133 (Router Solicitation), Type 134 (Router Advertisement), Type 135 (Neighbor Solicitation), Type 136 (Neighbor Advertisement) or Type 137 (Redirect) messages from CT to PT or PT to CT.	Networking Core	Terminal	
NET.IP.V6.MSGBYP.2	Threshold	HAIPEs shall support IPv6 user multicast traffic.	Networking Core	Terminal	
NET.IP.V6.MSGBYP.3	Threshold	HAIPEs shall have the capability to allow the user to configure mapping of PT Multicast Addresses to CT Multicast Addresses.	Networking Core	Terminal	
NET.IP.V6.MSGBYP.4	Objective	HAIPEs shall have the capability of mapping PT multicast addresses to CT multicast address "one to one" or "many to one".	Networking Core	Terminal	
NET.IP.V6.MSGBYP.5	Threshold	HAIPEs shall allow the user to select the MLD Host method	Networking Core	Terminal	

		or MLD Bypass method of MLD support.			
NET.IP.V6.MSGBYP.6	Threshold	HAIPES' method of MLD support shall default to the MLD Host Method.	Networking Core	Terminal	
NET.IP.V6.MSGBYP.7	Threshold	When configured as an MLD host on both the PT and CT interfaces, the HAIPE shall send MLDv2 Report Messages (as specified in RFC 3810) to all configured PT and CT multicast groups associated with key material.	Networking Core	Terminal	
NET.IP.V6.MSGBYP.8	Threshold	When configured as an MLD host on both the PT and CT interface, the HAIPE shall send a MLDv2 report in response to all MLDv2 queries for all configured PT and CT multicast groups as specified in RFC 3810.	Networking Core	Terminal	
NET.IP.V6.MSGBYP.9	Threshold	When configured for MLD Bypass, upon receipt of a MLDv2 Report message on the PT interface containing a multicast address that the HAIPE is configured with, HAIPEs shall map the PT multicast address to a CT multicast address and send a MLDv2 report message to the CT router.	Networking Core	Terminal	
NET.IP.V6.MSGBYP.10	Threshold	When configured for MLD Bypass, upon receipt of a MLDv2 Multicast Address Specific Query on the CT interface, HAIPEs shall map the CT multicast address to a PT multicast address and send a MLDv2 query message to the PT link.	Networking Core	Terminal	
NET.IP.V6.MSGBYP.11	Threshold	When configured for MLD Bypass and processing MLD messages from PT to CT and CT to PT, HAIPEs shall restrict the Multicast Address Field in the Multicast Address Specific Query to the set of member multicast addresses associated with a PPK.	Networking Core	Terminal	
NET.IP.V6.MSGBYP.12	Threshold	HAIPEs shall not bypass MLDv2 Report messages that contain host CT or PT IP addresses within the Source List of the Multicast Address Record field.	Networking Core	Terminal	
NET.IP.V6.MSGBYP.13	Threshold	HAIPEs shall not bypass any Neighbor Discovery or ICMPv6 Information messages from CT to PT or PT to CT.	Networking Core	Terminal	
NET.IP.V6.MSGBYP.14	Threshold	HAIPEs shall not bypass ICMP error messages from CT to PT or PT to CT.	Networking Core	Terminal	
NET.IP.V6.MSGBYP.15	Threshold	HAIPEs shall set the Hop Limit to 1 when bypassing MLD messages to the other interface for transmission.	Networking Core	Terminal	
NET.IP.V6.PT.1	Threshold	HAIPEs shall act as an IPv6 router on the PT interface as specified in RFCs 2640-2643	Networking Core	Terminal	

		and as modified herein.			
NET.IP.V6.PT.ICMP.1	Threshold	HAIPEs shall perform checksum calculation and populate the checksum field to the calculated value in generated ICMPv6 messages.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.ADD.1	Threshold	HAIPEs shall support being configured with two 64-bit prefixes for the PT interface.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.ADD.2	Objective	HAIPEs shall support being configured with at least four 64-bit prefixes for the PT interface.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.ADD.3	Threshold	The HAIPE PT Interface shall support at least two global Unicast addresses (Current and Deprecated).	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.ADD.4	Objective	The HAIPE PT Interface shall support at least four global Unicast Addresses (two Current and two Deprecated).	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.ADD.5	Threshold	HAIPEs shall support manual configuration of the global scoped address prefix and Interface Identifier.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.ADD.SLAC.1	Threshold	HAIPEs shall support Stateless Address Autoconfiguration for routers as specified in RFC 2462 and as modified herein.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.ADD.SLAC.2	Threshold	HAIPEs shall discard any Router Solicitation messages received on the PT interface that do not meet the format specified in RFC 2461 and as modified herein.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.ADD.SLAC.3	Threshold	HAIPEs shall have the capability to perform Duplicate Address Detection as specified in RFC 2462.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.ADD.SLAC.4	Threshold	The ability to perform Duplicate Address Detection on the PT interface shall be configurable.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.ADD.SLAC.5	Threshold	The ability to perform Duplicate Address Detection on the PT interface shall default to OFF.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.INFO.1	Objective	When a HAIPE PT interface receives a packet addressed to it that requires the generation of an ICMPv6 message (e.g., Echo Reply), HAIPEs shall create the appropriate ICMPv6 message as specified in RFC 2463.	Networking Core	Terminal	PT error/statistics reporting MAY not be necessary for terminal (non-INE).
NET.IP.V6.PT.ICMP.INFO.2	Threshold	HAIPEs shall not send ICMPv6 Error messages in response to received multicast packets.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.INFO.3	Threshold	HAIPEs shall code ICMPv6 Informational and Error messages using Type and Code as specified in RFC 2463.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.INFO.4	Threshold	HAIPEs shall have the capability to support ICMPv6 Informational and Error	Networking Core	Terminal	

		messages received on the HAIPE PT interface.			
NET.IP.V6.PT.ICMP.INFO.5	Threshold	HAIPEs' capability to support ICMPv6 Informational and Error messages received on the HAIPE PT interface shall be configurable.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.INFO.6	Threshold	HAIPEs' capability to support ICMPv6 Informational and Error messages received on the HAIPE PT interface shall default to OFF.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.INFO.7	Objective	Upon discarding a PT packet due to its Hop Limit value, HAIPEs shall generate and transmit an ICMPv6 time exceeded error message (Type 3, code 0) to the originator of the discarded PT packet.	Networking Core	Terminal	PT error/statistics reporting MAY not be necessary for terminal (non-INE).
NET.IP.V6.PT.ICMP.MLD.1	Threshold	HAIPEs shall support each of the messages listed in Table - NET.IP.V6.PT.ICMP.MLD.1 as specified in RFC 3810 and as modified herein.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.MLD.2	Threshold	HAIPEs shall populate the MLD message Source IP address field with the HAIPE PT link local IP address.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.MLD.3	Threshold	HAIPEs shall populate the MLD message Number of Sources field to ZERO.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.ND.1	Threshold	HAIPEs shall support ND functionality for routers as specified in RFC 2461 and as modified herein.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.ND.2	Objective	HAIPEs shall support Redirect Messages as specified in RFC 2461 and as modified herein.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.ND.3	Objective	HAIPEs shall support Neighbor Unreachability Detection as specified in RFC 2461.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.PMTU.1	Objective	HAIPEs shall discard any packet exceeding the PMTU value and send an ICMP Packet Too Big message to the offending PT host unless the ICMPv6 message would require the PT host to set the MTU to less than 1280 bytes.	Networking Core	Terminal	PT error/statistics reporting MAY not be necessary for terminal (non-INE).
NET.IP.V6.PT.ICMP.PMTU.2	Objective	HAIPEs shall report the CT PMTU to the PT network minus ESP overhead.	Networking Core	Terminal	
NET.IP.V6.PT.ICMP.PMTU.3	Objective	HAIPEs shall support reporting an MTU value to the PT link.	Networking Core	Terminal	PT error/statistics reporting MAY not be necessary for terminal (non-INE).
TP.CRYPT.SUITEA.KFILL.1	Threshold	HAIPEs shall support accepting key loads in accordance with EKMS 308.	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.KFMT.1	Threshold	HAIPEs shall be able to assign for use any PPK with valid DS100 Tag Parity, Key Parity and Parity Prime fields at any point in time (before, on, after)	Traffic Protection - Suite A Cryptography	Terminal POET ACM	Key validation performed by the POET ACM.

		relative to the effective date.			
TP.CRYPT.SUITEA.KFMT.2	Threshold	HAIPEs shall accept properly formatted EFF vector sets in accordance with EKMS 322.	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.KFMT.3	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.KFMT.4	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.KFMT.5	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.KFMT.6	Threshold	When filled with a PPK, HAIPEs shall calculate the DS100-1 Tag Parity in accordance with EKMS 308 and discard the PPK if the calculated value does not equal the 'Tag Parity' field.	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.KFMT.7	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.KMA.1	Threshold	HAIPE shall interpret effective dates as 0000 Greenwich Mean Time (GMT) by convention.	Traffic Protection - Suite A Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEA.KMA.2	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.KMA.3	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite A Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEA.KMA.4	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite A Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEA.KMA.5	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.KMA.6	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite A Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEA.KMA.7	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite A Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEA.KMA.8	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite A Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEA.KMA.9	Threshold	HAIPEs shall support the use of ACCORDION 3.0 in accordance with R21-TECH-03-02, An ACCORDION MEDLEY, dated 7 Feb 2002, for performing a Deterministic Update on a TEK.	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.KMA.10	Threshold	HAIPEs shall use the first Deterministic Update of the loaded PPK as the first TEK.	Traffic Protection - Suite A Cryptography	Terminal	
TP.CRYPT.SUITEA.KMA.11	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.KMA.12	Threshold	HAIPEs shall perform subsequent Deterministic Updates to the PPK using the result of the previous update.	Traffic Protection - Suite A Cryptography	Terminal POET ACM	

TP.CRYPT.SUITEA.KMA.13	Threshold	HAIPEs shall, when Exclusion Key is negotiated via IKE, perform a MOD-2 addition of the EFF calculated TEK with the negotiated EK(s), using the result as the TEK.	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.KMA.14	Threshold	HAIPE TEK creation or connection termination, as a function of Exclusion Keys, shall be per "Exclusion Key and it's Application to Foreign Interoperability", 7 June 2002 and "Guidance for Exclusion Key Specification", 8 Nov 2004 and as defined in this specification.	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.CEA.1	Threshold	HAIPEs shall encrypt data using MEDLEY in accordance with NSA MEDLEY Implementation Standard R21-TECH-30-01 and in a mode of operation as defined in Galois/Counter Mode of Operation (GCM) I311-038-04.	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.CEA.2	Threshold	HAIPEs shall appliy confidentiality to the following fields: IP datagram, TFC padding (Tunnel mode only), Cryptographic Padding, Pad Length, and Next Header.	Traffic Protection - Suite A Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEA.CEA.3	Threshold	HAIPEs shall created a 128 bit GHASH value by creating a Galois Counter Mode (GCM) Hash (GHASH(H,{},IV)) in accordance with Galois/Counter Mode of Operation (GCM) I311-038-04 to construct the initial State Variable for loading into the Galois/Counter Mode 128 bit SV register for each packet encrypted/decrypted.	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.CEA.4	Threshold	HAIPEs shall construct the IV that is used in the SV GHASH function as a concatentation of 32 bit (IPv4) or 128 bit (IPv6) CT source address, 32 bit SPI value, and 64 bit sequence number.	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.CEA.5	Threshold	HAIPEs shall transmit the high order 32-bits of the Extended Sequence Number in the 32-bit ESPv3 IV field.	Traffic Protection - Suite A Cryptography	Terminal	
TP.CRYPT.SUITEA.CEA.6	Threshold	HAIPEs shall format the 128 bit SV as specified in Table TP.CRYPT.SUITEA.CEA.6:	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.CEA.7	Threshold	HAIPEs shall populate the 128 bit SV/Per-block Counter with the calculated 128 bit GHASH value prior to performing the GCM encryption/decryption process for each packet transmitted/received.	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.CEA.8	Threshold	HAIPEs shall not allow the lower 32-bit SV Per Block Counter (PBC) to carry over from the MSb (bit-31 of PBC 31-0) into the upper 96-bits of the SV.	Traffic Protection - Suite A Cryptography	POET ACM	

TP.CRYPT.SUITEA.CEA.9	Threshold	HAIPEs shall populate the ICV field of the ESP payload with the number of bits negotiated in the Data Integrity Algorithm of the Galois Counter Mode (GCM) hash (n MSbs of (GHASH(H,A,C) XOR E(K,Y0))) in accordance with Galois/Counter Mode of Operation (GCM) I311-038-04.	Traffic Protection - Suite A Cryptography	POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.CRYPT.SUITEA.CEA.10	Threshold	HAIPEs shall provide integrity using GCM to the following fields: IP datagram, TFC Padding (Tunnel Mode Only), Padding, Pad Length, and Next Header.	Traffic Protection - Suite A Cryptography	POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.CRYPT.SUITEA.CEA.11	Threshold	HAIPEs shall discard all packets received that fail integrity checks.	Traffic Protection - Suite A Cryptography	POET ACM	
TP.CRYPT.SUITEA.CEA.12	Threshold	HAIPEs shall have 0-3 bytes of cryptographic padding.	Traffic Protection - Suite A Cryptography	Terminal	
TP.CRYPT.SUITEA.CEA.13	Threshold	HAIPEs shall drop any packet with a value of 0xFF in the Pad Length field.	Traffic Protection - Suite A Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEA.CEA.14	Threshold	The receiving HAIPE shall not use/validate Bits (7-2) of the Pad Length field. FUTURE: Bit 2 = '0' shall indicate that Bits (7-3) are Reserved. Bit 2 = '1' shall indicate that bits (7-3) will be in-band signaling.	Traffic Protection - Suite A Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEA.CEA.15	Threshold	HAIPEs shall populate padding bytes with 0x00.	Traffic Protection - Suite A Cryptography	Terminal	
TP.CRYPT.SUITEB.KFILL.1	Threshold	HAIPEs shall support accepting key loads in accordance with EKMS 308.	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.KFMT.1	Threshold	HAIPEs shall be able to assign for use any valid formatted Authenticated PPK at any point in time (before, on, after) relative to the effective date.	Traffic Protection - Suite B Cryptography	Terminal POET ACM	Key validation performed by the POET ACM.
TP.CRYPT.SUITEB.KFMT.2	Threshold	HAIPEs shall accept properly formatted EFF vector sets in accordance with EKMS 322.	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.KFMT.3	Threshold	HAIPE shall be capable of being filled with PPK material containing a DS-100-1 Tag as specified by EKMS 308, and as modified herein, with a Data field in accordance with Authenticated HAIPE IS Foreign Interoperability PPK Specification Draft (Authenticated PPK material).	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.KFMT.4	Threshold	HAIPE shall be capable of validating the authentication information present in the authenticated PPK material in accordance with Authenticated HAIPE IS Foreign Interoperability PPK Specification Draft Version .01 dated 22 February 2005.	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.KFMT.5	Threshold	HAIPE shall discard the PPK if the Key Material fails authentication check.	Traffic Protection - Suite B Cryptography	POET ACM	

TP.CRYPT.SUITEB.KFMT.6	Threshold	When filled with a PPK, HAIPESs shall calculate the Key Parity in accordance with Appendix A.2 of EKMS 322 over the 32 byte Data field (key material) and discard the PPK if the calculated value does not equal the 'Key Parity' field.	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.KFMT.7	Threshold	When filled with a PPK, HAIPEs shall calcualte the DS100-1 Tag Parity in accordance with EKMS 308 and discard the PPK if the calculated value does not equal the 'Tag Parity' field.	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.KMA.1	Threshold	HAIPE shall interpret effective dates as 0000 Greenwich Mean Time (GMT) by convention.	Traffic Protection - Suite B Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEB.KMA.2	Threshold	HAIPE shall use the first Deterministic Update of the loaded PPK as the first TEK.	Traffic Protection - Suite B Cryptography	Terminal	
TP.CRYPT.SUITEB.KMA.3	Threshold	HAIPEs shall support the use of a Key Update Function in accordance with R21-TECH-02-05, dated 10 January 2005 for performing a Deterministic Update on a TEK.	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.KMA.4	Threshold	HAIPEs shall perform subsequent Deterministic Updates to the PPK using the result of the previous update.	Traffic Protection - Suite B Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEB.KMA.5	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.KMA.6	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite B Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEB.KMA.7	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite B Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEB.KMA.8	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.KMA.9	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite B Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEB.KMA.10	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite B Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEB.KMA.11	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Suite B Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEB.KMA.12	Threshold	HAIPEs shall, when Exclusion Key is negotiated via IKE, perform a MOD-2 addition of the EFF calculated TEK with the negotiated EK(s), using the result as the TEK.	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.KMA.13	Threshold	HAIPE TEK creation or connection termination, as a function of Exclusion Keys, shall be per "Exclusion Key and it's Application to Foreign Interoperability", 7 June 2002 and "Guidance for Exclusion Key Specification", 8 Nov 2004 and as defined in this	Traffic Protection - Suite B Cryptography	POET ACM	

		specification.			
TP.CRYPT.SUITEB.CEA.1	Threshold	HAIPEs shall encrypt data using AES in Counter Mode as defined in Galois/Counter Mode of Operation (GCM) I311-038-04.	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.CEA.2	Threshold	HAIPEs shall apply confidentiality to the following fields: IP datagram, TFC padding (Tunnel mode only), Cryptographic Padding, Pad Length, and Next Header.	Traffic Protection - Suite B Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEB.CEA.3	Threshold	HAIPEs shall create a 128 but GHASH value by creating a Galois Counter Mode (GCM) Hash (GHASH(H,{},IV)) in accordance with Galois/Counter Mode of Operation (GCM) I311-038-04 to construct the initial State Variable for loading into the Galois/Counter Mode 128 bit SV register for each packet encrypted/decrypted.	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.CEA.4	Threshold	HAIPEs shall construct the IV used in the SV GHASH function as a concatentation of 32 bit (IPv4) or 128 bit (IPv6) CT source address, 32 bit SPI value, and 64 bit sequence number.	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.CEA.5	Threshold	HAIPEs shall transmit the high order 32-bits of the Extended Sequence Number in the 32-bit ESPv3 IV field.	Traffic Protection - Suite B Cryptography	Terminal	
TP.CRYPT.SUITEB.CEA.6	Threshold	HAIPEs shall format the 128 bit SV as specified in Table TP.CRYPT.SUITEB.CEA.6:	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.CEA.7	Threshold	HAIPEs shall populate the 128 bit SV/Per-block Counter with the calculated 128 bit GHASH value prior to performing the GCM encryption/decryption process for each packet transmitted/received.	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.CEA.8	Threshold	HAIPEs shall not allow the lower 32-bit SV Per Block Counter (PBC) to carry over from the MSb (bit-31 of PBC 31-0) into the upper 96-bits of the SV.	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.CEA.9	Threshold	HAIPEs shall populate the ICV field of the ESP payload with the number of bits negotiated in the Data Integrity Algorithm of the Galois Counter Mode (GCM) hash (n MSbs of (GHASH(H,A,C) XOR E(K,Y0))) in accordance with Galois/Counter Mode of Operation (GCM) I311-038-04.	Traffic Protection - Suite B Cryptography	POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.CRYPT.SUITEB.CEA.10	Threshold	HAIPEs shall provide integrity using GCM to the following fields: IP datagram, TFC Padding (Tunnel Mode Only), Padding, Pad Length, and Next Header.	Traffic Protection - Suite B Cryptography	POET ACM	Assuming ICV is computed and stored by the POET ACM.

TP.CRYPT.SUITEB.CEA.11	Threshold	HAIPEs shall discard all packets received that fail integrity checks.	Traffic Protection - Suite B Cryptography	POET ACM	
TP.CRYPT.SUITEB.CEA.12	Threshold	HAIPEs shall have 0-3 bytes of cryptographic padding.	Traffic Protection - Suite B Cryptography	Terminal	
TP.CRYPT.SUITEB.CEA.13	Threshold	HAIPEs shall drop any packet with a value of 0xFF in the Pad Length field.	Traffic Protection - Suite B Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEB.CEA.14	Threshold	The receiving HAIPE shall not use/validate Bits (7-2) of the Pad Length field. FUTURE: Bit 2 = '0' shall indicate that Bits (7-3) are Reserved. Bit 2 = '1' shall indicate that bits (7-3) will be in-band signaling.	Traffic Protection - Suite B Cryptography	Terminal POET ACM	
TP.CRYPT.SUITEB.CEA.15	Threshold	HAIPEs shall populate padding bytes with 0x00.	Traffic Protection - Suite B Cryptography	Terminal	
TP.CRYPT.LEG.KFILL.1	Threshold	HAIPEs shall support accepting key loads in accordance with EKMS 308.	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.KFMT.1	Threshold	HAIPEs shall be able to assign for use any PPK with valid DS100 Tag Parity, Key Parity and Parity Prime fields at any point in time (before, on, after) relative to the effective date.	Traffic Protection - Legacy Cryptography	Terminal	
TP.CRYPT.LEG.KFMT.2	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.KFMT.3	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.KFMT.4	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.KFMT.5	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.KFMT.6	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.KMA.1	Threshold	HAIPE shall interpret effective dates as 0000 Greenwich Mean Time (GMT) by convention.	Traffic Protection - Legacy Cryptography	Terminal POET ACM	
TP.CRYPT.LEG.KMA.2	Threshold	HAIPEs shall accept properly formatted EFF vector sets in accordance with EKMS 322.	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.KMA.3	Threshold	HAIPEs shall support the use of ACCORDION 3.0 in accordance with R21-TECH-03-02, An ACCORDION MEDLEY, dated 7 Feb 2002, for performing a Deterministic Update on a TEK that is used to protect MEDLEY encrypted traffic.	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.KMA.4	Threshold	HAIPEs shall support the use of ACCORDION 1.3 in accordance with KM-TG-0001-87, ACCORDION 1.3, dated 30 Oct 1987, for performing a Deterministic Update on a TEK that is used to protect BATON encrypted traffic.	Traffic Protection - Legacy Cryptography	POET ACM	

TP.CRYPT.LEG.KMA.5	Threshold	In the MEDLEY mode, prior to the first Deterministic Update, HAIPEs shall use the PPK, as loaded (1st bit received equals Msbit), as the first TEK.	Traffic Protection - Legacy Cryptography	Terminal	
TP.CRYPT.LEG.KMA.6	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.KMA.7	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	Terminal POET ACM	
TP.CRYPT.LEG.KMA.8	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	Terminal	
TP.CRYPT.LEG.KMA.9	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	Terminal POET ACM	
TP.CRYPT.LEG.KMA.10	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	Terminal POET ACM	
TP.CRYPT.LEG.KMA.11	Threshold	HAIPEs shall support Universal Changeover as defined in EKMS 322.	Traffic Protection - Legacy Cryptography	Terminal	
TP.CRYPT.LEG.KMA.12	Threshold	HAIPEs shall not allow configuration of a single PPK to both BATON and MEDLEY.	Traffic Protection - Legacy Cryptography	Terminal POET ACM	
TP.CRYPT.LEG.KMA.13	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	Terminal POET ACM	
TP.CRYPT.LEG.KMA.14	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	Terminal POET ACM	
TP.CRYPT.LEG.KMA.15	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	Terminal POET ACM	
TP.CRYPT.LEG.KMA.16	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.CEA.1	Threshold	HAIPEs shall support the encryption of data using MEDLEY in accordance with NSA MEDLEY Implementation Standard R21-TECH-30-01 and in a mode of operation as defined in the HAIPE Legacy Encryption Implementation Guide dated March 31, 2005.	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.CEA.2	Threshold	HAIPEs shall support the encryption of data using BATON in accordance with NSA BATON Implementation Standard R21-TECH-44-97 and in a mode of operation as defined in the HAIPE Legacy Encryption Implementation Guide dated March 31, 2005.	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.CEA.3	Threshold	HAIPEs shall perform the Integrity operation before Encryption.	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.CEA.4	Threshold	HAIPEs shall calculate the Integrity operation to include the SPI, Payload Sequence Number (PSEQN), PT IP Header, PT IP Payload, Padding, Dummy, Pad Length and Next Header fields.	Traffic Protection - Legacy Cryptography	POET ACM	
TP.CRYPT.LEG.CEA.5	Threshold	HAIPEs shall calculate the Integrity Function as a BIP-32,	Traffic Protection - Legacy	POET ACM	

		the result of a MOD-2 addition function of the 32 bit aligned words, the result of which is MOD-2 added with a 32 bit word of Hexadecimal A's.	Cryptography		
TP.CRYPT.LEG.CEA.6	Threshold	CLASSIFIED REQUIREMENT	Traffic Protection - Legacy Cryptography	POET ACM	
MGMT.1	Threshold	HAIPEs shall use a two-byte "Fixed ID" as registered in EKMS 308.	Management Extension	Terminal	
MGMT.2	Threshold	HAIPEs shall use the format defined in Table - MGMT.2 for the 48 bit burned-in ECU ESN.	Management Extension	Terminal	
MGMT.3	Threshold	HAIPEs shall populate the Station ID with 12 ASCII characters (upper case hexadecimal representation of the 48-bit ECU ESN) as shown in Table MGMT.3 and an additional 2 ASCII characters of space padding.	Management Extension	Terminal	
MGMT.4	Threshold	HAIPEs shall pre-configure the deviceVersionTable as follows: dvName is "Management", and dvVersion is "3.0.0".	Management Extension	Terminal	
MGMT.FD.1	Threshold	HAIPEs shall provide the capability to remotely initiate a download of a firmware image using Trivial File Transfer Protocol (TFTP).	Management Extension	Terminal	
MGMT.FD.2	Threshold	HAIPEs shall provide the capability to configure one or more Uniform Resource Identifiers (URIs) with IP Addresses and remote path for firmware download using TFTP.	Management Extension	Terminal	
MGMT.FD.3	Objective	HAIPEs shall provide the capability to configure one or more URIs with Fully Qualified Domain Names for firmware download using TFTP.	Management Extension	Terminal	
MGMT.FD.4	Threshold	HAIPEs shall implement TFTP processing as specified in RFC 1350 for transport of firmware images between PT servers and HAIPE devices.	Management Extension	Terminal	
MGMT.FD.5	Threshold	HAIPEs shall format TFTP messages as specified in RFC 1350.	Management Extension	Terminal	
MGMT.FD.6	Threshold	HAIPEs shall send a TFTP Read Request (RRQ) message to the configured URI to begin a firmware download.	Management Extension	Terminal	
MGMT.FD.7	Threshold	HAIPEs shall send RRQ messages using a mode of netascii as specified in RFC 1350.	Management Extension	Terminal	
MGMT.FD.8	Threshold	HAIPEs shall discard any received TFTP Write Requests (WRQ).	Management Extension	Terminal	
MGMT.FD.9	Threshold	HAIPEs shall send a NOTIFICATION message to the configured manager containing the appropriate TFTP Error Code upon receipt of a TFTP Error Message.	Management Extension	Terminal	

MGMT.FD.10	Objective	HAIPEs shall allow for the configuration of the desired block size to use for TFTP transfers.	Management Extension	Terminal	
MGMT.FD.11	Objective	HAIPEs shall have a capability to negotiate the block size used for TFTP transfers as specified in RFCs 2347 and 2348.	Management Extension	Terminal	
MGMT.FD.12	Objective	HAIPEs shall allow for the configuration of the TFTP maximum file transfer size supported.	Management Extension	Terminal	
MGMT.FD.13	Objective	HAIPEs shall have a capability to negotiate the TFTP Maximum File Size as specified in RFCs 2347 and 2349.	Management Extension	Terminal	
MGMT.FD.14	Objective	HAIPEs shall allow for the configuration of the desired TFTP Timeout Interval.	Management Extension	Terminal	
MGMT.FD.15	Objective	HAIPEs shall have a capability to negotiate the TFTP Timeout Interval as specified in RFCs 2347 and 2349.	Management Extension	Terminal	
MGMT.FD.16	Objective	HAIPEs shall utilize the Hyper Text Transfer Protocol (HTTP) Version 1.1 as specified in RFC 2616 for transport of firmware images between PT servers and HAIPE devices.	Management Extension	Terminal	
MGMT.FD.17	Objective	HAIPEs shall provide the capability to remotely initiate a download of a firmware image using HTTP.	Management Extension	Terminal	
MGMT.FD.18	Objective	HAIPEs shall initiate a firmware download connection request using a HTTP/1.1 GET as defined in RFC 2616.	Management Extension	Terminal	
MGMT.FD.19	Objective	HAIPEs shall download firmware images using a persistent HTTP connection as defined in Section 8.1.1 of RFC 2616 .	Management Extension	Terminal	
MGMT.FD.20	Objective	HAIPEs shall send a Connection Header with the connection-token "close" upon receipt of the entire firmware image.	Management Extension	Terminal	
MGMT.FD.21	Objective	HAIPEs shall be capable of understanding HTTP status codes as defined in Section 10 of RFC 2616.	Management Extension	Terminal	
MGMT.FD.22	Objective	HAIPEs shall provide the capability to configure one or more Uniform Resource Identifiers (URIs) with IP Addresses and remote path for firmware download using HTTP.	Management Extension	Terminal	
MGMT.FD.23	Objective	HAIPEs shall provide the capability to configure one or more URIs with Fully Qualified Domain Names for firmware download using HTTP.	Management Extension	Terminal	
MGMT.HMI.1	Threshold	HAIPEs shall provide a Human Machine Interface (HMI) for loading configuration information necessary to	Management Extension	Terminal	

		establish a Security Association (SA) with a peer HAIPE.			
MGMT.HMI.2	Threshold	HAIPEs shall provide a HMI for loading configuration information necessary to establish a secure connection with a HAIPE Management Station.	Management Extension	Terminal	
MGMT.HMI.3	Threshold	HAIPEs shall provide local access control for the administrator/operator.	Management Extension	Terminal	
MGMT.HMI.4	Threshold	HAIPEs shall allow the administrator/operator to set the HAIPE real-time clock.	Management Extension	Terminal	Current architecture has the real-time clock outside of the POET ACM boundary. If this changes, the allocation of this requirement will need to be re-evaluated.
MGMT.HMI.5	Objective	HAIPEs shall provide the capability to allow the administrator/operator to view existing SAs.	Management Extension	Terminal	
MGMT.HMI.6	Threshold	HAIPEs shall provide the capability to assign PPK effective dates (month and year).	Management Extension	Terminal	
MGMT.HMI.7	Threshold	HAIPEs shall allow the user to enable IPv4 on the PT and CT interfaces, IPv6 on the PT and CT interfaces, or both IPv4 and IPv6 on the PT and CT interfaces.	Management Extension	Terminal	
MGMT.HMI.8	Threshold	HAIPEs shall provide the capability to configure independent/unique IP Addresses and associated subnet masks for each interface.	Management Extension	Terminal	
MGMT.HMI.9	Threshold	HAIPEs shall provide the capability to configure an IPv4 and an IPv6 default gateway for the CT and PT network.	Management Extension	Terminal	
MGMT.HMI.10	Threshold	HAIPEs shall provide the administrator/operator with the capability to view the Software Version of the current executing software as well as the Version Identifiers of any other images stored in the device.	Management Extension	Terminal POET ACM	Version information must be provided upon request.
MGMT.HMI.11	Threshold	HAIPEs shall provide the administrator/operator with the capability to view the following tag information for any valid, loaded symmetric key which is accessible with the current CIK: Short Title, Edition, Register Number, Segment Number, and Classification.	Management Extension	Terminal POET ACM	Key information must be provided upon request.
MGMT.HMI.12	Threshold	HAIPEs shall provide the capability to configure incoming and outgoing selectors in the SPD.	Management Extension	Terminal	

MGMT.HMI.13	Threshold	HAIPEs shall provide the capability to assign processing requirements (PROTECT, DISCARD, BYPASS) for each SPD entry.	Management Extension	Terminal	
MGMT.HMI.14	Threshold	HAIPEs shall provide the capability to configure the Confidentiality Algorithm, PPK Short Title Selection, Peer HAIPE PT IP Address, Peer HAIPE CT IP Address, current and next PPK SPIs (ESPv3 only), and ESP version parameters for Manual Security Associations.	Management Extension	Terminal POET ACM	Connection information must be configurable from the terminal.
MGMT.HMI.15	Threshold	HAIPEs shall provide the capability to view and clear audit data.	Management Extension	Terminal	
MGMT.HMI.16	Threshold	HAIPEs shall provide the capability to assign a Changeover Key.	Management Extension	Terminal POET ACM	
MGMT.HMI.17	Threshold	HAIPEs shall provide the capability to manually configure Local Enclave Prefix Information.	Management Extension	Terminal	
MGMT.HMI.18	Threshold	HAIPEs shall provide the capability to manually configure Remote HAIPE Enclave Prefix Information (HAIPE CT and PT IP Addresses and associated PT Host IP Addresses).	Management Extension	Terminal	
MGMT.HMI.19	Threshold	HAIPEs shall provide the capability to assign (at any point - before, on or after - relative to the effective date), the EK "Key Type" (Global, Local, Privacy, Separate).	Management Extension	Terminal POET ACM	
MGMT.HMI.20	Threshold	HAIPEs shall provide the capability to assign (at any point - before, on or after - relative to the effective date) association of the EK to the corresponding Universal ID.	Management Extension	Terminal POET ACM	
MGMT.NET.1	Threshold	HAIPEs shall support the use of Simple Network Management Protocol version 3 (SNMPv3) for network management operations, as specified in IETF STD #62 (IETF RFCs 3411-3418).	Management Extension	Terminal	
MGMT.NET.2	Threshold	HAIPEs shall support sending SNMPv3 notifications over UDP to port 162.	Management Extension	Terminal	
MGMT.NET.3	Threshold	HAIPEs shall support the MIBs specified in the conformance requirements detailed in the HAIPE-V3-MIB, which will include requirements for both HAIPE specific MIBs and IETF specific Management Information Bases (MIBs) .	Management Extension	Terminal	
MGMT.NET.4	Threshold	HAIPEs shall not allow unauthenticated or unauthorized read or write access to the MIBs.	Management Extension	Terminal POET ACM	POET ACM MAY be required to perform authentication (e.g., classified configuration data).

MGMT.NET.5	Threshold	HAIPEs shall support security services (i.e., end-to-end authentication) specified in RFC 3414 (User-based Security Model (USM) for SNMPv3) and as modified herein for all communication between management devices (i.e., HAIPE Management Station (HMS) and the managed HAIPEs).	Management Extension	Terminal POET ACM	POET ACM MAY be required to perform authentication (e.g., classified configuration data).
MGMT.NET.6	Threshold	HAIPEs shall support the use of confidentiality and integrity protection to Over The Network Management traffic using the AES Cipher Algorithm in Cipher Feedback Mode, with a 128 bit key as specified in RFC 3826 and HMAC-SHA-96 as specified in RFC 3414.	Management Extension	Terminal POET ACM	POET ACM MAY be required to perform authentication (e.g., classified configuration data).
MGMT.NET.7	Threshold	HAIPEs SNMP-VIEW-BASED-ACM-MIB::vacmAccessSecurityLevel object, shall not accept any value other than SnmpSecurityLevel authPriv (3).	Management Extension	Terminal	
MGMT.NET.8	Threshold	HAIPEs shall default to an initial security configuration of initial-no-access-configuration per RFC 3415.	Management Extension	Terminal	
MGMT.NET.9	Threshold	On receipt of an ICMP Packet Too Big message on the CT interface which would result in setting the PT MTU less than 1280 bytes, HAIPEs shall send a Notification message to the configured manager using a Notification referenced by the HAIPE-V3-MIB.	Management Extension	Terminal	
MGMT.NET.10	Threshold	On receipt of an ICMP Destination Unreachable (Type 3, Code 4) message on the CT interface which would result in setting the PT MTU less than 512 bytes, HAIPEs shall send a Notification message to the configured manager using a Notification referenced by the HAIPE-V3-MIB.	Management Extension	Terminal	
NET.DISC.PD.GENCL.1	Threshold	HAIPEs shall allow for the manual configuration of a Peer Enclave Prefix Table containing Remote Prefixes, associated Peer HAIPE PT and CT IP Addresses, and Administrative Costs.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.2	Threshold	If the HAIPE has multiple Peer HAIPE PT and CT IP Addresses associated with a Remote Prefix, HAIPEs shall select the most specific PT prefix, then select by the lowest Administrative Cost.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.3	Objective	HAIPEs shall delete information on a Peer in the Peer Enclave Prefix Table if the HAIPE is unsuccessful at creating a Security Association	Generic Discovery Client Extension	Terminal	

		with the Peer.			
NET.DISC.PD.GENCL.4	Threshold	HAIPEs shall allow configuration of the PT UDP port on which all Generic Discovery Client messages will be sent and received.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.5	Threshold	HAIPEs shall format Generic Discovery Client messages as properly formatted PT UDP packets as specified in RFC 768.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.6	Threshold	HAIPEs shall code the Message Type field in Generic Discovery Client messages with the values described in Table NET.DISC.PD.GENCL.6.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.7	Threshold	HAIPEs shall support configuration of an Administrative Cost to assign to local Prefixes that are learned using automated mechanisms.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.8	Threshold	HAIPEs shall allow for the manual configuration of the Registration Transmission Address Table containing an ordered list of PT (and associated CT) addresses to which the HAIPE will send REGISTRATION messages.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.9	Threshold	Upon configuration (including startup and reboot) with a Registration Transmission Address and associated key material, HAIPEs shall send a REGISTRATION message to all addresses in the Registration Transmission Address Table, containing the prefixes in the Local Enclave Prefix Table.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.10	Threshold	HAIPEs shall format the Prefix information as shown in Table NET.DISC.PD.GENCL.10.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.11	Threshold	HAIPEs shall code Prefix Actions as shown in Table NET.DISC.PD.GENCL.11.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.12	Threshold	HAIPEs shall format the REGISTRATION message as shown in Table NET.DISC.PD.GENCL.12.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.13	Threshold	HAIPEs shall send a REGISTRATION message populated only with changes, to all addresses in the Registration Transmission Address Table, every time a change occurs to the Local Enclave Prefix Table.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.14	Threshold	HAIPEs shall send a REGISTRATION message, to all addresses in the Registration Transmission Address Table, when the Prefix Lifetime sent in a previous REGISTRATION expires.	Generic Discovery Client Extension	Terminal	

NET.DISC.PD.GENCL.15	Threshold	HAIPEs shall retransmit REGISTRATION messages to all addresses in the Registration Transmission Address Table until receiving one REGISTRATION/ACK, formatted as shown in Table NET.DISC.PD.GENCL.15.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.16	Threshold	HAIPEs shall have a configurable timer indicating the amount of time to wait for a REGISTRATION/ACK message before retransmitting the REGISTRATION message.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.17	Threshold	HAIPEs shall allow configuration of the REGISTRATION message re-transmittal counter. Note: A configuration value of ZERO would indicate continuously sending the message until receiving an ACK.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.18	Threshold	HAIPEs shall allow manual configuration of the Solicitation Transmission Address Table containing an ordered list of PT (and associated CT) addresses to which the HAIPE will send Solicitation messages.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.19	Threshold	Upon receipt of a packet on the PT interface for which the SPD requires protection, HAIPEs shall create or use an existing SA with the Peer HAIPE associated with the most specific prefix in the Peer Enclave Prefix Table.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.20	Threshold	Upon receipt of a packet on the PT interface for which there is not a matching Prefix in the Peer Enclave Prefix Table, the HAIPE shall send a Solicitation QUERY message to the first address in the Solicitation Transmission Address Table.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.21	Threshold	HAIPEs shall format Solicitation QUERY messages as shown in Table NET.DISC.PD.GENCL.21.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.22	Threshold	HAIPEs shall have a configurable timer indicating the amount of time to wait for a Solicitation RESPONSE message before retransmitting the Solicitation QUERY.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.23	Objective	HAIPEs shall allow configuration of the Solicitation QUERY message re-transmittal counter.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.24	Threshold	Upon reaching the configured Solicitation QUERY message retransmit counter, HAIPEs shall send a Solicitation QUERY message to the next configured IP Address in the Solicitation Transmission Address table.	Generic Discovery Client Extension	Terminal	

NET.DISC.PD.GENCL.25	Threshold	HAIPEs shall allow for the configuration of a Solicitation Reception Address Table containing an ordered list of PT (and associated CT) addresses to which the HAIPE will listen for Solicitation QUERY messages.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.26	Threshold	Upon receipt of a Solicitation QUERY message, for which a match is found for the Target Host IP Address value in the Local Enclave Prefix Table, the HAIPE shall send a Solicitation RESPONSE message formatted as shown in Table NET.DISC.PD.GENCL.26 to the Source HAIPE identified in the QUERY message .	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.27	Threshold	HAIPEs shall set the option count to 0x00 when they transmit a query message	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.28	Threshold	HAIPEs shall be able to process a received QUERY message containing one or more options even if the HAIPE does not recognize one or more of the option types associated with these options.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.29	Threshold	Upon receipt of a Solicitation RESPONSE message, HAIPEs shall associate the target host with an existing SA for the Target HAIPE or initiate an IKE exchange to create a SA with the HAIPE fronting the target prefix.	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.GENCL.30	Threshold	HAIPEs shall pre-configure the deviceVersionTable as follows: dvName is "GenericDiscoveryClient", and dvVersion is "3.0.0".	Generic Discovery Client Extension	Terminal	
NET.DISC.PD.LEG.1	Threshold	HAIPEs shall be able to use Legacy Discovery to dynamically learn the PT and CT IP Addresses of a peer and create a dynamic SA or use an existing SA.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.2	Threshold	HAIPEs shall generate a Discovery PROBE message upon receipt of a PT packet destined to a host where the PT or CT IP Addresses of the destination HAIPE fronting the host is not known.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.3	Threshold	HAIPEs shall support statically configured HAIPE PT and CT IP Addresses for a remote prefix.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.4	Threshold	HAIPEs shall accept a Discovery message with a size of up to 576 bytes in length.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.5	Threshold	HAIPEs shall be configurable with a multicast address to which PROBE messages may be sent .	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.6	Threshold	HAIPEs shall join the multicast group which configuration indicates as the Discovery	Legacy Discovery Extension	Terminal	

		group.			
NET.DISC.PD.LEG.7	Threshold	HAIPEs shall support the capability to configure a cryptographic suite (combination of SA data attributes) to use in protecting Discovery messages.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.8	Threshold	HAIPEs shall send all encrypted PROBE messages to the multicast address configured for Discovery.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.9	Threshold	HAIPEs shall create Discovery messages as UDP packets with a destination port of 3623 (decimal).	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.10	Threshold	HAIPEs shall create a UDP Payload of the PROBE message containing the options shown in Table NET.DISC.PD.LEG.10.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.11	Objective	HAIPEs shall include the options as coded in Table NET.DISC.PD.LEG.11 for the PROBE message.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.12	Threshold	HAIPEs shall send a TRYME message upon receipt of a PROBE message containing a Target Host IP Address it fronts.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.13	Threshold	HAIPEs shall code the UDP payload of the TRYME message as shown in Table NET.DISC.PD.LEG.13.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.14	Objective	HAIPEs shall include the options as coded in Table NET.DISC.PD.LEG.14 for the TRYME message.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.15	Threshold	HAIPEs shall code the Vendor Specific Data field to include a 2-octet Vendor ID followed by the Vendor Specific Information.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.16	Threshold	HAIPEs shall code the Vendor Specific Option to be equal to or less than 128 bytes.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.17	Threshold	HAIPEs shall not use information in the Discovery message CT or PT header to validate information in the Discovery message body.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.18	Threshold	HAIPEs shall ignore options not recognized in Discovery messages and continue to process the message.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.19	Threshold	Upon receipt of a TRYME message, HAIPEs shall associate the target host (destination address) with an existing SA or initiate an IKE exchange to create a SA with the HAIPE fronting the target host.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.20	Threshold	The capability to dynamically Discover a remote HAIPE and establish an SA shall be configurable (ON or OFF).	Legacy Discovery Extension	Terminal	

NET.DISC.PD.LEG.21	Threshold	The capability to dynamically Discover a remote HAIPE and establish an SA shall default to ON.	Legacy Discovery Extension	Terminal	
NET.DISC.PD.LEG.22	Threshold	HAIPEs shall pre-configure the deviceVersionTable as follows: dvName is "LegacyDiscovery", and dvVersion is "3.0.0".	Legacy Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.1	Threshold	HAIPEs shall use Routing Information Protocol version 2 (RIPv2) as specified in RFC 2453 on the PT interface to maintain a database of the networks that are locally reachable from the PT IPv4 interface.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.2	Threshold	HAIPEs shall use Routing Information Protocol, next generation (RIPng) as specified in RFC 2080 on the PT interface to maintain a database of the networks that are locally reachable from the PT IPv6 interface.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.3	Threshold	HAIPEs shall allow the manual configuration of a Local Enclave Prefix Table containing the PT prefixes that are locally reachable from the PT interface.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.4	Threshold	HAIPEs shall listen on the PT IPv4 interface for RIPv2 messages as specified in RFC 2453 from a router on the local PT link and use the information to populate the Local Enclave Prefix Table.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.5	Objective	HAIPEs shall listen on the PT IPv4 interface for RIPv2 messages as specified in RFC 2453 from two routers simultaneously on the local PT link and use the information to populate the Local Enclave Prefix Table.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.6	Threshold	HAIPEs shall have the capability to send RIPv2 messages as specified in RFC 2453 on the PT IPv4 interface to indicate the default route.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.7	Threshold	HAIPEs shall listen for RIPng messages as specified in RFC 2080 on the PT IPv6 interface from a router on the local PT link and use the information to populate the Local Enclave Prefix Table.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.8	Objective	HAIPEs shall listen for RIPng messages as specified in RFC 2080 on the PT IPv6 interface from two router simultaneously on the local PT link and use the information to populate the Local Enclave Prefix Table.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.9	Threshold	HAIPEs shall have the capability to send RIPng messages as specified in RFC 2080 on the PT IPv6 interface to indicate the default route.	PT RIP Discovery Extension	Terminal	

NET.DISC.LD.PTRIP.10	Threshold	HAIPEs shall support a RIP update timer on the PT interface for sending RIPv2 messages.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.11	Threshold	HAIPEs shall support a RIP update timer on the PT interface for sending RIPng messages.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.12	Threshold	HAIPEs shall support a RIP route-expiration timeout on the PT interface for RIPv2.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.13	Threshold	HAIPEs shall support a RIP route-expiration timeout on the PT interface for RIPng.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.14	Threshold	HAIPEs shall consider PT Local Enclave Prefix Table entries that are not refreshed with RIPv2 or RIPng updates by the end of the route-expiration time as invalid.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.15	Threshold	HAIPEs shall support a RIP route-flush (garbage collection) timer on the PT interface for RIPv2.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.16	Threshold	HAIPEs shall support a RIP route-flush (garbage collection) timer on the PT interface for RIPng.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.17	Threshold	HAIPEs shall remove PT Local Enclave Prefix table entries that are not refreshed with RIPv2 or RIPng updates by the end of the route-flush time.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.18	Threshold	HAIPEs shall support the ENABLING and DISABLING of listening on the PT interface for RIPv2 messages.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.19	Threshold	HAIPEs shall allow for the configuration of the metric (between 1 and 15) that is to be advertised with the RIPv2 default route.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.20	Threshold	HAIPEs shall support the ENABLING and DISABLING of sending RIPv2 messages indicating the default route.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.21	Threshold	HAIPEs shall support the ENABLING and DISABLING of listening on the PT interface for RIPng messages.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.22	Threshold	HAIPEs shall allow for the configuration of the metric (between 1 and 15) that is to be advertised with the RIPng default route.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.23	Threshold	HAIPEs shall support the ENABLING and DISABLING of sending RIPng messages indicating the default route.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.24	Threshold	HAIPEs shall support simultaneous enabling of RIPv2 and RIPng on the PT interface.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.25	Threshold	HAIPEs shall be configurable to allow default routes learned via RIPv2 or RIPng from the PT router.	PT RIP Discovery Extension	Terminal	

NET.DISC.LD.PTRIP.26	Threshold	HAIPEs shall default the ability to filter (discard) default routes learned via RIPv2 or RIPng from the PT router to ON.	PT RIP Discovery Extension	Terminal	
NET.DISC.LD.PTRIP.27	Threshold	HAIPEs shall pre-configure the deviceVersionTable as follows: dvName is "PTRIPDiscovery", and dvVersion is "3.0.0".	PT RIP Discovery Extension	Terminal	
NET.IP.V4.PDUN.1	Threshold	HAIPEs shall allow a user to enable, on a per SA basis, the transmission of Peer Destination Unreachable Notification messages.	Reachability Extension	Terminal	
NET.IP.V4.PDUN.2	Threshold	Upon receipt of a unicast packet that post-decryption is not in the Local Enclave Prefix Table, HAIPEs shall send a Peer Destination Unreachable Notification (ICMP Host Unreachable (Type 3, Code 1)) message to the peer HAIPE PT IP address, including the destination IP address of the PT packet in the body of the ICMP message.	Reachability Extension	Terminal	
NET.IP.V4.PDUN.3	Objective	Upon receipt of a unicast packet that post-decryption is not permitted by the SPD, HAIPEs shall send a Peer Destination Unreachable Notification (ICMP Port Unreachable (Type 3, Code 3)) message to the Source Host IP Address of the offending packet, including the destination IP address of the PT packet in the body of the ICMP message.	Reachability Extension	Terminal	
NET.IP.V4.PDUN.4	Threshold	Upon receipt of a Peer Destination Unreachable Notification (ICMP Host Unreachable (Type 3, Code 1)) message, HAIPEs shall initiate a new peer discovery process to find a HAIPE fronting the PT destination address contained in the ICMP message payload.	Reachability Extension	Terminal	
NET.IP.V4.PHRD.1	Threshold	HAIPEs shall allow a user to enable, on a per SA basis, the transmission of Peer HAIPE Reachability Detection Messages to the Peer HAIPE PT interface or internal IP address.	Reachability Extension	Terminal	
NET.IP.V4.PHRD.2	Threshold	HAIPEs shall allow a user to configure the rate at which it sends Peer HAIPE Reachability Detection Messages on SAs where Peer HAIPE Reachablity Detection has been enabled.	Reachability Extension	Terminal	
NET.IP.V4.PHRD.3	Threshold	HAIPEs shall default the rate at which Peer HAIPE Reachability Detection Message are transmitted to ZERO. Note: ZERO indicates that a HAIPE will never send Peer HAIPE Reachability Detection messages.	Reachability Extension	Terminal	

NET.IP.V4.PHRD.4	Objective	HAIPEs shall support use of traffic as an indicator of Peer HAIPE Reachability, as described in Section 5.4 of RFC 3706 and not send Peer HAIPE Reachability Detection messages at the configured rate.	Reachability Extension	Terminal	
NET.IP.V4.PHRD.5	Threshold	HAIPEs shall send Peer HAIPE Reachability Detection (PT ICMP Echo Request (Type 8, Code 0)) messages with TTL=2, to the Peer HAIPE PT interface or internal IP address.	Reachability Extension	Terminal	
NET.IP.V4.PHRD.6	Threshold	Upon receipt of a Peer HAIPE Reachability Detection (PT ICMP Echo Request (Type 8, Code 0)) message from a HAIPE in the SAD, the HAIPE shall respond with a Peer HAIPE Reachability Detection (PT ICMP Echo Response (Type 0, Code 0)) message addressed to the PT source address of the ICMP Echo Request.	Reachability Extension	Terminal	
NET.IP.V4.PHRD.7	Threshold	HAIPEs shall mark an SA endpoint as unreachable when the peer HAIPE failed to respond to a sequence of Peer HAIPE Reachability Detection, PT ICMP Echo Requests.	Reachability Extension	Terminal	
NET.IP.V4.PHRD.8	Threshold	HAIPEs shall allow configuration of the number of missed PT ICMP Echo Requests allowed before marking a SA endpoint as unreachable.	Reachability Extension	Terminal	
NET.IP.V4.PHRD.9	Threshold	Upon receipt of a Peer HAIPE Reachability Detection (PT ICMP Echo Response) from a HAIPE in the SAD marked unreachable, HAIPEs shall mark the SA endpoint as reachable.	Reachability Extension	Terminal	
NET.IP.V4.PHRD.10	Objective	Upon receipt of traffic from a HAIPE in the SAD marked unreachable, HAIPEs shall mark the SA endpoint as reachable.	Reachability Extension	Terminal	
NET.IP.V4.PHRD.11	Threshold	HAIPEs shall initiate the Discovery process for the PT destination upon receipt of a packet that will be transmitted to a SA endpoint marked unreachable.	Reachability Extension	Terminal	
NET.IP.V4.PHRD.12	Threshold	HAIPEs shall continue to transmit packets to a SA endpoint marked unreachable, if there is not another reachable SA endpoint associated with the proper set of selectors.	Reachability Extension	Terminal	
NET.IP.V4.PHRD.13	Threshold	If another reachable SA endpoint associated with the proper set of selectors exists for packets previously being sent to an SA endpoint that has been marked unreachable, the HAIPE shall	Reachability Extension	Terminal	

		transmit the packets for the remote PT prefix to the reachable SA endpoint.			
NET.IP.V4.PHRD.14	Threshold	HAIPEs shall pre-configure the deviceVersionTable as follows: dvName is "Reachability" and dvVersion is "3.0.0"	Reachability Extension	Terminal	
NET.IP.V6.PDUN.1	Threshold	HAIPEs shall allow a user to enable, on a per SA basis, the transmission of Peer Destination Unreachable Notification messages.	Reachability Extension	Terminal	
NET.IP.V6.PDUN.2	Threshold	Upon receipt of a unicast packet that post-decryption is not in the Local Enclave Prefix Table, HAIPEs shall send a Peer Destination Unreachable Notification (ICMPv6 Address Unreachable (Type 1, Code 3)) message to the peer HAIPE PT IP address, including the destination IP address of the PT packet in the body of the ICMPv6 message.	Reachability Extension	Terminal	
NET.IP.V6.PDUN.3	Objective	Upon receipt of a unicast packet that post-decryption is not permitted by the SPD, HAIPEs shall send a Peer Destination Unreachable Notification (ICMPv6 Destination Administratively Prohibited (Type 1, Code 1)) message to the Source Host IP Address of the offending packet, including the destination IP address of the PT packet in the body of the ICMP message.	Reachability Extension	Terminal	
NET.IP.V6.PDUN.4	Threshold	Upon receipt of a Peer Destination Unreachable Notification (ICMPv6 Address Unreachable (Type 1, Code 3)) message, HAIPEs shall initiate a new peer discovery process to find a HAIPE fronting the PT destination address contained in the ICMPv6 message payload.	Reachability Extension	Terminal	
NET.IP.V6.PHRD.1	Threshold	HAIPEs shall allow a user to enable, on a per SA basis, the periodic transmission of Peer HAIPE Reachability Detection Messages to the Peer HAIPE PT interface or internal IP address.	Reachability Extension	Terminal	
NET.IP.V6.PHRD.2	Threshold	HAIPEs shall allow a user to configure the rate at which it sends Peer HAIPE Reachability Detection Messages on SAs where Peer HAIPE Reachablity Detection has been enabled.	Reachability Extension	Terminal	
NET.IP.V6.PHRD.3	Threshold	HAIPEs shall default the rate at which Peer HAIPE Reachability Detection Message are transmitted to ZERO. Note: ZERO indicates that a HAIPE will never send Peer HAIPE Reachability	Reachability Extension	Terminal	

		Detection messages.			
NET.IP.V6.PHRD.4	Objective	HAIPEs shall support use of traffic as an indicator of Peer HAIPE Reachability, as described in Section 5.4 of RFC 3706 and not send Peer HAIPE Reachability Detection messages at the configured rate.	Reachability Extension	Terminal	
NET.IP.V6.PHRD.5	Threshold	HAIPEs shall send Peer HAIPE Reachability Detection (PT ICMPv6 Echo Request (Type 128, Code 0)) messages with TTL=2, to the Peer HAIPE PT interface or internal IP address.	Reachability Extension	Terminal	
NET.IP.V6.PHRD.6	Threshold	Upon receipt of a Peer HAIPE Reachability Detection (PT ICMPv6 Echo Request (Type 128, Code 0)) from a HAIPE in the SAD, the HAIPE shall respond with a Peer HAIPE Reachability Detection (PT ICMPv6 Echo Response (Type 129, Code 0)) message addressed to the PT source address of the ICMPv6 Echo Request.	Reachability Extension	Terminal	
NET.IP.V6.PHRD.7	Threshold	HAIPEs shall mark a SA endpoint as unreachable when the peer HAIPE failed to respond to a sequence of Peer HAIPE Reachability Detection, PT ICMPv6 Echo Requests.	Reachability Extension	Terminal	
NET.IP.V6.PHRD.8	Threshold	HAIPEs shall allow configuration of the number of missed PT ICMPv6 Echo Requests allowed before marking a SA endpoint as unreachable.	Reachability Extension	Terminal	
NET.IP.V6.PHRD.9	Threshold	Upon receipt of a Peer HAIPE Reachability Detection (PT ICMPv6 Echo Response) from a HAIPE in the SAD marked unreachable, HAIPEs shall mark the SA endpoint as reachable.	Reachability Extension	Terminal	
NET.IP.V6.PHRD.10	Objective	Upon receipt of traffic from a HAIPE in the SAD marked unreachable, HAIPEs shall mark the SA endpoint as reachable.	Reachability Extension	Terminal	
NET.IP.V6.PHRD.11	Threshold	HAIPEs shall initiate the Discovery process for the PT destination upon receipt of a packet that will be transmitted to a SA endpoint marked unreachable.	Reachability Extension	Terminal	
NET.IP.V6.PHRD.12	Threshold	HAIPEs shall continue to transmit packets to an SA endpoint marked unreachable, if there is not another reachable SA endpoint associated with the proper set of selectors.	Reachability Extension	Terminal	
NET.IP.V6.PHRD.13	Threshold	If another reachable SA endpoint associated with the proper set of selectors exists for packets previously being	Reachability Extension	Terminal	

		sent to an SA endpoint that has been marked unreachable, the HAIPE shall transmit the packets for the remote PT prefix to the reachable SA endpoint.			
TP.ESP.LEG.1	Threshold	HAIPEs shall support transmission and reception of user data in ESP Tunnel Mode as specified in RFC 2406 and as modified herein.	Legacy ESP Extension	Terminal POET ACM	
TP.ESP.LEG.2	Threshold	HAIPEs shall format the ESP packet as specified in Table - TP.ESP.LEG.2.	Legacy ESP Extension	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.ESP.LEG.3	Threshold	HAIPEs shall format the SPI as shown in Table - TP.ESP.LEG.3.	Legacy ESP Extension	Terminal	
TP.ESP.LEG.4	Threshold	HAIPEs shall generate the SPI as detailed in Table - TP.ESP.LEG.4.	Legacy ESP Extension	Terminal POET ACM	SPI is coded using information from the DS100-1 tag.
TP.ESP.LEG.5	Threshold	HAIPEs shall utilize the SPI created at TEK setup through all Deterministic Key Updates until a new SPI is created resulting from a new PPK.	Legacy ESP Extension	Terminal	
TP.ESP.LEG.6	Threshold	HAIPEs shall ignore the Unencrypted ESP Sequence Number value in received packets.	Legacy ESP Extension	Terminal	
TP.ESP.LEG.7	Threshold	HAIPEs shall format the State Variable as shown in Table - TP.ESP.LEG.7.	Legacy ESP Extension	Terminal POET ACM	SV is coded using TEK information (update count) and random data.
TP.ESP.LEG.8	Threshold	HAIPEs shall encrypt all information from the Payload Sequence Number to the Integrity Check Value fields inclusive.	Legacy ESP Extension	Terminal POET ACM	The network interfaces are responsible for preparing the data in such a way that the appropriate data is encrypted correctly.
TP.ESP.LEG.9	Threshold	HAIPEs shall increment the Payload Sequence Number across Deterministic Key Updates.	Legacy ESP Extension	Terminal	.
TP.ESP.LEG.10	Threshold	HAIPEs shall support the Sliding Window approach as specified in Section 3.4.3 of RFC 2406 for the Payload Sequence Number.	Legacy ESP Extension	Terminal	
TP.ESP.LEG.11	Threshold	HAIPEs shall add padding starting from the least significant bit (LSb) of the PT packet.	Legacy ESP Extension	Terminal	
TP.ESP.LEG.12	Threshold	HAIPEs shall not apply Payload Sequence Number processing on packets received on multi-sender SAs.	Legacy ESP Extension	Terminal	
TP.ESP.LEG.13	Threshold	HAIPEs shall pre-configure the deviceVersionTable as follows: dvName is "LegacyESP", and	Legacy ESP Extension	Terminal	

		dvVersion is "3.0.0".			
TP.PE.SAD.MAN.30	Threshold	When performing a Deterministic Update on a TEK, HAIPEs configured to use Legacy ESP shall use the "next" TEK starting at 0200 GMT to protect transmit traffic.	Legacy ESP Extension	Terminal POET ACM	
TP.PE.SAD.MAN.31	Threshold	When performing a Deterministic Update on a TEK, HAIPEs configured to use Legacy ESP shall accept received traffic encrypted with the "next" TEK starting at 0105 GMT.	Legacy ESP Extension	Terminal POET ACM	
TP.PE.SAD.MAN.32	Threshold	When performing a Deterministic Update on a TEK, HAIPEs configured to use Legacy ESP shall continue to accept received traffic encrypted with the "current" TEK until 0255 GMT.	Legacy ESP Extension	Terminal POET ACM	
TP.PE.SAD.MAN.33	Threshold	When performing a Deterministic Update on a TEK configured to use Legacy ESP, at 0255 GMT, HAIPEs shall delete the "current" TEK.	Legacy ESP Extension	Terminal POET ACM	.
TP.ESP.ESPV3.TRPMD.1	Threshold	HAIPEs shall perform processing of ESPv3 Transport Mode packets as specified in the HAIPE IS v3.0.0 Traffic Protection Core and as modified herein.	ESPv3 Transport Mode Extension	Terminal POET ACM	The requirements in this document are satisfied by both the POET ACM and the remaining components of the terminal.
TP.ESP.ESPV3.TRPMD.2	Threshold	When configured for Transport Mode, HAIPEs shall format the ESPv3 transport mode packet as specified in Table - TP.ESP.ESPv3.TRPMD.2.	ESPv3 Transport Mode Extension	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.ESP.ESPV3.TRPMD.3	Threshold	HAIPEs shall encrypt all information from the Transport Layer Datagram to the Next Header fields inclusive.	ESPv3 Transport Mode Extension	Terminal POET ACM	The network interfaces are responsible for preparing the data in such a way that the appropriate data is encrypted correctly.
TP.ESP.ESPV3.TRPMD.4	Threshold	HAIPEs shall not apply TFC padding to Transport Mode ESPv3 packets.	ESPv3 Transport Mode Extension	Terminal	
TP.ESP.ESPV3.TRPMD.5	Objective	HAIPEs shall support a mode of encapsulation in which Transport Mode packets are sent without the addition of the ICV field in the ESPv3 packet as specified in Table - TP.ESP.ESPV3.TRPMD.5.	ESPv3 Transport Mode Extension	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.ESP.ESPV3.TRPMD.6	Threshold	HAIPEs shall pre-configure the deviceVersionTable as follows: dvName is "ESPv3TransportMode", and dvVersion is "3.0.0".	ESPv3 Transport Mode Extension	Terminal	
TP.PE.SAD.MAN.23	Threshold	HAIPEs shall provide the capability to configure the	ESPv3 Transport Mode Extension	Terminal	

		ESPv3 transport mode (with ICV or without ICV) for multicast Security Associations.			
TP.PE.SAD.MAN.24	Threshold	HAIPEs shall provide the capability to configure the ESPv3 transport mode (with ICV or without ICV) for manual Security Associations.	ESPv3 Transport Mode Extension	Terminal	
TP.PE.SAD.MAN.25	Threshold	HAIPEs shall provide the capability to configure ESPv3 transport mode with an Integrity Check Value of 96 bits.	ESPv3 Transport Mode Extension	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.PE.SAD.MAN.26	Threshold	HAIPEs shall provide the capability to configure ESPv3 transport mode with an Integrity Check Value of 0 bits.	ESPv3 Transport Mode Extension	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.PE.SAD.MAN.27	Objective	HAIPEs shall provide the capability to configure ESPv3 transport mode with an Integrity Check Value of 32 bits.	ESPv3 Transport Mode Extension	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.PE.SAD.MAN.28	Objective	HAIPEs shall provide the capability to configure ESPv3 transport mode with an Integrity Check Value of 64 bits.	ESPv3 Transport Mode Extension	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.PE.SAD.MAN.29	Objective	HAIPEs shall provide the capability to configure ESPv3 transport mode with an Integrity Check Value of 128 bits.	ESPv3 Transport Mode Extension	Terminal POET ACM	Assuming ICV is computed and stored by the POET ACM.
TP.PE.SPD.13	Threshold	HAIPEs shall allow for the use of the OPAQUE value with Security Policy Database Selectors.	Granular Selector Capability Extension	Terminal	
TP.PE.SPD.14	Threshold	HAIPEs shall perform Granular Selector processing as specified in the HAIPE IS 3.0.0 Traffic Protection Core and as modified herein.	Granular Selector Capability Extension	Terminal	
TP.PE.SPD.15	Threshold	HAIPEs shall allow for the use of an IPv4 Destination IP Address range as a Selector in the Security Policy Database.	Granular Selector Capability Extension	Terminal	
TP.PE.SPD.16	Threshold	HAIPEs shall allow for the use of an IPv6 Destination IP Address range as a Selector in the Security Policy Database.	Granular Selector Capability Extension	Terminal	
TP.PE.SPD.17	Threshold	HAIPEs shall allow for the use of a Source IPv4 Address range as a Selector in the Security Policy Database.	Granular Selector Capability Extension	Terminal	
TP.PE.SPD.18	Threshold	HAIPEs shall allow for the use of a Source IPv6 Address range as a Selector in the Security Policy Database.	Granular Selector Capability Extension	Terminal	
TP.PE.SPD.19	Threshold	HAIPEs shall allow for the use of a Transport Layer protocol as a Selector in the Security Policy Database.	Granular Selector Capability Extension	Terminal	
TP.PE.SPD.20	Threshold	HAIPEs shall allow for the use of a Source/Destination Port range as a Selector in the Security Policy Database.	Granular Selector Capability Extension	Terminal	
TP.PE.SPD.21	Objective	HAIPEs shall be capable of using DSCP classifiers for	Granular Selector Capability	Terminal	

		Security Associations.	Extension		
TP.PE.SPD.22	Threshold	HAIPEs shall pre-configure the deviceVersionTable as follows: dvName is "GranularSelector" and dvVersion is "3.0.0"	Granular Selector Capability Extension	Terminal	

C.

D. APPENDIX C: LEF IS 2.0 REQUIREMENT ALLOCATION

(U) The following table presents the allocation of the LEF IS 2.0 requirements to the POET ACM, portions of the terminal other than the ACM, or to both. Notes are provided as necessary to provide additional detail or justification. This table is provided for reference purposed only

Table C-1 (U) LEF IS REQUIREMENT ALLOCATION

Requirement ID	Requirement Type	Requirement Text	PLATFORM Allocation	Notes
2LEF4020	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF4020) support Non-conditioned Baseband Data format for all cryptographic modes.	Terminal	
2LEF4040	Threshold	(U//FOUO) The transmitting LEF ECU Â“MUSTÂ” (2LEF4040) provide data on the positive edge of the data clock	Terminal	
2LEF4060	Threshold	(U//FOUO) The receiving LEF ECU Â“MUSTÂ” (2LEF4060) sample data on the negative edge of the data clock.	Terminal	
2LEF4080	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF4080) support Synchronous Data Mode for all cryptographic modes.	Terminal	
2LEF4100	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF4100) support Redundant Synchronization Mode for all cryptographic modes.	Terminal	
2LEF4120	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF4120) support Full-Duplex Communication Mode for all cryptographic modes.	Terminal POET ACM	Duplex, data-rate information directly impact cryptographic engine trade-offs.
2LEF4140	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF4140) support Data Rates between 9.6Kbps Â– 45Mbps for all cryptographic modes.	Terminal POET ACM	Duplex, data-rate information directly impact cryptographic engine trade-offs.
2LEF4160	Objective	(U//FOUO) The LEF ECU Â“MAYÂ” (2LEF4160) support a data rate that is lower than or greater than the required data rates.	Terminal POET ACM	Duplex, data-rate information directly impact cryptographic engine trade-offs.
2LEF4180	Objective	(U//FOUO) The LEF ECU Â“MAYÂ” (2LEF4180) support a Broadcast (one-way) Communication Mode.	Terminal	
2LEF4220	Objective	(U//FOUO) The LEF ECU Â“MAYÂ” (2LEF4220) implement additional modes of operation.	Terminal POET ACM	Additional modes may impact POET ACM.
2LEF4200	Threshold	When operating in Broadcast Mode, the LEF ECU Â“MUSTÂ” (2LEF4200) utilize Symmetric Keys.	Terminal POET ACM	
2LEF5020	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5020) support the MEDLEY algorithm in Counter Mode as specified in the Classified Appendix and MEDLEY Implementation Standard (NSA Doc. #0N679197).	POET ACM	
2LEF5040	Objective	(U//FOUO) The LEF ECU Â“MAYÂ” (2LEF5040) implement additional encryption algorithms with the approval of NSA.	Terminal POET ACM	
2LEF5060	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5060) be capable of using the NSA authentication algorithm in conjunction with MEDLEY. Refer to the LEF Classified Appendix for implementation details.	POET ACM	TBS in the classified appendix to the LEF IS.
2LEF5080	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5080) support the enabling/disabling of cryptographic authentication for MEDLEY.	Terminal POET ACM	TBS in the classified appendix to the LEF IS.

2LEF5100	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5100) default to the cryptographic authentication enabled selection.	Terminal	
2LEF5120	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5120) support the NIST AES (FIPS 197) algorithm in Counter Mode. Refer to the LEF Classified Appendix for implementation details.	POET ACM	TBS in the classified appendix to the LEF IS.
2LEF5140	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5140) use the NSA authentication algorithm (NSA Doc. # TBS) in conjunction with AES. Refer to the LEF Classified Appendix for implementation details.	POET ACM	TBS in the classified appendix to the LEF IS.
2LEF5160	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5160) support the enabling/disabling of cryptographic authentication for AES.	Terminal POET ACM	TBS in the classified appendix to the LEF IS.
2LEF5180	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5180) default to the cryptographic authentication enabled selection.	Terminal	
2LEF5200	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5200) incorporate the following cryptographic redundant synchronization sequence (Table 5-3) to establish link synchronization.	Terminal POET ACM	TBS in the classified appendix to the LEF IS.
2LEF5220	Threshold	(U//FOUO) An LEF ECU that detects incoming idle bits after achieving two-way cryptographic synchronization Â“MUSTÂ” (2LEF5220) cease encryption and decryption of traffic data.	Terminal	
2LEF5240	Threshold	(U//FOUO) The initiating LEF ECU Â“MUSTÂ” (2LEF5240) send idle bits until the responding ECU returns idle bits.	Terminal	
2LEF5260	Threshold	(U//FOUO) The responding LEF ECU Â“MUSTÂ” (2LEF5260) respond to idle bits by sending 256 idle bits.	Terminal	
2LEF5280	Threshold	(U//FOUO) The initiating LEF ECU Â“MUSTÂ” (2LEF5280) send a minimum of 256 additional idle bits after the detection of idle bits from the responding ECU.	Terminal	
2LEF5300	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5300) implement an idle detection scheme that has at least 99.99% probability of detection at a Bit Error Rate (BER) of 10-3 within 256 bits.	Terminal	
2LEF5320	Threshold	(U//FOUO) The idle detection scheme Â“MUSTÂ” (2LEF5320) have no more than one false detection in 1012 random ciphertext bits. Analysis or simulation may be used to verify that the detection scheme can meet these requirements.	Terminal	
2LEF5340	Threshold	(U//FOUO) The LEF ECU Â“MUST NOTÂ” (2LEF5340) encrypt the frame marker.	Terminal	
2LEF5360	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5360) code the frame marker as specified in Table 5-4. The frame marker consists of 465 binary bits, divided into 31 Phi coded groups, each 15 bits long. A Phi code is either 000010100110111, representing a logical zero, or 111101011001000, representing a one. The groups of Phi code represent a 31-bit sequence of 0000000000000000000000000011101. The last 5 bits of this sequence is known as a Barker code, commonly used for frame synchronization in digital communication systems.	Terminal	
2LEF5380	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5380) implement a framing detection scheme that has at least 99.99% probability of detection at a Bit Error Rate (BER) of 10-3. This limit includes the probability of missing it plus any false detection prior to the last bit. Analysis or simulation may be used to verify that the detection scheme can meet this requirement.	Terminal	
2LEF5400	Threshold	(U//FOUO) In Full-Duplex Mode, if an LEF ECU fails to detect the frame marker and has sent the frame marker, it Â“MUSTÂ” (2LEF5400) begin sending idle bits after reaching the timeout period.	Terminal	
2LEF5420	Threshold	The timeout period Â“MUSTÂ” (2LEF5420) equal thirty (30) seconds to account for the worst-case round trip transmission delay.	Terminal	
2LEF5440	Objective	Other timeout values Â“MAYÂ” (2LEF5440) be implemented as configurable settings.	Terminal	
2LEF5460	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5460) support the Key Generator (KG) operating in counter mode. The local receive and remote transmit LEF ECU counters are synchronized using the Initialization Vector (IV). This process establishes identical initial conditions for the receiving KG of the local unit and the transmitting KG of the remote unit. Refer	Terminal POET ACM	TBS in the classified appendix to the LEF IS.

		to the LEF Classified Appendix for details on the IV.		
2LEF5480	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5480) transmit the IV immediately after completing transmission of frame marker.	Terminal	
2LEF5500	Threshold	(U//FOUO) The LEF ECU Â“MUST NOTÂ” (2LEF5500) encrypt the IV.	Terminal	
2LEF5520	Threshold	(U//FOUO) The LEF ECU initiating synchronization "MUST" (2LEF5520) send a minimum of 256 idle bits.	Terminal	
2LEF5540	Threshold	(U//FOUO) After idle bit transmission, the initiating LEF ECU "MUST" (2LEF5540) send the Phi encoded 465-bit frame marker pattern in accordance with Section 5.3.2.	Terminal	
2LEF5560	Threshold	(U//FOUO) After frame marker transmission, the initiating LEF ECU "MUST" (2LEF5560) send the IV Bits followed by the Sync Check Bits. Refer to the LEF Classified Appendix for details on the IV and Sync Check.	Terminal POET ACM	TBS in the classified appendix to the LEF IS.
2LEF5580	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5580) support the filling and use of one EFF vector set.	POET ACM	
2LEF5600	Objective	(U//FOUO) The LEF ECU Â“MAYÂ” (2LEF5600) support the filling and use of multiple EFF vector sets.	POET ACM	
2LEF5620	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5620) support the filling and use of one symmetric key. Note: OTAR requires temporary storage of the next TEK.	POET ACM	
2LEF5640	Objective	(U//FOUO) The LEF ECU Â“MAYÂ” (2LEF5640) support the filling and use of multiple symmetric keys.	POET ACM	
2LEF5660	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5660) support EKMS 322 requirements for EFF.	POET ACM	
2LEF5680	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5680) be able to load and assign for use any PPK with valid DS100 Tag Parity, Key Parity and Parity Prime fields IAW the Classified Appendix (PPK format).	Terminal POET ACM	
2LEF5700	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5700) provide fill capability for EFF vector sets from a DTD (AN/CYZ-10) or other approved device using the DS-101/DS-102 protocols in accordance with EKMS 308 requirements.	POET ACM	
2LEF5720	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5720) provide fill capability with symmetric keys from a DTD (AN/CYZ-10) or other approved device using the DS-101/DS-102 protocols in accordance with EKMS 308 requirements.	POET ACM	
2LEF5740	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5740) provide fill capability for both EFF vector sets and symmetric keys in black form (using Benign Fill techniques as specified in EKMS 217).	POET ACM	
2LEF5760	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF5760) incorporate a standard 6-pin fill connector for interfacing to key material devices. The required fill connector is specified by NSA drawing number 0N241774, Connector, Plug, Audio, 6-Contact, and must be placed on the front panel assembly and positioned for ease of access for the mating connector.	POET ACM	
2LEF6020	Threshold	(U//FOUO) In-Band Commands Â“MUST NOTÂ” (2LEF6020) be sourced from the plain text interface. In-Band Commands are intended to be sourced and consumed entirely within the LEF ECU communication devices.	Terminal	In-band commands are sourced by the RED processor, encrypted, and transmitted by the BLACK processor.
2LEF6040	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF6040) align in-band commands on a crypto block boundary.	Terminal	In-band commands are sourced by the RED processor, encrypted, and transmitted by the BLACK processor.
2LEF6060	Threshold	(U//FOUO) The LEF ECU Â“MUST NOTÂ” (2LEF6060) send in-band commands unencrypted in the ciphertext data stream.	Terminal	In-band commands are sourced by the RED processor, encrypted, and transmitted by the BLACK processor.
2LEF7010	Objective	(U//FOUO) The LEF ECU Â“MAYÂ” (2LEF7010) be capable of performing OTAR using broadcast mode.	Terminal	More specific requirements apply to the POET ACM.
2LEF7020	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF7020) be capable of performing OTAR in Full-Duplex mode.	Terminal	More specific requirements apply to the POET ACM.

2LEF7030	Threshold	(U//FOUO) The receiving LEF ECU Â“MUSTÂ” (2LEF7030) be capable of performing OTAR in a mode that does not require user intervention at the receiving LEF ECU (i.e., Non-Cooperative Mode).	Terminal	More specific requirements apply to the POET ACM.
2LEF7040	Threshold	(U//FOUO) The initiating LEF ECU Â“MUSTÂ” (2LEF7040) perform cryptographic resynchronization using the current key as the traffic key (since the initiating LEF ECU operator will manually select the next key).	Terminal POET ACM	
2LEF7050	Threshold	(U//FOUO) The responding LEF ECU(s) Â“MUSTÂ” (2LEF7050) perform cryptographic resynchronization using the next key (resulting from OTAR) as the traffic key (no receiving LEF ECU operator action required).	Terminal POET ACM	
2LEF7060	Threshold	(U//FOUO) The LEF ECU "MUST" (2LEF7060) support temporary storage of key material that is used for subsequent OTAR/OTAT transmission. This applies to ECUs initiating OTAR inband commands.	Terminal POET ACM	
2LEF7070	Threshold	(U//FOUO) The LEF ECU "MUST" (2LEF7070) support manual selection, via HMI, of the transmitted OTAR key for use as the new traffic key. This applies to ECUs initiating OTAR inband commands.	Terminal POET ACM	
2LEF7080	Threshold	(U//FOUO) In Full-Duplex mode, the initiating ECU Â“MUSTÂ” (2LEF7080) continuously send OTAR_INIT messages until an OTAR_RDY message is received.	Terminal	
2LEF7090	Threshold	(U//FOUO) A single OTAR_INFO message Â“MUSTÂ” (2LEF7090) then be sent followed by continuous OTAR_ACK messages until an OTAR_ACK or OTAR_NACK message is received.	Terminal POET ACM	OTAR_INFO message contains the (wrapped) new key.
2LEF7100	Threshold	(U//FOUO) Upon receiving an OTAR_ACK message, the initiating ECU Â“MUSTÂ” (2LEF7100) proceed to synchronize with the current TEK (the latest TEK used for traffic), not the TEK transmitted in the OTAR_INFO message.	Terminal POET ACM	
2LEF7110	Threshold	(U//FOUO) Upon receiving an OTAR_NACK message, the initiating ECU Â“MUSTÂ” (2LEF7110) retry sending the OTAR_INFO message until receiving OTAR_ACK or the same OTAR_INFO message has been sent a total of three times.	Terminal POET ACM	OTAR_INFO message contains the (wrapped) new key.
2LEF7120	Threshold	(U//FOUO) In Full-Duplex mode and upon receiving an OTAR_INIT message, the responding ECU Â“MUSTÂ” (2LEF7120) continuously send OTAR_RDY messages until an OTAR_INFO message is received.	Terminal	
2LEF7130	Threshold	(U//FOUO) Upon validation of the OTAR_INFO message, the responding ECU Â“MUSTÂ” (2LEF7130) use the unwrapped TEK from the OTAR_INFO message as the next TEK and Â“MUSTÂ” (2LEF7140) continuously send OTAR_ACK messages until idle bits are received from the initiating ECU.	Terminal POET ACM	OTAR_INFO message contains the (wrapped) new key.
2LEF7150	Threshold	(U//FOUO) If the OTAR_INFO message is invalid, the responding ECU Â“MUSTÂ” (2LEF7150) continuously send OTAR_NACK messages until idle bits are received from the initiating ECU or an OTAR_INIT message is received in order to retry the OTAR procedure as detailed in the corresponding state diagram.	Terminal	
2LEF7160	Threshold	(U//FOUO) In Broadcast mode, the initiating ECU Â“MUSTÂ” (2LEF7160) send eight (8) OTAR_INIT messages.	Terminal	
2LEF7170	Threshold	(U//FOUO) After the OTAR_INIT messages, the initiating ECU Â“MUSTÂ” (2LEF7170) send one (1) OTAR_INFO message.	Terminal POET ACM	OTAR_INFO message contains the (wrapped) new key.
2LEF7180	Threshold	(U//FOUO) After the OTAR_INFO message, the initiating ECU Â“MUSTÂ” (2LEF7180) continuously send OTAR_ACK messages for one (1) second.	Terminal	
2LEF7190	Threshold	(U//FOUO) If there are more wrapped TEKs to transmit, the initiating ECU Â“MUSTÂ” (2LEF7190) to repeat the sequence of OTAR_INIT and OTAR_INFO until all TEKs have been transmitted.	Terminal POET ACM	OTAR_INFO message contains the (wrapped) new key.
2LEF7200	Threshold	(U//FOUO) If there are no more wrapped TEKs to transmit, the initiating ECU Â“MUSTÂ” (2LEF7200) initiate resynchronization with the current TEK (the latest TEK used for traffic), not the TEK transmitted in the OTAR_INFO message.	Terminal POET ACM	

2LEF7210	Threshold	(U//FOUO) In Broadcast mode and upon receiving an OTAR_INIT message, the receiving ECU Â“MUSTÂ” (2LEF7210) wait to receive one OTAR_INFO message.	Terminal	
2LEF7220	Threshold	(U//FOUO) Upon validation of the OTAR_INFO message, the receiving ECU Â“MUSTÂ” (2LEF7220) load the unwrapped TEK from the OTAR_INFO message as the current TEK and wait for another OTAR_INFO message or idle bits indicating resynchronization.	Terminal POET ACM	OTAR_INFO message contains the (wrapped) new key.
2LEF7230	Threshold	(U//FOUO) If the OTAR_INFO message is invalid (i.e., the TEK was wrapped with a different KEK and was not intended for this receiving ECU), the receiving ECU Â“MUSTÂ” (2LEF7230) wait for another OTAR_INFO message or idle bits indicating resynchronization.	Terminal	
2LEF7240	Threshold	(U//FOUO) The initiating LEF ECU Â“MUSTÂ” (2LEF7240) bypass cryptographic authentication for all OTAR messages commencing upon the block boundary in which OTAR_INIT is transmitted.	Terminal POET ACM	Bypass is a function of the POET ACM.
2LEF7250	Threshold	(U//FOUO) Upon completion of the OTAR (successful or not), the LEF ECU Â“MUSTÂ” (2LEF7250) resume authentication if enabled, at the beginning of the very next encrypted block.	Terminal POET ACM	
2LEF7260	Threshold	(U//FOUO) The receiving LEF ECU Â“MUSTÂ” (2LEF7260) be capable of processing unauthenticated OTAR commands/data messages.	Terminal POET ACM	
2LEF7270	Threshold	(U//FOUO) In Full-Duplex Mode, the initiating LEF ECU Â“MUSTÂ” (2LEF7270) perform additional OTAR attempts (to a maximum of 3) if the previous attempt has failed (i.e., OTAR_NACK received).	Terminal	
2LEF7280	Threshold	(U//FOUO) In Full-Duplex Mode, the initiating LEF ECU Â“MUSTÂ” (2LEF7280) stop the OTAR attempts, indicate the error on the HMI and resynchronize if the OTAR has been attempted three (3) times without success.	Terminal	More specific requirements apply to the POET ACM.
2LEF7290	Threshold	(U//FOUO) To initiate an OTAR the initiating LEF ECU Â“MUSTÂ” (2LEF7290) send the OTAR_INIT message to the receiver.	Terminal	
2LEF7300	Threshold	(U//FOUO) In Broadcast Mode, the initiating LEF ECU Â“MUSTÂ” (2LEF7300) transmit the OTAR_INIT message a minimum of eight (8) times to ensure an acceptable probability of detection at the receiver in a 10-3 Bit Error Rate (BER) environment.	Terminal	
2LEF7310	Threshold	(U//FOUO) In Full-Duplex Mode, the initiating LEF ECU Â“MUSTÂ” (2LEF7310) transmit the OTAR_INIT message until an OTAR_RDY message is received from the responding LEF ECU.	Terminal	
2LEF7320	Threshold	(U//FOUO) In Full-Duplex Mode, the receiving LEF ECU Â“MUSTÂ” (2LEF7320) cease the transmission of traffic and Â“MUSTÂ” (2LEF7330) send the OTAR_RDY message to the initiating LEF ECU upon receipt of an OTAR_INIT message.	Terminal	
2LEF7340	Threshold	(U//FOUO) In Full-Duplex Mode, the responding LEF ECU Â“MUSTÂ” (2LEF7340) repeat the OTAR_RDY message until OTAR_ACK is received.	Terminal	
2LEF7350	Threshold	(U//FOUO) In Full-Duplex Mode, the initiating LEF ECU Â“MUSTÂ” (2LEF7350) send the OTAR_INFO message in response to receipt of an OTAR_RDY message.	Terminal POET ACM	OTAR_INFO message contains the (wrapped) new key.
2LEF7360	Threshold	(U//FOUO) In Broadcast Mode the initiating LEF ECU Â“MUSTÂ” (2LEF7360) send the OTAR_INFO message immediately after sending the eight (8) OTAR_INIT messages.	Terminal POET ACM	OTAR_INFO message contains the (wrapped) new key.
2LEF7370	Threshold	(U//FOUO) The initiating LEF ECU Â“MUSTÂ” (2LEF7370) wrap the OTAR_INFO Key Data using ACCORDION 3.0 with the pre-placed Key Encryption Key (KEK).	Terminal POET ACM	OTAR_INFO message contains the (wrapped) new key.
2LEF7380	Threshold	(U//FOUO) The initiating LEF ECU Â“MUSTÂ” (2LEF7380) transmit the OTAR_INFO message MSB to LSB (i.e., Algorithm field first, CRC field last).	Terminal	More specific requirements apply to the POET ACM.
2LEF7390	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF7390) calculate the 32-bit CRC using a linear recursive sequence generator, with a polynomial equation as defined in EKMS 322, Appendix A.	Terminal	
2LEF7400	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF7400) process the OTAR_INFO message MSb-LSb during the CRC calculation.	Terminal	

2LEF7410	Threshold	(U//FOUO) The initiating LEF ECU Â“MUSTÂ” (2LEF7410) code the OTAR_INFO message using an (8,4) Hamming.	Terminal	
2LEF7420	Threshold	(U//FOUO) In Full-Duplex mode, the responding LEF ECU Â“MUSTÂ” (2LEF7420) transmit the OTAR_ACK message upon successful processing of the OTAR_INFO message.	Terminal	
2LEF7430	Threshold	(U//FOUO) In Broadcast mode, after sending the OTAR_INFO message, the initiating LEF ECU Â“MUSTÂ” (2LEF7430) transmit OTAR_ACK messages for a minimum of one second. This allows time for the responder to process the OTAR_INFO message in preparation for synchronization or another OTAR sequence (in a netted situation in which the message validation failed).	Terminal	
2LEF7440	Threshold	(U//FOUO) In Full-Duplex mode, upon receipt of OTAR_ACK, the initiating LEF ECU Â“MUSTÂ” (2LEF7440) either transmit the next OTAR_INIT command or, if there is no more key traffic, perform a cryptographic resynchronization with the responder.	Terminal	
2LEF7450	Threshold	(U//FOUO) In Full-Duplex mode, the receiving LEF ECU Â“MUSTÂ” (2LEF7450) transmit the OTAR_NACK message upon unsuccessful processing of the OTAR_INFO message.	Terminal	
2LEF7460	Threshold	(U//FOUO) The receiving LEF ECU Â“MUSTÂ” (2LEF7460) continue to repeat the transmission of OTAR_NACK messages until a resynchronization or an OTAR_INIT message is received.	Terminal	
2LEF7470	Threshold	(U//FOUO) The responding LEF ECU Â“MUST NOTÂ” (2LEF7470) initiate another OTAR transmission during an active OTAR message sequence.	Terminal	
2LEF7480	Threshold	(U//FOUO) If the initiating LEF ECU receives an OTAR_INIT message from the responder of an active OTAR exchange, the initiating LEF ECU Â“MUSTÂ” (2LEF7480) abort the OTAR exchange and initiate resynchronization using the current key.	Terminal	
2LEF7490	Threshold	(U//FOUO) The responding LEF ECU Â“MUSTÂ” (2LEF7490) route the incoming key to the attached key fill device when connected to a key fill device in receive variable mode.	Terminal POET ACM	
2LEF7500	Threshold	(U//FOUO) After sending OTAR_ACK the initiating LEF ECU Â“MUSTÂ” (2LEF7500) either transmit the next OTAR_INIT command, or send idle bits to initiate synchronization.	Terminal	
2LEF7510	Threshold	(U//FOUO) Upon completion of OTAT processing the responding LEF ECU Â“MUSTÂ” (2LEF7510) resynchronize on the current TEK.	Terminal POET ACM	
2LEF8020	Threshold	(U//FOUO) The LEF ECU "MUST NOT" (2LEF8020) pass traffic to the plaintext interface until new key is calculated and an EFF_ACK is received.	Terminal POET ACM	
2LEF8040	Threshold	(U//FOUO) The LEF ECU "MUST NOT" (2LEF8040) perform a key update operation on an EFF generated key.	Terminal POET ACM	
2LEF8060	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF8060) provide for a bootstrap variable capability in order to support the initial EFF exchange from a Cold Start condition.	Terminal POET ACM	TBS in the classified appendix to the LEF IS.
2LEF8080	Threshold	(U//FOUO) The LEF ECU "MUST" (2LEF8080) perform an EFF exchange immediately after a cold start using the EFF bootstrap variable.	Terminal POET ACM	
2LEF8100	Threshold	(U//FOUO) The initiating LEF ECU Â“MUSTÂ” (2LEF8100) bypass authentication during all EFF messages commencing upon the block boundary in which EFF_INIT is transmitted.	Terminal POET ACM	Assuming bypass is a function of the POET ACM.
2LEF8120	Threshold	(U//FOUO) The initiating LEF ECU Â“MUSTÂ” (2LEF8120) resume authentication, if enabled, after the EFF protocol is complete (successful or not) at the beginning of the next encrypted block of data.	Terminal POET ACM	
2LEF8140	Threshold	(U//FOUO) The responding LEF ECU Â“MUSTÂ” (2LEF8140) be capable of processing unauthenticated EFF commands/data.	Terminal POET ACM	
2LEF8160	Threshold	(U//FOUO) The responding LEF ECU Â“MUSTÂ” (2LEF8160) bypass authentication on all EFF messages sent to the initiating LEF ECU.	Terminal POET ACM	Assuming bypass is a function of the POET ACM.

2LEF8180	Threshold	(U//FOUO) The responding LEF ECU Â“MUSTÂ” (2LEF8180) resume authentication, if enabled, after the EFF protocol is complete (successful or not) at the beginning of the next encrypted block of data.	Terminal POET ACM	
2LEF8200	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF8200) format EFF messages as shown in Table 8-1.	Terminal	
2LEF8220	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF8220) perform EFF key calculation in accordance with EKMS 322 and the LEF Classified Appendix.	POET ACM	TBS in the classified appendix to the LEF IS.
2LEF8240	Threshold	(U//FOUO) To initiate an EFF exchange, the initiating LEF ECU Â“MUSTÂ” (2LEF8240) send the EFF_INIT message to the responder.	Terminal	
2LEF8260	Threshold	(U//FOUO) The initiating LEF ECU Â“MUSTÂ” (2LEF8260) continuously transmit the EFF_INIT code until the responding LEF ECU sends the EFF_INFO message.	Terminal	
2LEF8280	Threshold	(U//FOUO) The responding LEF ECU Â“MUSTÂ” (2LEF8280) send the EFF_INFO message upon receipt of the EFF_INIT.	Terminal POET ACM	
2LEF8300	Threshold	(U//FOUO) Upon receipt of a valid EFF_INFO header, both the initiating and responding LEF ECUs Â“MUSTÂ” (2LEF8300) continuously send EFF_INFO + EFF_MSG until receipt of a valid EF F_ACK + EFF_MSG sequence.	Terminal POET ACM	
2LEF8320	Threshold	(U//FOUO) An LEF ECU Â“MUSTÂ” (2LEF8320) verify the EFF_INFO + EFF_MSG by validating the CRC.	Terminal	
2LEF8340	Threshold	(U//FOUO) After receipt of a valid EFF_INFO message, the LEF ECU Â“MUSTÂ” (2LEF8340) transmit EFF_ACK + EFF_MSG, an indication of successful reception of the EFF_MSG.	Terminal POET ACM	
2LEF8360	Threshold	(U//FOUO) The initiator/responder, upon receiving EFF_ACK + EFF_MSG and successfully calculating a key, Â“MUSTÂ” (2LEF8360) indicate success with the transmission of a minimum of (256) 64-bit EFF_ACK messages, or until receiving idle bits indicating a resynchronization.	Terminal	
2LEF8380	Threshold	(U//FOUO) The LEF ECU, upon receiving EFF_ACK + EFF_MSG and successfully calculating a new key, Â“MUSTÂ” (2LEF8380) use the new EFF TEK for the subsequent resynchronization. (Note: This ensures that the current key will not be re-used during recovery from a data loss condition or a reset during the EFF sequence).	Terminal POET ACM	
2LEF8400	Threshold	(U//FOUO) The LEF ECU "MUST" (2LEF8400) cease encrypting traffic if there is a failure during an EFF exchange.	Terminal	
2LEF8420	Threshold	(U//FOUO) The LEF ECU "MUST" (2LEF8420) perform a resynchronization if idles are seen before EDAC during an EFF exchange.	Terminal	
2LEF8440	Threshold	(U//FOUO) Prior to reaching the Â“EDAC & Check MSG CRCÂ” state, the LEF ECU "MUST" (2LEF8440) use the current TEK for traffic for any failure (i.e., power failure or link failure).	Terminal POET ACM	
2LEF8460	Threshold	(U//FOUO) After completing the Â“EDAC & Check MSG CRCÂ” processing, if there is a failure during an EFF exchange, the LEF ECU "MUST" (2LEF8460) cease encrypting traffic, and enter a condition from which a Cold Start is required.	Terminal POET ACM	
2LEF8480	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF8480) indicate the failure of the EFF exchange processing via an HMI indicator.	Terminal	
2LEF9020	Threshold	(U//FOUO) A key update Â“MUSTÂ” (2LEF9020) be performed on the current traffic key using the ACCORDION 3.0 algorithm.	Terminal POET ACM	
2LEF9040	Threshold	(U//FOUO) A key update Â“MUST NOTÂ” (2LEF9040) be performed using any EFF derived traffic key.	Terminal	
2LEF9060	Threshold	(U//FOUO) To initiate a Key Update, the initiating LEF ECU Â“MUSTÂ” (2LEF9060) send the UPDATE_INIT message.	Terminal	
2LEF9080	Threshold	(U//FOUO) In Full-Duplex mode, a Key Update initiator Â“MUSTÂ” (2LEF9080) send the UPDATE_INIT message continuously until the responder replies with an UPDATE_ACK or an UPDATE_INIT message. Note: The UPDATE_INIT response may occur in the rare circumstance that both units attempt to initiate a Key Update	Terminal	

		simultaneously.		
2LEF9100	Threshold	(U//FOUO) In Full-Duplex mode, a Key Update responder Â“MUSTÂ” (2LEF9100) send the UPDATE_ACK message continuously upon detection of a valid UPDATE_INIT message until a successful detection of idle bits.	Terminal	
2LEF9120	Threshold	(U//FOUO) In Broadcast mode, the UPDATE_INIT message Â“MUSTÂ” (2LEF9120) be repeated eight (8) times in order to be properly detected by a receiving LEF ECU and to provide enough time to load the updated key.	Terminal	
2LEF9140	Threshold	(U//FOUO) In Broadcast mode, the receiving LEF ECU Â“MUSTÂ” (2LEF9140) use the updated key as the traffic key upon detection of a valid UPDATE_INIT message followed by idle bits.	Terminal POET ACM	
2LEF9160	Threshold	(U//FOUO) In Full-Duplex mode, the responding LEF ECU, when in the Â“Send UPDATE_ACKÂ” state (i.e., sending UPDATE_ACK messages and waiting for a resynchronization), Â“MUSTÂ” (2LEF9160) perform a timeout of thirty (30) seconds (minimum) while waiting for a resynchronization.	Terminal	
2LEF9180	Threshold	Upon expiration of the timeout, the LEF ECU Â“MUSTÂ” (2LEF9180) send idle bits to perform a resynchronization on the previous traffic key. This is to prevent an LEF ECU from getting stuck upon false detection of an UPDATE_INIT message.	Terminal	
2LEF9200	Threshold	(U//FOUO) In Broadcast mode, the initiating LEF ECU Â“MUSTÂ” (2LEF9200) use the updated key as the traffic key upon initiating a resynchronization.	Terminal POET ACM	
2LEF10020	Threshold	(U//FOUO) The LEF ECU "MUST" (2LEF10020) have an HMI indication that an EFF failure has occurred.	Terminal	
2LEF10040	Threshold	(U//FOUO) The LEF ECU "MUST" (2LEF10040) have an HMI support for enabling/disabling of cryptographic authentication for the active cryptographic algorithm.	Terminal	
2LEF10060	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF10060) have the HMI default to the cryptographic authentication enabled selection.	Terminal	
2LEF10080	Threshold	(U//FOUO) The LEF ECU HMI Â“MUSTÂ” (2LEF10080) support HMI selection to manually perform a key update.	Terminal	More specific requirements apply to the POET ACM.
2LEF10100	Threshold	(U//FOUO) If the LEF ECU supports Broadcast Mode it Â“MUSTÂ” (2LEF10100) support HMI selection of the Broadcast Mode of operation.	Terminal	
2LEF10120	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF10120) support HMI selection of symmetric (PPK) or asymmetric EFF keying modes.	Terminal	
2LEF10140	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF10140) indicate the status of a key load operation, (i.e., location of key and whether load was successful).	Terminal POET ACM	
2LEF10160	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF10160) support HMI selection of Full-Duplex Mode of operation.	Terminal	
2LEF10180	Threshold	(U//FOUO) The LEF ECU Â“MUSTÂ” (2LEF10180) support the HMI selection of cryptographic authentication enable/disable.	Terminal	More specific requirements apply to the POET ACM.