IPv6 Ready Logo

Phase-2 Conformance
Test Specification
IPsec

Technical Document

Revision 2.0.0b

IPv6 Forum *http://www.ipv6forum.org/*
Ipv6 Ready Logo Committee *http://www.ipv6ready.org/*

Modification Record

Version	Date	Editor	Modification
2.0.0	2017-02-24	Timothy Carlin	Reorganized sections Separated ESP from Architecture tests Common Configuration for Manual Keys and Policies Updated Algorithm Requirements according to RFC7321bis Added CHAHA20-POLY1305 to ADVANCED encryption algorithms Changed AES-CBC(128-bit) and NULL from ADVANCED to BASIC encryption algorithms Changed 3DES-CBC from BASIC to ADVANCED encryption algorithms Added AES-GCM(128-bit) to BASIC encryption algorithms Added AES-CBC (192-bit), AES-CBC(256-bit), AES-GCM(192-bit), and AES-GCM(256-bit) to ADVANCED encryption algorithms Changed HMAC-SHA-256 from ADVANCED to BASIC Integrity algorithms Added AES-GMAC(128-bit) to BASIC Integrity algorithms Added HMAC-SHA-384, HMAC-SHA-512, AES-GMAC(192-bit), and AES-GMAC(256-bit) to ADVANCED Integrity algorithms Added test cases for AES-CBC(128-bit) HMAC-SHA-256 (Section 5.2.9, 6.2.9) Added test cases for AES-CBC HMAC-SHA-384 (Section 5.2.10, 6.2.10) Added test cases for AES-CBC(256-bit) HMAC-SHA-512 (Section 5.2.11, 6.2.11) Added test cases for AES-GCM NULL (Section 5.2.12, 6.2.12), RFC 4106 “The Use of Galois/Counter Mode (GCM) in Ipsec Encapsulating Security Payload (ESP)” Added test cases for NULL AES-GMAC (Section 5.2.13, 6.2.13), RFC 4543 “The Use of Galois Message Integrity Code (GMAC) in Ipsec ESP and AH Modified formatting and fixed typos
1.11.0	2011-10-05	Timothy Carlin	Added Section 5.3.6 to verify that End-Node can process a tunneled ICMPv6 Packet Too Big Message and correctly reassemble/fragment packet Modified Section 5.1 End-Node Transport Mode Packet Too Big Reception to fragment inbound Echo Request. Removed ESP Null Authentication Tests Typos and Bug Fixes
1.10.0	2010-05-31	Timothy Carlin	Support Authentication Algorithm HMAC-SHA-256 in RFC 4868 (Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with Ipsec) (Section 5.2.8, and 6.2.8) Added the description to Section 6.1.6 Possible Problems
1.9.2	2010-02-03		Corrected pre-shared key at subsection 5.1.5 Corrected packet format of dummy packet at subsection 6.1.7 Clarified relationship between steps in procedure and Observable Result at all subsections.
1.9.1	2009-01-07		Support the passive node which doesn’t have ping6 application (as Possible Problems in Section 5.1.2)
1.9.0	2008-12-09		

1.8.1	2007-10-11	Support RFC 4312 (The Camellia Cipher Algorithm and Its Use With Ipsec) (Section 5.2.7, 6.2.7) Use Ipv6 prefix defined in RFC 3849 for the documentation Remove ESN test cases (Section 5.1.12, 6.1.14)
1.8.0	2007-05-27	Support Ipsec v3
1.7.7	2006-05-06	Correct 5.3.4 Category
1.7.6	2005-12-22	Correct expected MTU value in ICMP Packet Too Big message for 6.1.5 Packet Too Big Forwarding
1.7.5	2005-09-20	Correct the maximum MTU value for 6.1.4 Packet Too Big Transmission.
1.7.4	2005-06-13	Fix typos
1.7.3	2005-06-07	Removed test for Packet Too Big Forwarding (Known Original Host) for SGW.
1.7.2	2005-05-20	Fix typos
1.7.1	2005-05-18	Change Security Policy for 5.3.2.
1.7	2005-05-08	Add Sequence Number Increment Test. Add ICMP Error Test.
1.6	2005-03-01	Change Keys Add Select SPD test for tunnel mode
1.5	2004-11-26	Change packet description of 5.1.4
1.4	2004-11-19	Change Host to End-Node, Default algorithms changed to (3DES-CBC, HMAC-SHA1) for Architecture test. Editorial fix
1.3	2004-09-24	
1.2	2004-09-22	
1.1	2004-09-13	
1.0	2004-09-08	

Acknowledgments

Ipv6 Forum would like to acknowledge the efforts of the following organizations in the development of this test specification.

- TAHI Project
- University of New Hampshire – Interoperability Laboratory (UNH-IOL)
- IRISA

Table of Contents

Ipv6 Ready Logo **0**
Modification Record **1**
Acknowledgments **3**
Table of Contents **4**
Introduction **6**
Phases of the Ipv6 Logo Program **7**
Requirements **8**
  Equipment Type **8**
  Security Protocol **8**
  Mode **8**
  Keying **8**
  Test Traffic **9**
  Category **9**
  Required Tests **10**
References **12**
    Algorithms 12
    Architecture 12
Test Topology **13**
Description **17**
Common Configurations **18**
  Common Configuration: Sections 1 and 2 .... **19**
    Global Security Associations 19
    ESP 19
    IKEv2 Settings 19
    Manual Settings *(if necessary)* 20
  Common Configuration: Section 3 .... **21**
Section 1: End-Node **22**
  1.1. Ipsec/ESP Architecture (Transport Mode) **23**
    Ipsec.Conf.1.1.1. Select SPD 24
    Ipsec.Conf.1.1.2. Select SPD (ICMP Type) 27
    Ipsec.Conf.1.1.3. Sequence Number Increment 33
    Ipsec.Conf.1.1.4. Packet Too Big Reception 36
    Ipsec.Conf.1.1.5. Receipt of No Next Header 41
    Ipsec.Conf.1.1.6. Bypass Policy 45
    Ipsec.Conf.1.1.7. Discard Policy 48
    Ipsec.Conf.1.1.8. Transport Mode Padding 51
    Ipsec.Conf.1.1.9. Invalid SPI 55
    Ipsec.Conf.1.1.10. Invalid ICV 58
  1.2. Ipsec/ESP Architecture (Tunnel Mode) **61**
    Ipsec.Conf.1.2.1. Tunnel Mode with End-Node 62
    Ipsec.Conf.1.2.2. Tunnel Mode with SGW 64
    Ipsec.Conf.1.2.3. Tunnel Mode Select SPD 67
    Ipsec.Conf.1.2.4. Tunnel Mode Padding 71
    Ipsec.Conf.1.2.5. Tunnel Mode Fragmentation 76

Section 2: SGW Test **82**
2.1. Ipsec/ESP Architecture **83**
Ipsec.Conf.2.1.1. Select SPD 84
Ipsec.Conf.2.1.2. Select SPD (2 Hosts) 89
Ipsec.Conf.2.1.3. Sequence Number Increment 95
Ipsec.Conf.2.1.4. Packet Too Big Transmission 98
Ipsec.Conf.2.1.5. Packet Too Big Forwarding 101
Ipsec.Conf.2.1.6. Receipt of No Next Header 106
Ipsec.Conf.2.1.7. Bypass Policy 110
Ipsec.Conf.2.1.8. Discard Policy 113
Ipsec.Conf.2.1.9. Tunnel Mode Padding 116
Ipsec.Conf.2.1.10. Invalid SPI 121
Ipsec.Conf.2.1.11. Invalid ICV 124
Ipsec.Conf.2.1.12. Tunnel Mode with End-Node 127

Section 3: ESP **130**
3.1. ESP Algorithms **131**
ESP Common Configurations 131
Ipsec.Conf.3.1.1. End-Node ESP Algorithms (Transport Mode) 133
Ipsec.Conf.3.1.2. End-Node ESP Algorithms (Tunnel Mode) 135
Ipsec.Conf.3.1.3. SGW ESP Algorithms 137

Appendix A: Annex-5.1.2 for the Passive Node **140**
Using UDP application to invoke ICMPv6 Destination Unreachable (Port unreachable) **141**
Invoking Neighbor Unreachability Detection **146**

Appendix B: Manual Settings Disallowed **154**
AES-CCM **154**
AES-GCM **154**
AES-GMAC **155**
ChaCha20-Poly1305 ..... **155**

Introduction

The Ipv6 forum plays a major role to bring together industrial actors, to develop and deploy the next generation of IP protocols. Contrary to Ipv4, which started with a small closed group of implementers, the universality of Ipv6 leads to a huge number of implementations. Interoperability has always been considered as a critical feature in the Internet community.

Due to the large number of Ipv6 implementations, it is important to provide the market a strong signal proving the level of interoperability across various products. To avoid confusion in the mind of customers, a globally unique logo program should be defined. The Ipv6 logo will give confidence to users that Ipv6 is currently operational. It will also be a clear indication that the technology will still be used in the future. To summarize, this logo program will contribute to the feeling that Ipv6 is available and ready to be used.

Phases of the Ipv6 Logo Program

Phase 1
In the first stage, the Logo will indicate that the product includes Ipv6 mandatory core protocols and can interoperate with other Ipv6 implementations.

Phase 2
The "Ipv6 ready" step implies a proper care, technical consensus and clear technical references. The Ipv6 ready logo will indicate that a product has successfully satisfied strong requirements stated by the Ipv6 Ready Logo Committee (v6RLC).
To avoid confusion, the logo "Ipv6 Ready" will be generic. The v6RLC will define the test profiles with associated requirements for specific functionalities.

Phase 3
Same as Phase 2 with Ipsec mandated.

Requirements

To obtain the Ipv6 Ready Logo Phase-2 for Ipsec (Ipsec Logo), the Node Under Test (NUT) must satisfy following requirements.

Equipment Type

- End-Node (EN)

  A node that uses Ipsec only for itself. Hosts and Routers can be End-Nodes.

- Security Gateway (SGW)

  A node that can provide Ipsec Tunnel Mode for nodes behind it. Routers can be SGWs.

Security Protocol

NUTs must utilize ESP regardless of the type of the NUT. The Ipv6 Ready Logo Program does not test AH.

Mode

The mode requirement depends on the type of NUT.

- End-Node:

  If the NUT is an End-Node, it must pass all of the Transport Mode mode tests. If the NUT supports tunnel mode, it must pass all of the Tunnel Mode tests (i.e. Tunnel mode is an advanced functionality for End-Node NUTs).

- SGW:

  If the NUT is a SGW, it must pass all of the Tunnel Mode tests.

Keying

Previous versions of this test suite required Manual Keying by default, as a minimum requirement. Developments in industry best practices have shown that Manual Keys pose a significant security risk.

According to RFC 7321bis, Section 3:

```
Manual Keying is not be used as it is inherently dangerous.  Without
any keying protocol, it does not offer Perfect Forward Secrecy
(“PFS”) protection.  Deployments tend to never be reconfigured with
fresh session keys.  It also fails to scale and keeping SPI’s unique
amongst many servers is impractical.  This document was written for
deploying ESP/AH using IKE (RFC7298) and assumes that keying happens
using IKEv2.

If manual keying is used anyway, ENCR_AES_CBC MUST be used, and
```

```
ENCR_AES_CCM, ENCR_AES_GCM and ENCR_CHACHA20_POLY1305 MUST NOT be
used as these algorithms require IKE.
```

Following this recommendation, a configuration using Dynamic Keying, facilitated by IKE is used by default, and specifically IKEv2. IKEv1 is obsolete and not supported. Devices which support only Manual Keys will not successfully pass these tests, as the BASIC combined-mode (AEAD) algorithms require Dynamic Keying.

When IKEv2 is used, the encryption keys and Integrity keys are negotiated dynamically. The tester should support the alternative of using IKE with dynamic keys to execute the tests. Manual Keys may be used in tests that have indicated they are acceptable. These tests are run with IKEv2, and if necessary, run again with Manual Keys.

Test Traffic

All tests use ICMP Echo Request and Echo Reply messages by default. ICMP is independent from any implemented application and this adds clarity to the test. If the NUT cannot apply Ipsec for ICMPv6 packets, it is acceptable to use other protocols rather than ICMPv6.

In this case, the device must support ICMPv6, TCP, or UDP. The application and port number are unspecified when TCP or UDP packets are used. The test coordinator should support any ports associated with an application used for the test. Applicants must mention the specific protocol and port that was used to execute the tests.

Category

In this document, the tests and algorithms are categorized into two types: BASIC and ADVANCED

ALL NUTs are required to support BASIC. ADVANCED tests are required for all NUTs which support ADVANCED encryption/Integrity algorithms. Each test description contains a Category section. The section lists the requirements to satisfy each test.

Required Tests

Test Case	Title	Ipv6Ready Requirement
Ipsec.Conf.1.1.1	Select SPD	EN: Basic
Ipsec.Conf.1.1.2 Part A	Select SPD (Select ICMPv6 Type)	EN: Basic
Ipsec.Conf.1.1.2 Part B	Select SPD (Select TCP Port)	EN: Basic
Ipsec.Conf.1.1.3	Sequence Number Increment	EN: Basic
Ipsec.Conf.1.1.4	Packet Too Big Reception	EN: Basic
Ipsec.Conf.1.1.5 Part A	Receipt of No Next Header	EN: Basic
Ipsec.Conf.1.1.5 Part B	Receipt of No Next Header (TFC)	EN: Advanced
Ipsec.Conf.1.1.6	Bypass Policy	EN: Basic
Ipsec.Conf.1.1.7	Discard Policy	EN: Basic
Ipsec.Conf.1.1.8 Part A	Transport Mode Padding	EN: Basic
Ipsec.Conf.1.1.8 Part B	Transport Mode Padding (TFC)	EN: Advanced
Ipsec.Conf.1.1.9	Invalid SPI	EN: Basic
Ipsec.Conf.1.1.10	Invalid ICV	EN: Basic
Ipsec.Conf.1.2.1	Tunnel Mode with End-Node	EN: Basic
Ipsec.Conf.1.2.2	Tunnel Mode with SGW	EN: Basic
Ipsec.Conf.1.2.3	Tunnel Mode Select SPD	EN: Basic
Ipsec.Conf.1.2.4 Part A	Tunnel Mode Padding	EN: Basic
Ipsec.Conf.1.2.4 Part B	Tunnel Mode Padding (TFC)	EN: Advanced
Ipsec.Conf.1.2.5	Tunnel Mode Fragmentation	EN: Basic
Ipsec.Conf.2.1.1	Select SPD	SGW: Basic
Ipsec.Conf.2.1.2	Select SPD (Two Hosts)	SGW: Basic
Ipsec.Conf.2.1.3	Sequence Number Increment	SGW: Basic
Ipsec.Conf.2.1.4	Packet Too Big Transmission	SGW: Basic
Ipsec.Conf.2.1.5	Packet Too Big Forwarding	SGW: Basic
Ipsec.Conf.2.1.6 Part A	Receipt of No Next Header	SGW: Basic
Ipsec.Conf.2.1.6 Part B	Receipt of No Next Header (TFC)	SGW: Advanced
Ipsec.Conf.2.1.7	Bypass Policy	SGW: Basic
Ipsec.Conf.2.1.8	Discard Policy	SGW: Basic
Ipsec.Conf.2.1.9 Part A	Transport Mode Padding	SGW: Basic
Ipsec.Conf.2.1.9 Part B	Transport Mode Padding (TFC)	SGW: Advanced
Ipsec.Conf.2.1.10	Invalid SPI	SGW: Basic
Ipsec.Conf.2.1.11	Invalid ICV	SGW: Basic
Ipsec.Conf.2.1.12	Tunnel Mode with End-Node	SGW: Basic
Ipsec.Conf.3.1.1	End-Node ESP Algorithms EN: Must run Test Parts marked "Basic" SGW: All Test Parts are "Advanced"	EN:Basic SGW: Advanced
Ipsec.Conf.3.1.2	End-Node ESP Algorithms EN: Must run Test Parts marked "Basic" SGW: All Test Parts are "Advanced"	EN:Basic SGW: Advanced
Ipsec.Conf.3.1.3	SGW ESP Algorithms SGW: Must run Test Parts marked "Basic"	EN: N/A SGW: Basic
Ipsec.Conf.3.1.X Part A	NULL/SHA256	Basic
Ipsec.Conf.3.1.X Part B	AES128/SHA1	Basic
Ipsec.Conf.3.1.X Part C	AES128/SHA256	Basic
Ipsec.Conf.3.1.X Part D	AES256/SHA256	Basic
Ipsec.Conf.3.1.X Part E	AES256/SHA512	Advanced
Ipsec.Conf.3.1.X Part F	AESCCM128/AESXCBC	Advanced
Ipsec.Conf.3.1.X Part G	AESCCM256/AESXCBC	Advanced
Ipsec.Conf.3.1.X Part H	AESGCM128	Basic
Ipsec.Conf.3.1.X Part I	AESGCM256	Basic
Ipsec.Conf.3.1.X Part J	AESGMAC128	Basic
Ipsec.Conf.3.1.X Part K	AESGMAC256	Basic

References

This test specification focuses on the following Ipsec related RFCs.

Algorithms		
RFC2404	HMAC-SHA1	The Use of HMAC-SHA-1-96 within ESP and AH. C. Madson, R. Glenn. November 1998. (Format: TXT=13089 bytes) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC2404)
RFC2410	NULL Encryption	The NULL Encryption Algorithm and Its Use With Ipsec. R. Glenn, S. Kent. November 1998. (Format: TXT=11239 bytes) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC2410)
RFC2451	ESP CBC	The ESP CBC-Mode Cipher Algorithms. R. Pereira, R. Adams. November 1998. (Format: TXT=26400 bytes) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC2451)
RFC3566	AES-XCBC-MAC	The AES-XCBC-MAC-96 Algorithm and Its Use With Ipsec. S. Frankel, H. Herbert. September 2003. (Format: TXT=24645 bytes) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC3566)
RFC3602	AES-CBC	The AES-CBC Cipher Algorithm and Its Use with Ipsec. S. Frankel, R. Glenn, S. Kelly. September 2003. (Format: TXT=30254 bytes) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC3602)
RFC3686	AES-CTR	Using Advanced Encryption Standard (AES) Counter Mode With Ipsec Encapsulating Security Payload (ESP). R. Housley. January 2004. (Format: TXT=43777 bytes) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC3686)
RFC4106	GCM with ESP	The Use of Galois/Counter Mode (GCM) in Ipsec Encapsulating Security Payload (ESP). J. Viega, D. McGrew. June 2005. (Format: TXT=23399 bytes) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC4106)
RFC4309	AES-CCM	Using Advanced Encryption Standard (AES) CCM Mode with Ipsec Encapsulating Security Payload (ESP). R. Housley. December 2005. (Format: TXT=28998 bytes) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC4309)
RFC4543	GMAC with ESP	The Use of Galois Message Authentication Code (GMAC) in Ipsec ESP and AH. D. McGrew, J. Viega. May 2006. (Format: TXT=29818 bytes) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC4543)
RFC4868	HMAC-SHA256, 384, 512	Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with Ipsec. S. Kelly, S. Frankel. May 2007. (Format: TXT=41432 bytes) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC4868)
RFC7634	ChaCha20 Poly1305	ChaCha20, Poly1305, and Their Use in the Internet Key Exchange Protocol (IKE) and Ipsec. Y. Nir. August 2015. (Format: TXT=27513 bytes) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC7634)
RFC7321bis	ESP Req	TBD
Architecture		
RFC4301	Ipsec Arch	Security Architecture for the Internet Protocol. S. Kent, K. Seo. December 2005. (Format: TXT=262123 bytes) (Obsoletes RFC2401) (Updates RFC3168) (Updated by RFC6040, RFC7619) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC4301)
RFC4303	ESP	IP Encapsulating Security Payload (ESP). S. Kent. December 2005. (Format: TXT=114315 bytes) (Obsoletes RFC2406) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC4303)
RFC4443	ICMPv6	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (Ipv6) Specification. A. Conta, S. Deering, M. Gupta, Ed.. March 2006. (Format: TXT=48969 bytes) (Obsoletes RFC2463) (Updates RFC2780) (Updated by RFC4884) (Status: DRAFT STANDARD) (DOI: 10.17487/RFC4443)
RFC7296	IKEv2	Internet Key Exchange Protocol Version 2 (IKEv2). C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, T. Kivinen. October 2014. (Format: TXT=354358 bytes) (Obsoletes RFC5996) (Updated by RFC7427, RFC7670) (Also STD0079) (Status: INTERNET STANDARD) (DOI: 10.17487/RFC7296)

Test Topology

For End-Node vs. End-Node Transport/Tunnel Mode Test

1. Set global address of NUT via SLAAC(NUT_Network0)
2. Set MTU of NUT via RA (MTU value is 1500 for Network 0)
3. Ipsec Transport Mode between NUT and TN1 and TN2

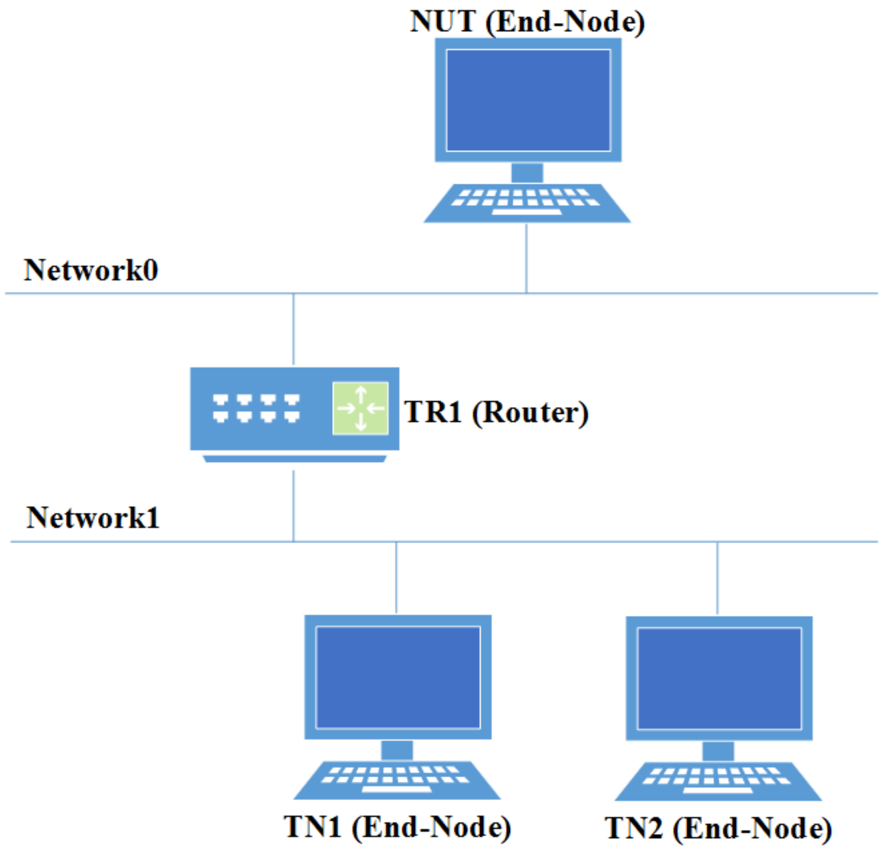


Figure 1 Topology for End-Node: Transport and Tunnel mode with End-Node

For End-Node vs. SGW Tunnel Mode Test

1. Set global address to NUT by RA
2. Set MTU to NUT by RA (MTU value is 1500 for Network 0)
3. Ipsec Tunnel Mode between NUT and TN1.

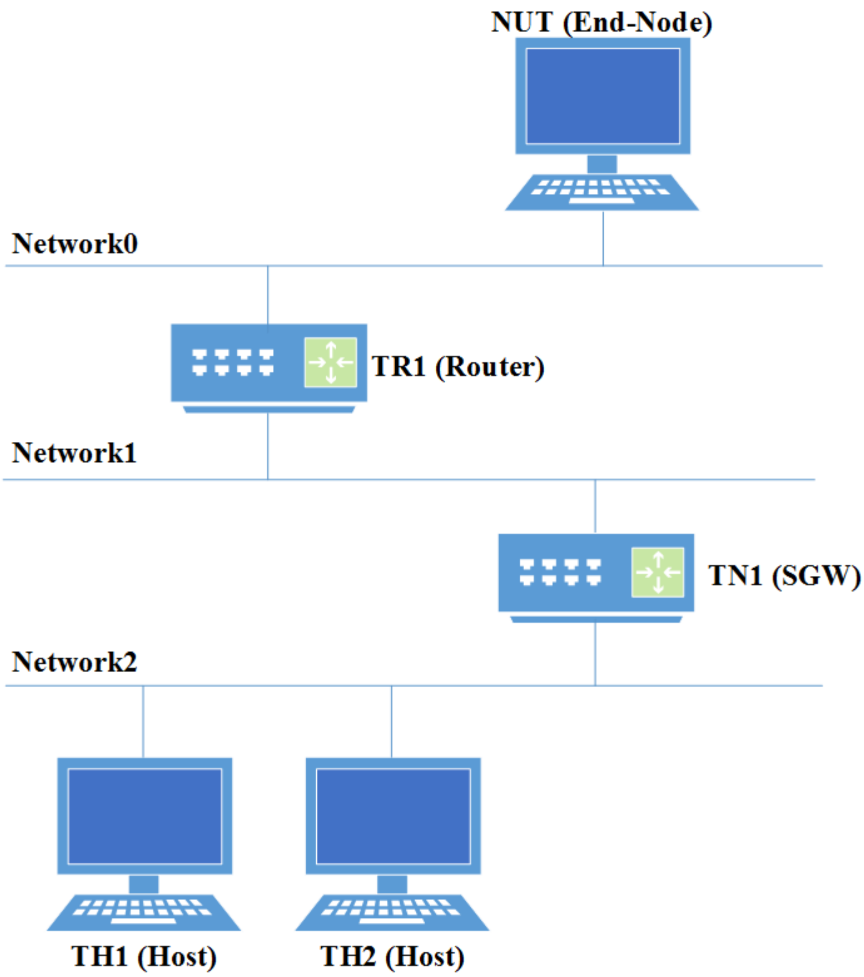


Figure 2 Topology for End-Node: Tunnel mode with SGW

For SGW: Tunnel Mode with End-Node Test

1. Set global address of NUT manually (NUT_Network0, NUT_Network1)
2. Set routing table of NUT manually (TR1_Network1 for Network2)
3. Set MTU of NUT manually for Network 0 and Network1 (MTU value is 1500 for Network 0 and Network1)
4. Ipsec Tunnel Mode between NUT and TH2.

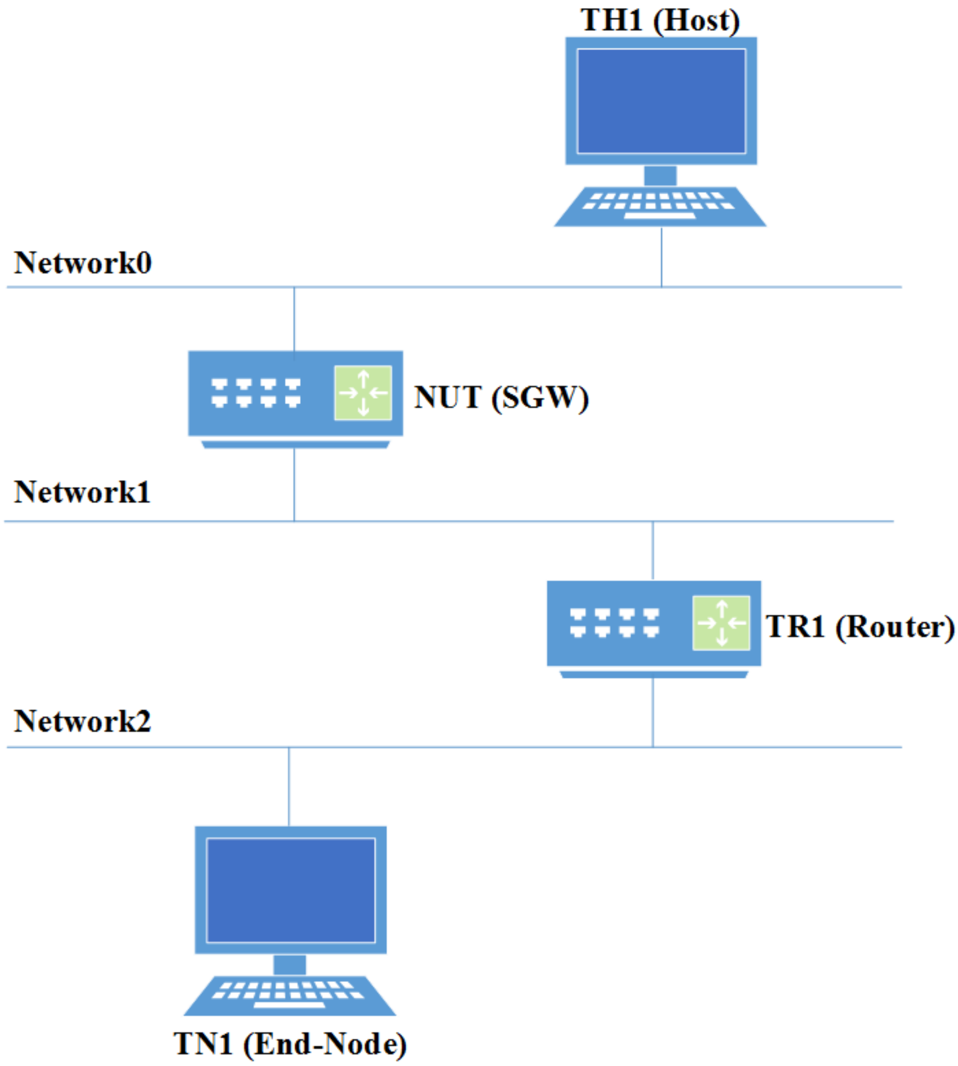


Figure 3 Topology for SGW: Tunnel mode with End-Node

For SGW: Tunnel Mode Test

1. Set global address of NUT manually (NUT_Network0, NUT_Network1)
2. Set routing table of NUT manually (TR1_Network1 for Network2, Network3 and Network4)
3. Set MTU of NUT manually for Network 0 and Network1 (MTU value is 1500 for Network 0 and Network1)

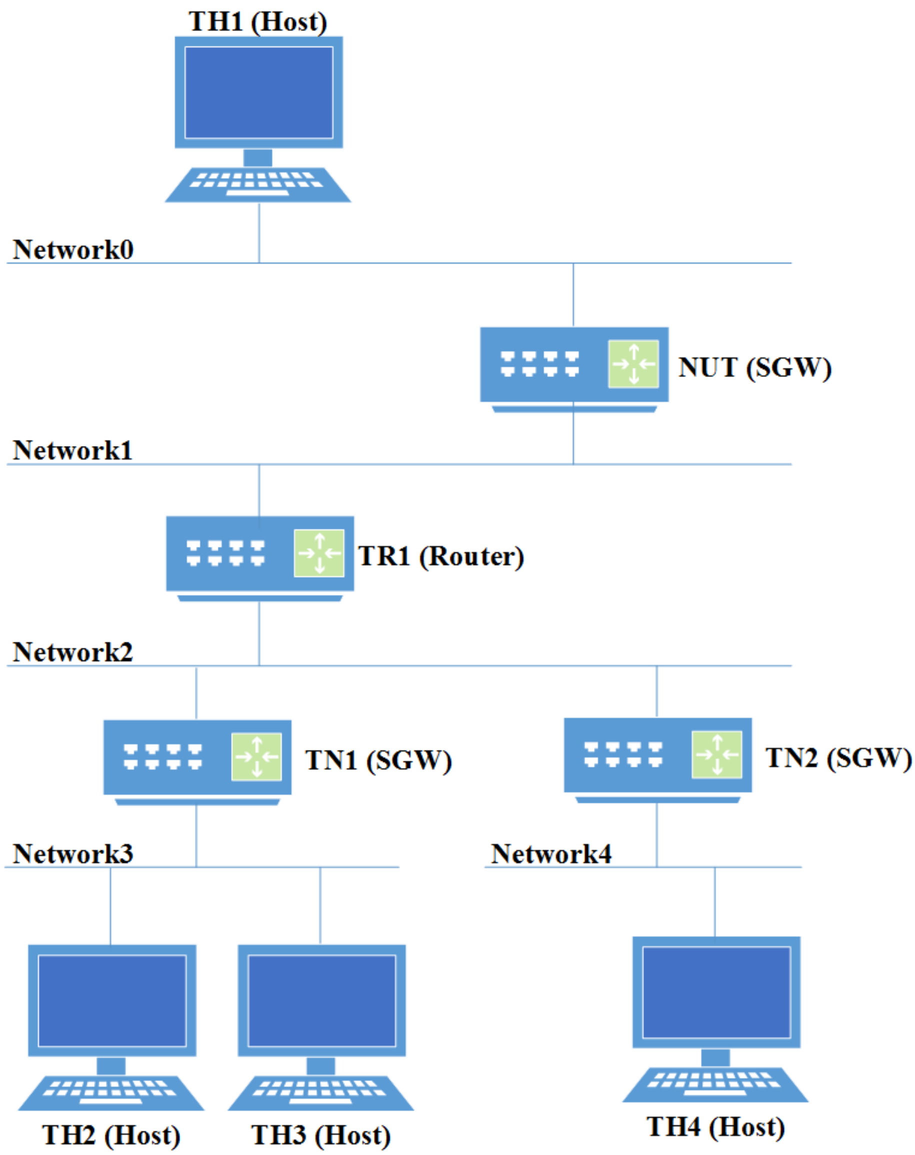


Figure 4 Topology for SGW: Tunnel mode with SGW

Description

Each test scenario consists of the following parts.

Purpose:	The 'Purpose' is the short statement describing what the test attempts to achieve. It is usually phrased as a simple assertion of the future or capability to be tested.
Initialization:	The 'Initialization' section describes how to initialize and configure the NUT before starting each test. If a value is not provided, then the protocol's default value is used.
Database	The 'Database' section describes the needed configuration for the Policy Database for the test case.
Packets:	The 'Packets' section describes the simple format of the packets used in the test. In this document, the packet name is represented in Italic style font.
Procedure:	The 'Procedure' describes the step-by-step instructions for carrying out the test.
Observable Results:	The 'Observable Results' section describes the expected result. The NUT passes the test if the results described in this section are obtained.
Possible Problems:	The 'Possible Problems' section contains a description of known issues with the test procedure, which may affect test results in certain situations.

Common Configurations

This section defines the Common Configurations referenced by various test cases.

Common Configuration: Sections 1 and 2

The Common Configurations described below should be utilized for test cases in Sections 1 and 2, unless otherwise modified or specified by the test case. Both End-Node and SGW devices should utilize the configurations described below.

Global Security Associations

Unless otherwise specified, the dynamically negotiated settings and algorithms below are used for every test case.

The IKEv2 settings apply for test cases that use 1 or more Security Association, however the Traffic Selectors may change, and are specified in the test case.

IKEv2 is the preferred mechanism for negotiating keys and configuring settings. If necessary, the Manual Settings may be used in the absence of IKEv2, or for debugging.

ESP	
ESP Encryption Algorithm	ENCR_AES_CBC (128-bit)
ESP Integrity Algorithm	AUTH_HMAC_SHA2_256_128

IKEv2 Settings	
IKE Encryption Algorithm	ENCR_AES_CBC (128-bit)
IKE Integrity Algorithm	AUTH_HMAC_SHA2_256_128
IKE PRF Algorithm	PRF_HMAC_SHA2_256
IKE DH Group	14 (2048-bit MODP Group)
Authentication Method	PSK: IPSECTEST12345678!
ID Type	ID_IPV6_ADDR

Manual Settings *(if necessary)*	
SA1-I	
Direction	Incoming
SPI	0x1000
Encryption Key	ipv6readaescin01
Integrity Key	ipv6readylogoph2ipsecsha2256in01
SA1-O	
Direction	Outgoing
SPI	0x2000
Encryption Key	ipv6readaescout1
Integrity Key	ipv6readylogoph2ipsecsha2256out1
SA2-I	
Direction	Incoming
SPI	0x3000
Encryption Key	ipv6readaescin02
Integrity Key	ipv6readylogoph2ipsecsha2256in02
SA2-O	
Direction	Outgoing
SPI	0x4000
Encryption Key	ipv6readaescout2
Integrity Key	ipv6readylogoph2ipsecsha2256out2

Common Configuration: Section 3

Reference the list of algorithms specified in the Section 3.1: ESP Common Configurations.

Section 1: End-Node

This Chapter describes the test specification for End-Node.
The test specification consists of 2 sections. One is regarding “Ipsec Architecture” and the other is regarding “Encryption and Integrity Algorithms”.

1.1. Ipsec/ESP Architecture (Transport Mode)

Scope:

Following tests focus on Ipsec Architecture.

Overview:

Tests in this section verify that a node properly process and transmit based on the Security Policy Database and Security Association Database.

Ipsec.Conf.1.1.1. Select SPD

Purpose:

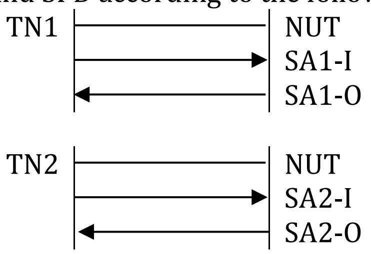
Verify that a NUT (End-Node) selects appropriate SPD based on Address

Initialization:

- Network Topology
  - o Connect the devices according to Common Topology 1
- Configuration
  - o Use Global Security Associations

Databases:

Set NUT’s SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Transport
Remote Traffic Selector	TN1_Network1
Local Traffic Selector	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Policy 2	
Peer	TN2_Network1
Mode	Transport
Remote Address	TN2_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA2-I
Outgoing SA	SA2-O

Packets:

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	*Dynamic1 or 0x1000*
	Sequence	1
	Encrypted Data/ICV	SA1-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA1-I's ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	*Dynamic2 or 0x2000*
	Sequence	1
	Encrypted Data/ICV	SA1-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA1-O's ESP

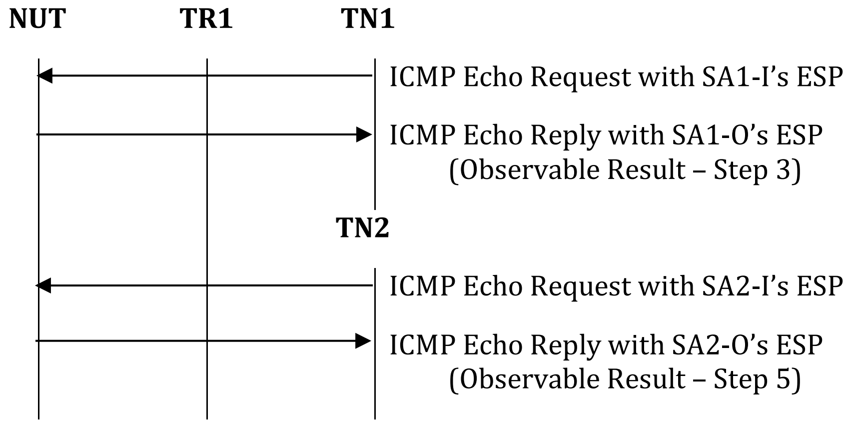
IP Header	Source Address	TN2_Network1
	Destination Address	NUT_Network0
ESP	SPI	*Dynamic3 or* 0x3000
	Sequence	1
	Encrypted Data/ICV	SA2-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA2-I's ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN2_Network1
ESP	SPI	*Dynamic4 or* 0x4000
	Sequence	1
	Encrypted Data/ICV	SA2-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA2-O's ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits *ICMP Echo Request with SA1-I's ESP*	
3.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with SA1-O's ESP*
4.	TN2 transmits *ICMP Echo Request with SA2-I's ESP*	
5.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with SA2-O's ESP*

Possible Problems:

None

Ipsec.Conf.1.1.2. Select SPD (Next Layer Protocol Selectors)

Purpose:

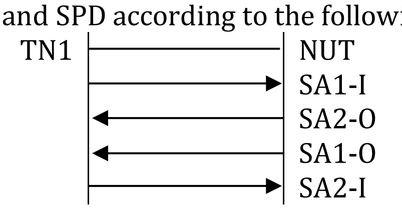
Verify that a NUT (End-Node) selects appropriate SPD based different Next Layer Protocol Selectors, including: ICMPv6 Type, TCP port

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:



Part A: Select ICMPv6 Type

Policy 1	
Peer	TN1_Network1
Mode	Transport
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ICMPv6/128 (Echo Request)
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Policy 2	
Peer	TN1_Network1
Mode	Transport
Remote Address	TN2_Network1
Local Address	NUT_Network0
Protocol/Port	ICMPv6/129 (Echo Reply)
If using Manual Keys include:	
Incoming SA	SA2-I
Outgoing SA	SA2-O

Packets:

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	*Dynamic1 or 0x1000*
	Sequence	1
	Encrypted Data/ICV	SA1-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA1-I's ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	*Dynamic2 or* 0x4000
	Sequence	1
	Encrypted Data/ICV	SA2-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA2-O's ESP

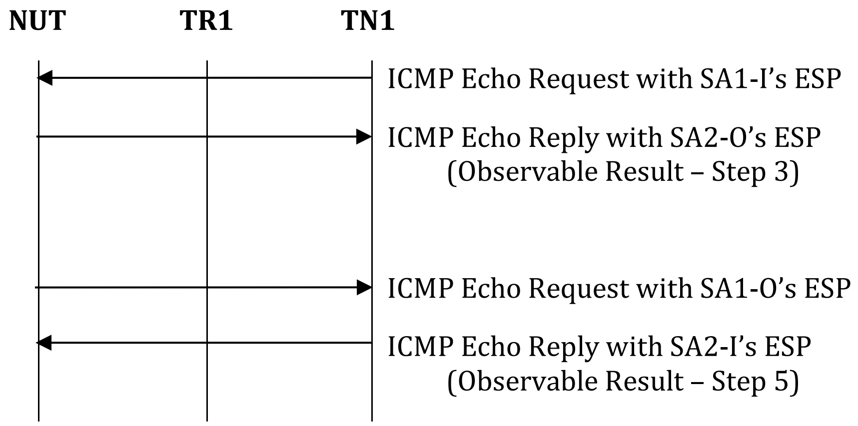
IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	*Dynamic3 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	SA1-O
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA1-O's ESP

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	*Dynamic4 or* 0x3000
	Sequence	1
	Encrypted Data/ICV	SA2-I
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA2-I's ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits *ICMP Echo Request with SA1-I's ESP*	
3.	Observe the packets transmitted on Network 0	The NUT transmits ICMP Echo Reply with SA2-O's ESP
4.	Transmit ICMP Echo Request with SA1-O's ESP from the NUT to the Global unicast address of TN1	
5.	Observe the packets transmitted on Network 0	TN1 transmits ICMP Echo Reply with SA2-I's ESP

Part B: Select TCP Port

Policy 1	
Peer	TN1_Network1
Mode	Transport
Remote Address/Port	TN1_Network1/50001
Local Address/Port	NUT_Network0/55005
Protocol	TCP
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Policy 2	
Peer	TN1_Network1
Mode	Transport
Remote Address/Port	TN1_Network1/60001
Local Address/Port	NUT_Network0/65005
Protocol	TCP
If using Manual Keys include:	
Incoming SA	SA2-I
Outgoing SA	SA2-O

Packets:

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	*Dynamic1 or 0x1000*
	Sequence	1
	Encrypted Data/ICV	SA1-I
TCP	Type	SYN
	Source Port	50001
	Destination Port	55005

TCP SYN with SA1-I's ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	SA2-O
TCP	Type	RST
	Source Port	55005
	Destination Port	50001

TCP RST Reply with SA1-O's ESP

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0

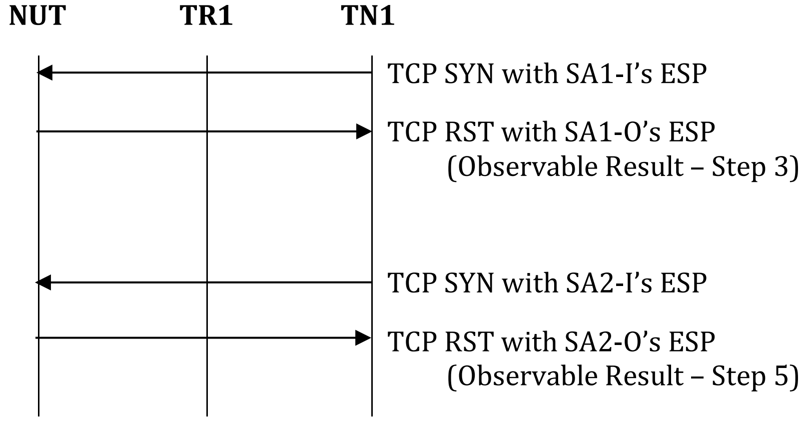
ESP	SPI	*Dynamic3 or 0x3000*
	Sequence	1
	Encrypted Data/ICV	SA1-I
TCP	Type	SYN
	Source Port	60001
	Destination Port	65005

TCP SYN with SA1-I's ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	*Dynamic4 or* 0x4000
	Sequence	1
	Encrypted Data/ICV	SA2-O
TCP	Type	RST
	Source Port	65005
	Destination Port	60001

TCP RST Reply with SA1-O's ESP

Procedure:



Step	Action	Expected Result
6.	Initialize the NUT	
7.	TN1 transmits *TCP SYN with SA1-I's ESP*	
8.	Observe the packets transmitted on Network 0	The NUT transmits TCP RST with SA1-O's ESP
9.	Transmit *TCP SYN with SA2-I's ESP from the NUT* to the Global unicast address of TN1	

10.	Observe the packets transmitted on Network 0	TN1 transmits TCP RST with SA2-O's ESP

Possible Problems:

- Part A: NUT may be a passive node that does not implement an application for sending Echo Requests. One of the following methods to perform this test is required for the passive node:
  - Using UDP application to invoke ICMPv6 Destination Unreachable (Port unreachable) (see Appendix-A Section 1.1)
  - Invoking Neighbor Unreachability Detection (see Appendix-A Section 1.2)
- Part B:
  - Ensure the NUT has no service listening on the prescribed ports, or select alternative ports.

Ipsec.Conf.1.1.3. Sequence Number Increment

Purpose:

Verify that a NUT (End-Node) increases sequence number correctly, starting with 1.

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:

TN1 NUT
SA-I
SA-O

Policy 1	
Peer	TN1_Network1
Mode	Transport
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

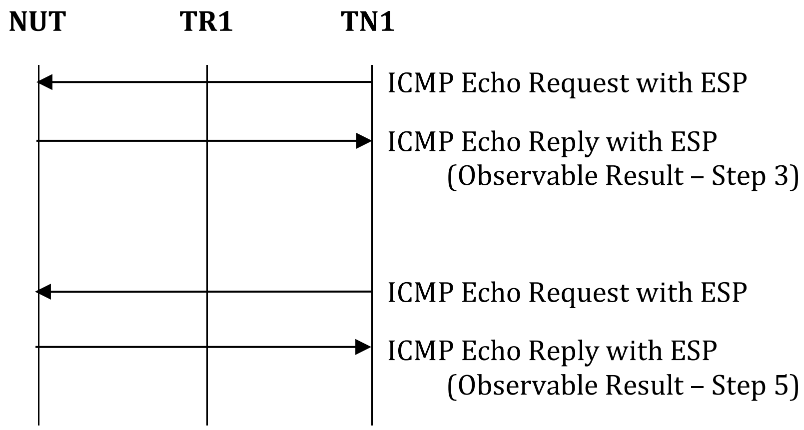
IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	*Dynamic1 or 0x1000*
	Sequence	$1^{st} = 1$, $2^{nd} = 2$
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	*Dynamic2 or 0x2000*
	Sequence	$1^{st} = 1$, $2^{nd} = 2$
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits *ICMP Echo Request with ESP*	
3.	Observe the packets transmitted on Network 0	The NUT transmits an *ICMP Echo Reply with ESP* with an ESP Sequence Number of 1
4.	TN1 transmits *ICMP Echo Request with ESP*	
5.	Observe the packets transmitted on Network 0	The NUT transmits an *ICMP Echo Reply with ESP*

		with an ESP Sequence Number of 2

Possible Problems:

None

Ipsec.Conf.1.1.4. Packet Too Big Reception

Purpose:

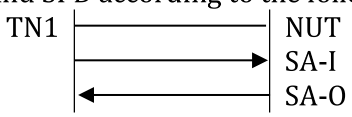
Verify that a NUT (End-Node) can fragment and reassemble fragments correctly.

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - Use Global Security Associations
  - In addition, configure TR1_Network1 to have an MTU of 1280 bytes.

Databases:

Set NUT's SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Transport
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
	Payload Length	1240
Fragment Header	Offset	0
	More	1
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP 1

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
	Payload Length	116
Fragment Header	Offset	154
	More	0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP 2

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
	Payload Length	1340
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	TR1_Network1
	Destination Address	NUT_Network0
ICMP	Type	2 (Packet Too Big)
	MTU	1280
	Data	*1232Byte of ICMP Echo Reply with ESP*

ICMP Error Message (Packet Too Big)

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
	Payload Length	1240
Fragment	Offset	0
	More Flag	1
ESP	SPI	Dynamic2 or 0x2000

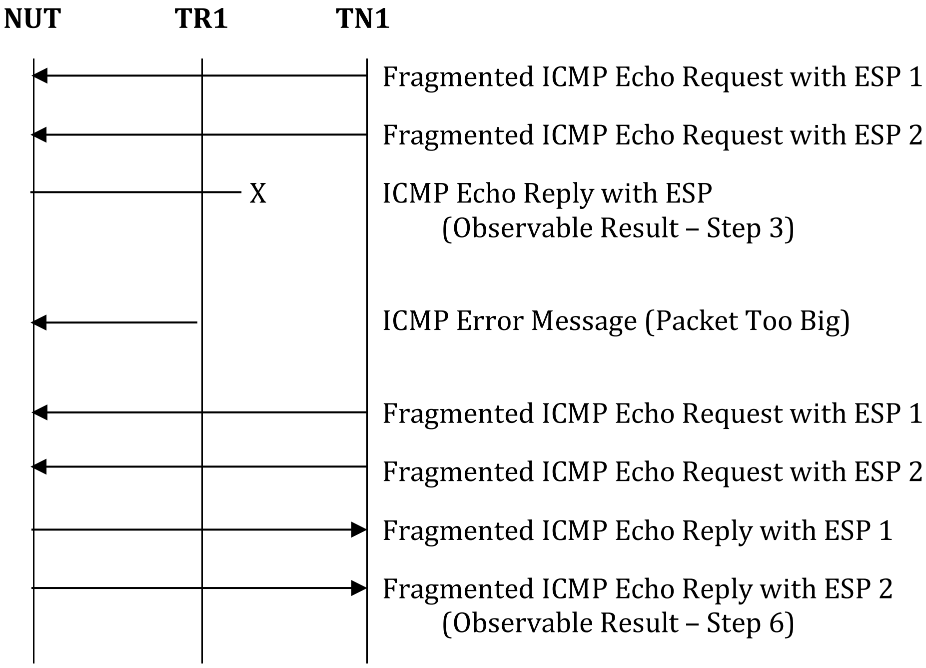
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

Fragmented ICMP Echo Reply with ESP 1

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
	Payload Length	116
Fragment	Offset	154
	More Flag	0
Data	Data	Rest of *ICMP Echo Reply with ESP*

Fragmented ICMP Echo Reply with ESP 2

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits *ICMP Echo Request with ESP*	
3.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with ESP*
4.	TR1 transmits *ICMP Error Message (Packet Too Big)* to the NUT	
5.	TN1 sends *Fragmented ICMP Echo Request with ESP 1* and *Fragmented ICMP Echo Request with ESP 2*	
6.	Observe the packets transmitted on Network 0	The NUT transmits *Fragmented ICMP Echo Reply with ESP 1 and Fragmented ICMP Echo Reply with ESP 2*

Possible Problems:

None

Ipsec.Conf.1.1.5. Receipt of No Next Header

Purpose:

Verify that a NUT (End-Node) processes the dummy packet (the protocol value 59) correctly.

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:

TN1 —— NUT

→ SA-I

← SA-O

Policy 1	
Peer	TN1_Network1
Mode	Transport
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA-I's ESP

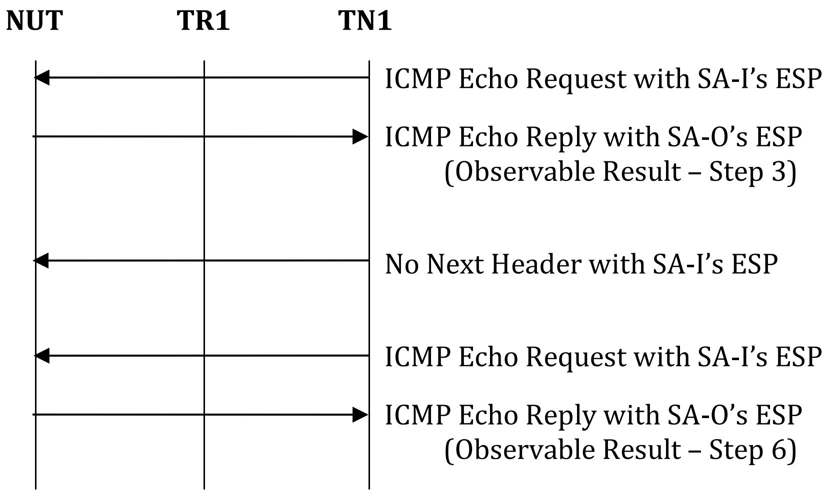
IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA-O's ESP

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
	Next Header	no next header (59)
Upper Layer	Data	empty

No Next Header with SA-I's ESP

Procedure:



Part A: No Next Header

Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits *ICMP Echo Request with SA-I's ESP*	
3.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with SA-O's ESP*
4.	TN1 transmits *No Next Header with SA-I's ESP* (The ESP sequence number must be incremented according to the packet transmitted at step 2)	
5.	TN1 transmits *ICMP Echo Request with SA-O's ESP* (The ESP sequence number must be incremented according to the packet transmitted at step 4)	
6.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with SA-O's ESP*

Part B: TFC Padding with No Next Header

Step	Action	Expected Result
7.	Initialize the NUT	
8.	TN1 transmits *ICMP Echo Request with SA-I's ESP*	
9.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with SA-O's ESP*
10.	TN1 transmits *No Next Header with SA-O's ESP* (The ESP sequence number must be incremented according to the packet transmitted at step 2 and the data in the upper layer consists of random bytes as the plaintext portion)	
11.	TN1 transmits *ICMP Echo Request with SA-O's ESP* (The ESP sequence number must be incremented according to the packet transmitted at step 4)	
12.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with SA-O's ESP*

Possible Problems:

None

Ipsec.Conf.1.1.6. Bypass Policy

Purpose:

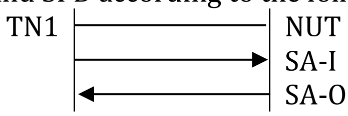
Verify that a NUT (End-Node) can utilize Bypass Policy

Initialization:

- Network Topology
    - Connect the devices according to Common Topology 1
- Configuration
    - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Transport
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Policy 2	
Peer	TN2_Network1
Mode	BYPASS
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
	Payload Length	1460
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
	Payload Length	1460
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

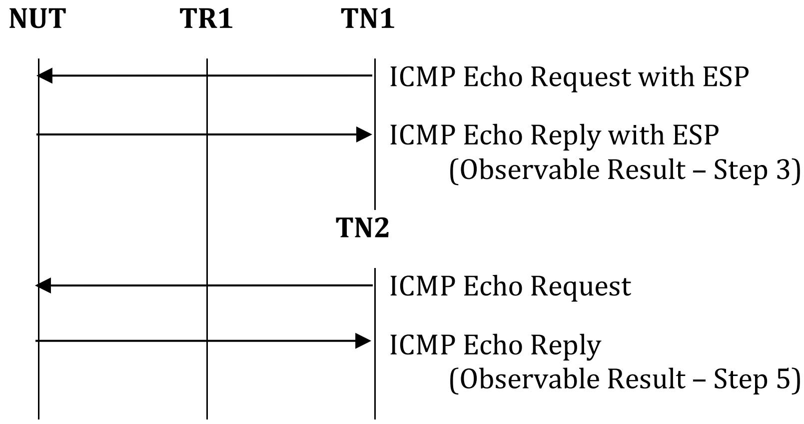
IP Header	Source Address	TN2_Network1
	Destination Address	NUT_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	NUT_Network0
	Destination Address	TN2_Network1
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits *ICMP Echo Request with ESP*	
3.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with SA-O's ESP*
4.	TN2 transmits *ICMP Echo Request*	
5.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply*

Possible Problems:

Instead of specifying an address to bypass, a "bypass others by default" policy may also be enabled to discard address not covered by an Ipsec policy.

Ipsec.Conf.1.1.7. Discard Policy

Purpose:

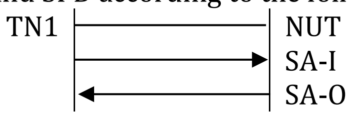
Verify that a NUT (End-Node) can utilize discard policy

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Transport
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Policy 2	
Peer	TN2_Network1
Mode	DISCARD
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
	Payload Length	1460
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
	Payload Length	1460
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

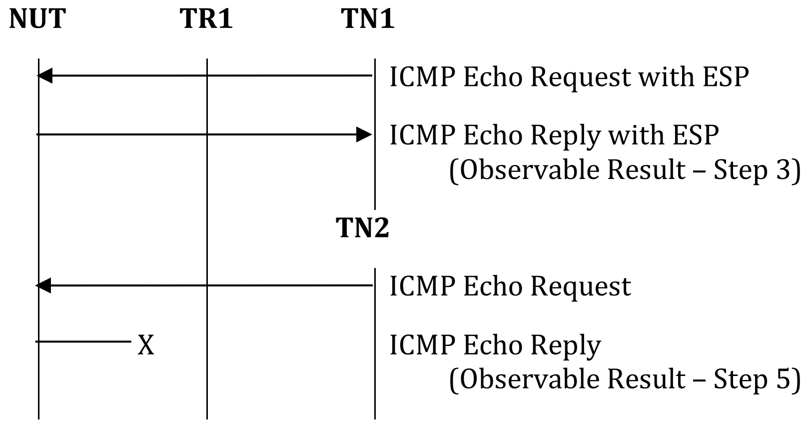
IP Header	Source Address	TN2_Network1
	Destination Address	NUT_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	NUT_Network0
	Destination Address	TN2_Network1
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits *ICMP Echo Request with ESP*	
3.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with ESP*
4.	TN2 transmits *ICMP Echo Request*	
5.	Observe the packets transmitted on Network 0	The NUT never transmits *ICMP Echo Reply*

Possible Problems:

Instead of specifying an address to discard, a “discard others by default” policy may also be enabled to discard addresses not covered by an Ipsec policy.

Ipsec.Conf.1.1.8. Transport Mode Padding

Purpose:

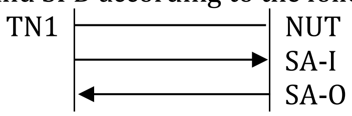
Verify that a NUT (End-Node) supports padding & padding byte handling

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Transport
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Part A: Transport Mode Padding

Packets:

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
	Padding	Sequential
	Padding Length	**7**
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP 1

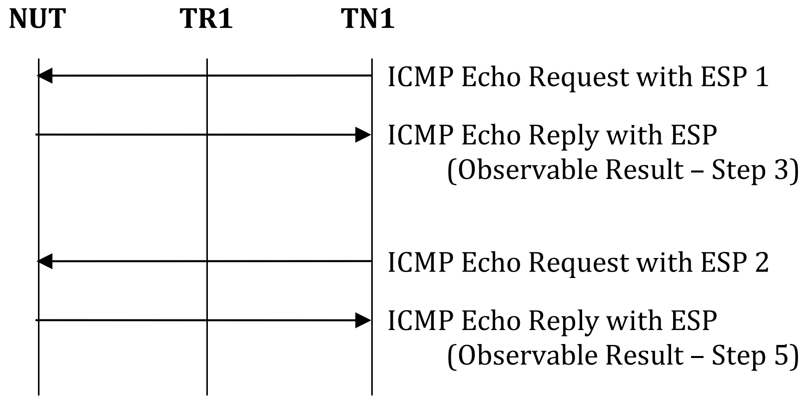
IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
	Padding	Sequential
	Padding Length	**255**
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP 2

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
	Padding Length	7+8n (0 <= n <= 31)
ICMP	Type	129 (Echo Reply)
	Data Length	7

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits ICMP Echo Request with ESP 1	
3.	Observe the packets transmitted on Network 0	The NUT transmits ICMP Echo Reply with ESP
4.	TN1 transmits ICMP Echo Request with ESP 2	
5.	Observe the packets transmitted on Network 0	The NUT transmits ICMP Echo Reply with ESP

Part B: TFC enabled Transport Mode Padding

Packets:

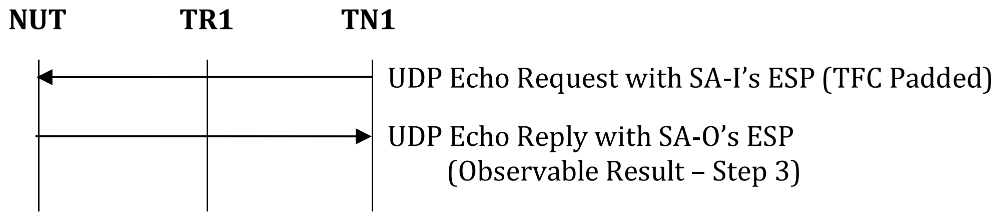
IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
UDP	Source Port	10000
	Destination Port	7 (echo)

UDP Echo Request with SA-I's ESP (TFC Padded)

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
UDP	Source Port	7 (echo)
	Destination Port	10000

UDP Echo Reply with SA-O's ESP

Procedure:



Step	Action	Expected Result
6.	Initialize the NUT	
7.	TN1 transmits UDP Echo Request with SA-I's ESP (TFC Padded)	
8.	Observe the packets transmitted on Network 0	The NUT transmits UDP Echo Reply with SA-O's ESP

Possible Problems:

None

Ipsec.Conf.1.1.9. Invalid SPI

Purpose:

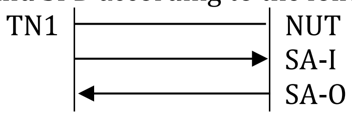
Verify that a NUT (End-Node) correctly processes an, otherwise valid, packet with an invalid SPI

Initialization:

- Network Topology
    - Connect the devices according to Common Topology 1
- Configuration
    - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Transport
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP 1

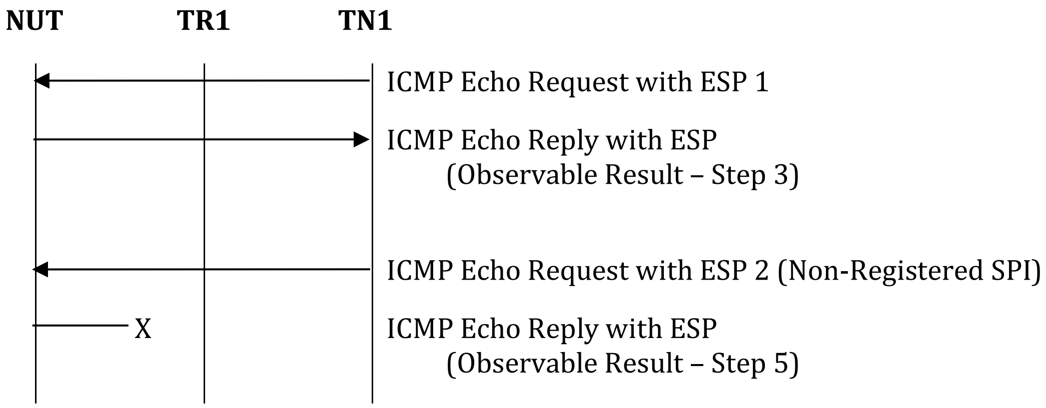
IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	0x9000 (Different from SA-I's SPD)
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP 2 (Non-Registered SPI)

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits ICMP Echo Request with ESP 1	
3.	Observe the packets transmitted on Network 0	The NUT transmits ICMP Echo Reply with ESP
4.	TN1 transmits ICMP Echo Request with ESP 2 (Non-Registered)	
5.	Observe the packets transmitted on Network 0	The NUT never transmits ICMP Echo Reply with ESP

Possible Problems:

None

Ipsec.Conf.1.1.10. Invalid ICV

Purpose:

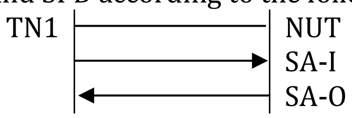
Verify that a NUT (End-Node) correctly processes an, otherwise valid, packet with an invalid ICV

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Transport
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)
	Data	“EchoData”

ICMP Echo Request with ESP 1

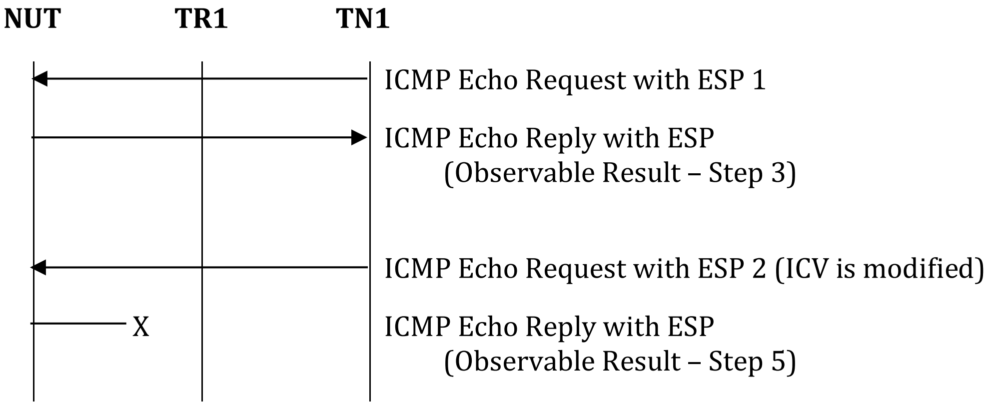
IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)
	Data	“EchoData”

ICMP Echo Reply with ESP

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	2
	Encrypted Data/ICV	SA-I
	ICV	aaaaaaaaaaaaaaaaaa......
ICMP	Type	128 (Echo Request)
	Data	“cracked”

ICMP Echo Request with ESP 2 (ICV is modified)

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits ICMP Echo Request with ESP 1	
3.	Observe the packets transmitted on Network 0	The NUT transmits ICMP Echo Reply with ESP
4.	TN1 transmits ICMP Echo Request with ESP 2 (ICV is modified)	
5.	Observe the packets transmitted on Network 0	The NUT never transmits ICMP Echo Reply with ESP

Possible Problems:

None

1.2. Ipsec/ESP Architecture (Tunnel Mode)

Ipsec.Conf.1.2.1. Tunnel Mode with End-Node

Purpose:

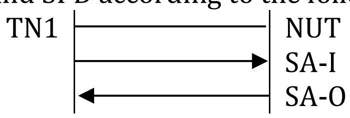
Verify that a NUT (End-Node) can build Ipsec tunnel mode with End-Node correctly.

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

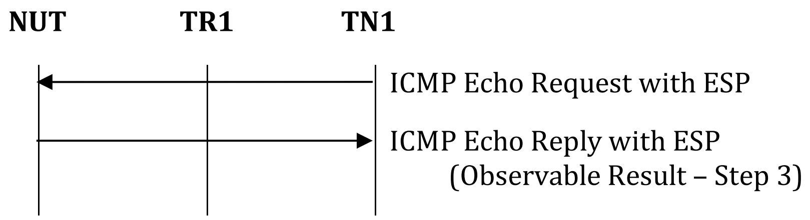
IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Network 0	The NUT transmits ICMP Echo Reply with ESP

Possible Problems:

None

Ipsec.Conf.1.2.2. Tunnel Mode with SGW

Purpose:

Verify that a NUT (End-Node) can build Ipsec tunnel mode with SGW correctly

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 2
- Configuration
  - Use Global Security Associations

Databases

Set NUT's SAD and SPD according to the following:

TH1 — TN1 | NUT
SA-I →
SA-O ←

Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Address	Network2
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

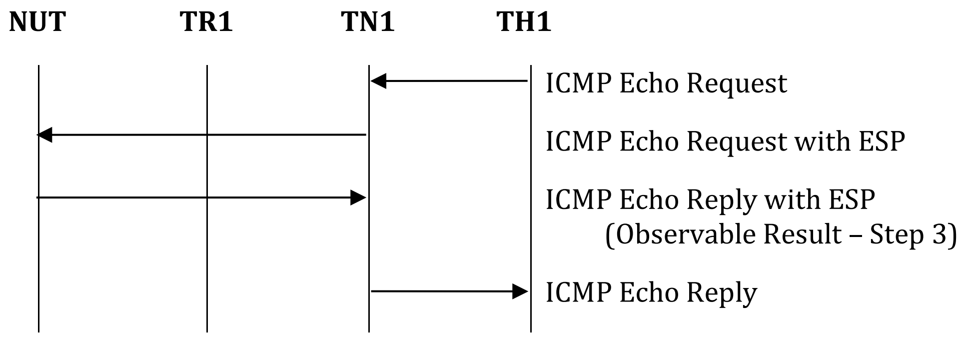
IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH1_Network2
	Destination Address	NUT_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Network0
	Destination Address	TH1_Network2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
2.	Initialize the NUT	
3.	TN1 transmits ICMP Echo Request with ESP	
4.	Observe the packets transmitted on Network 0	The NUT transmits ICMP Echo Reply with ESP

Possible Problems:

None

Ipsec.Conf.1.2.3. Tunnel Mode Select SPD

Purpose:

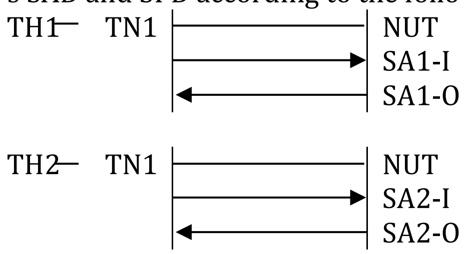
Verify that a NUT (End-Node) can select the correct SA and Policy between two hosts behind the same SGW

Initialization:

- Network Topology
  - o Connect the devices according to Common Topology 2
- Configuration
  - o Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	TH1_Network2
Local Traffic Selector	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Policy 2	
Peer	TN1_Network1
Mode	Tunnel
Remote Address	TH2_Network2
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA2-I
Outgoing SA	SA2-O

Packets:

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	*Dynamic1 or 0x1000*
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH1_Network2
	Destination Address	NUT_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP 1

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	*Dynamic2 or 0x2000*
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Network0
	Destination Address	TH1_Network2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP 1

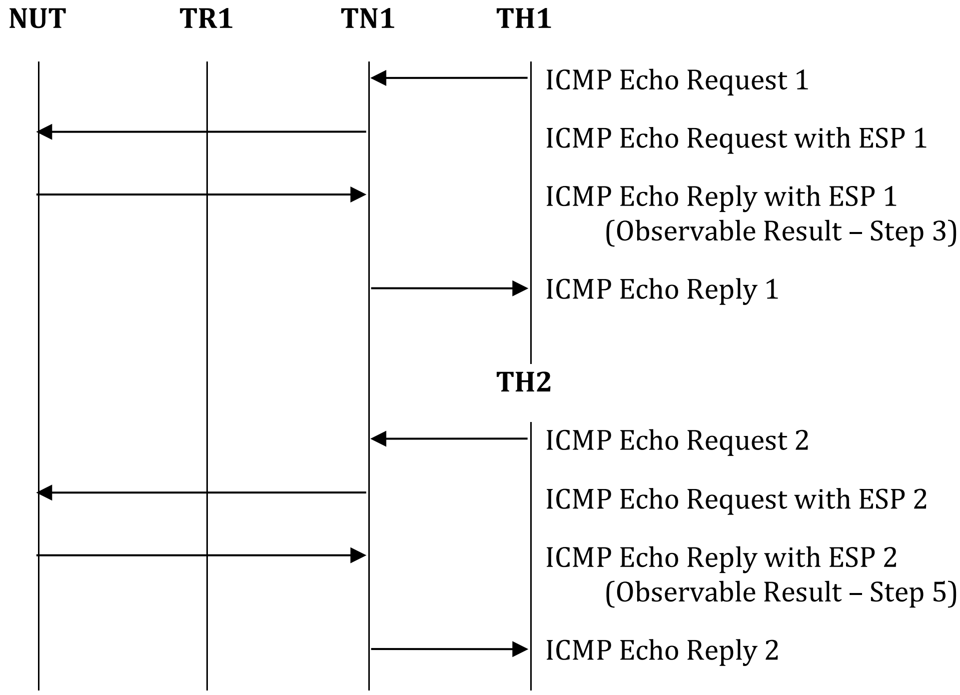
IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	*Dynamic3 or* 0x3000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH2_Network2
	Destination Address	NUT_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP 2

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	*Dynamic4 or* 0x4000
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Network0
	Destination Address	TH2_Network2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP 2

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits ICMP Echo Request with ESP 1	
3.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with ESP 1*
4.	TN1 transmits ICMP Echo Request with ESP 2	
5.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with ESP 2*

Possible Problems:

None

Ipsec.Conf.1.2.4. Tunnel Mode Padding

Purpose:

Verify that a NUT (End-Node) supports padding & padding byte handling

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 2
- Configuration
  - Use Global Security Associations

Databases

Set NUT's SAD and SPD according to the following:

TH1 — TN1 NUT

SA-I

SA-O

Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Address	Network2
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Part A: Tunnel Mode Padding

Packets:

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
	Padding	sequential
	Padding Length	7
IP Header	Source Address	TH1_Network2
	Destination Address	NUT_Network0
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP 1

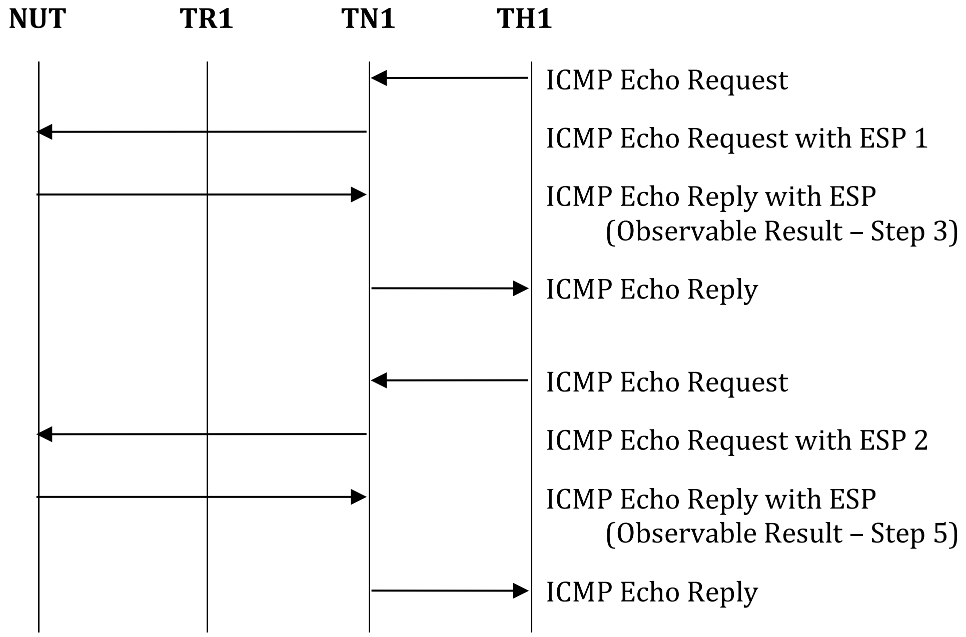
IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
	Padding	sequential
	Padding Length	255
IP Header	Source Address	TH1_Network2
	Destination Address	NUT_Network0
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP 2

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
	Padding Length	7+8n (0 <= n <= 31)
IP Header	Source Address	NUT_Network0
	Destination Address	TH1_Network2
ICMP	Type	129 (Echo Reply)
	Data Length	7

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits ICMP Echo Request with ESP 1	
3.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with ESP 1*
4.	TN1 transmits ICMP Echo Request with ESP 2	
5.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with ESP 2*

Part B: TFC enabled Tunnel Mode Padding

Packets:

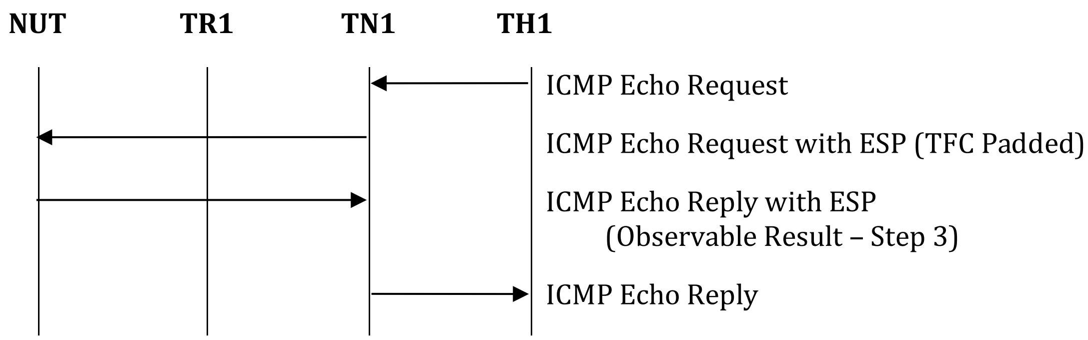
IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH1_Network2
	Destination Address	NUT_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP (TFC Padded)

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Network0
	Destination Address	TH1_Network2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
6.	Initialize the NUT	
7.	TN1 transmits *ICMP Echo Request with ESP (TFC Padded)*	
8.	Observe the packets transmitted on Network 0	The NUT transmits ICMP Echo Reply with ESP

Possible Problems:

None

Ipsec.Conf.1.2.5. Tunnel Mode Fragmentation

Purpose:

Verify that a NUT can reassemble/fragment packets correctly inside ESP Tunnel

Initialization:

- Network Topology
    - Connect the devices according to Common Topology 2
- Configuration
    - Use Global Security Associations

Databases

Set NUT's SAD and SPD according to the following:

TH1 — TN1 NUT
SA-I
SA-O

Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Address	Network2
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

IP Header	Source Address	TH1_Network2
	Destination Address	NUT_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH1_Network2
	Destination Address	NUT_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Network0
	Destination Address	TH1_Network2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TH1_Network2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

IP Header	Source Address	TH1_Network2
	Destination Address	NUT_Network0
	Payload Length	*1stPL*(=MTU-40) (e.g., 1240)
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request 1

IP Header	Source Address	TH1_Network2
	Destination Address	NUT_Network0
	Payload Length	*2ndPL*(=1476-1stPL)
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of *ICMP Echo Request*

Fragmented ICMP Echo Request 2

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH1_Network2
	Destination Address	NUT_Network0
	Payload Length	*1stPL*
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP 1

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH1_Network2
	Destination Address	NUT_Network0
	Payload Length	*2ndPL*
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of *ICMP Echo Request*

Fragmented ICMP Echo Request with ESP 2

IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_NETWORK2
	Destination Address	NUT_Network0
ICMP	Type	2 (Packet Too Big)
	MTU	1280 <= n <= 1430 (e.g., 1280)
	Data	1232Byte of *ICMP Echo Reply B*

ICMP Packet Too Big with ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Network0
	Destination Address	TH1_Network2
	Payload Length	*1stPL*
Fragment	Offset	0
	More Flag	1
ICMP	Type	129 (Echo Reply)

Fragmented ICMP Echo Reply with ESP 1

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Network0
	Destination Address	TH1_Network2
	Payload Length	*2ndPL*
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of *ICMP Echo Reply*

Fragmented ICMP Echo Reply with ESP 2

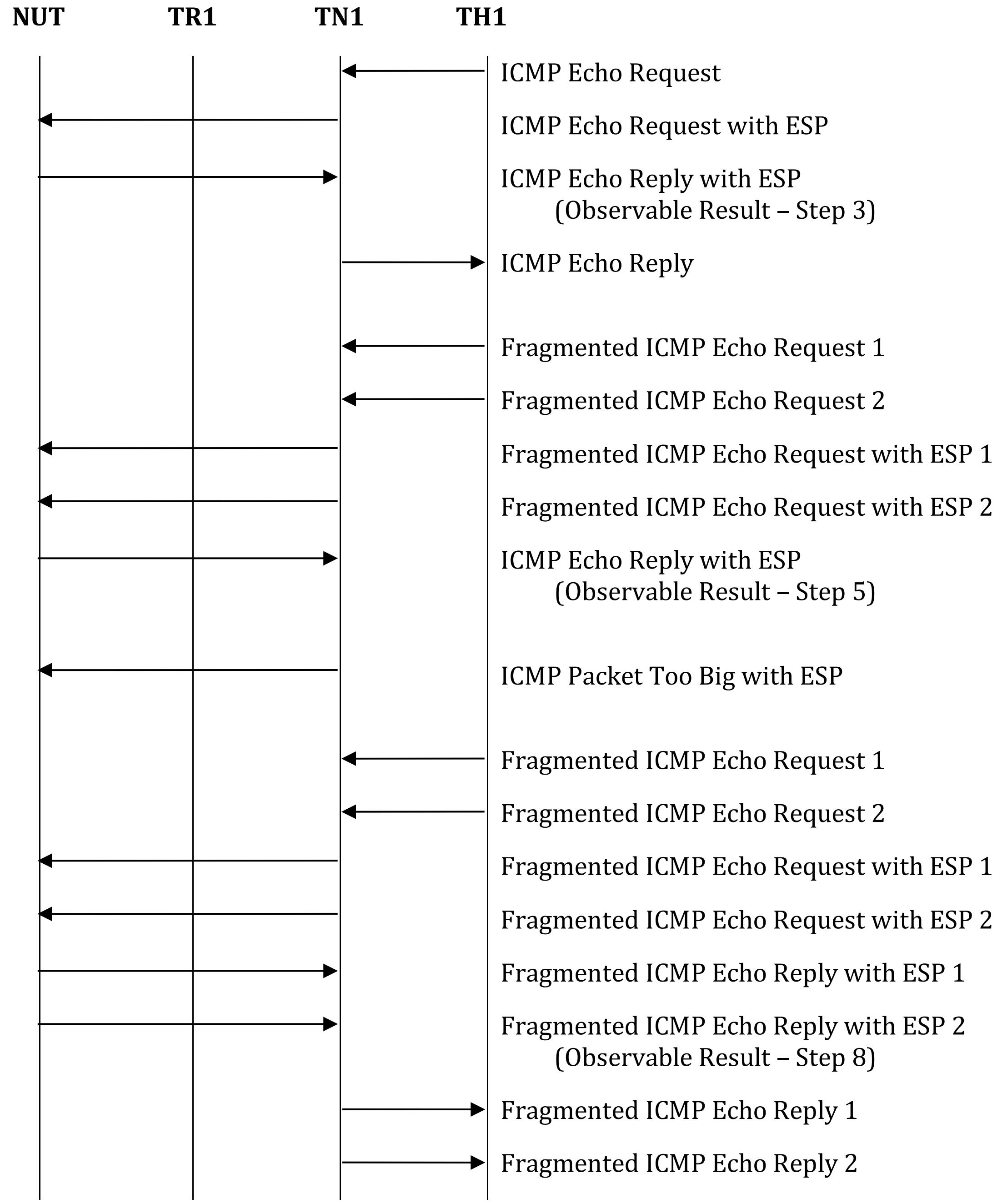
IP Header	Source Address	NUT_Network0
	Destination Address	TH1_Network2
	Payload Length	*1stPL*(=MTU-40) (e.g., 1240)
Fragment	Offset	0
	More Flag	1
ICMP	Type	129 (Echo Reply)

Fragmented ICMP Echo Reply 1

IP Header	Source Address	NUT_Network0
	Destination Address	TH1_Network2
	Payload Length	*2ndPL*(=1476-1stPL)
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of *ICMP Echo Reply*

Fragmented ICMP Echo Reply 2

Procedure:


NUT
TR1
TN1
TH1
ICMP Echo Request
ICMP Echo Request with ESP
ICMP Echo Reply with ESP
(Observable Result – Step 3)
ICMP Echo Reply
Fragmented ICMP Echo Request 1
Fragmented ICMP Echo Request 2
Fragmented ICMP Echo Request with ESP 1
Fragmented ICMP Echo Request with ESP 2
ICMP Echo Reply with ESP
(Observable Result – Step 5)
ICMP Packet Too Big with ESP
Fragmented ICMP Echo Request 1
Fragmented ICMP Echo Request 2
Fragmented ICMP Echo Request with ESP 1
Fragmented ICMP Echo Request with ESP 2
Fragmented ICMP Echo Reply with ESP 1
Fragmented ICMP Echo Reply with ESP 2
(Observable Result – Step 8)
Fragmented ICMP Echo Reply 1
Fragmented ICMP Echo Reply 2

Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 sends *ICMP Echo Request with ESP* from TH1 to NUT	
3.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Reply with ESP* to TH1
4.	TN1 sends *Fragmented ICMP Echo Request with ESP 1* and *Fragmented ICMP Echo Request with ESP 2* from TH1 to the NUT	
5.	Observe the packets transmitted on Network 0	The NUT reassembles ICMP Echo Request and transmits fully assembled *ICMP Echo Reply with ESP* to TH1
6.	TN1 sends *ICMP Packet Too Big Message with ESP* to the NUT	
7.	TN1 sends *ICMP Echo Request with ESP 1* and *ICMP Echo Request with ESP 2* from TH1 to the NUT	
8.	Observe the packets transmitted on Network 0	The NUT reassembles ICMP Echo Request and transmits *Fragmented ICMP Echo Reply with ESP 1* and *Fragmented ICMP Echo Reply with ESP 2* to TH1

Possible Problems:

None

Section 2: SGW Test

This Chapter describes the test specification for SGW.
The test specification consists of 2 parts. One is regarding “Ipsec Architecture” and another part is regarding to “Encryption and Integrity Algorithms”.

2.1. Ipsec/ESP Architecture

Scope:

Following tests focus on Ipsec Architecture for SGW devices.

Overview:

Tests in this section verify that a node properly process and transmit based on the Security Policy Database and Security Association Database.

Ipsec.Conf.2.1.1. Select SPD (2 SGW Peers)

Purpose:

Verify that a NUT (SGW) selects appropriate SPD

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 4
- Configuration
  - Use Global Security Associations

Databases

Set NUT's SAD and SPD according to the following:

TH3_Network3 — TN1 NUT — TH1_Network0
SA1-I
SA1-O

TH4_Network4 — TN2 NUT — TH1_Network0
SA2-I
SA2-O

Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	Network3
Local Traffic Selector	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Policy 2	
Peer	TN2_Network1
Mode	Tunnel
Remote Address	Network4
Local Address	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA2-I
Outgoing SA	SA2-O

Packets

IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 1

IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA1-I
IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA1-I's ESP

IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply 1

IP Header	Source Address	NUT_Network1
	Destination Address	TN1_Network2
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA1-O
IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA1-O's ESP

IP Header	Source Address	TH4_Network4
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 2

IP Header	Source Address	TN2_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic3 or 0x3000
	Sequence	1
	Encrypted Data/ICV	SA2-I
IP Header	Source Address	TH4_Network4
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA2-I's ESP

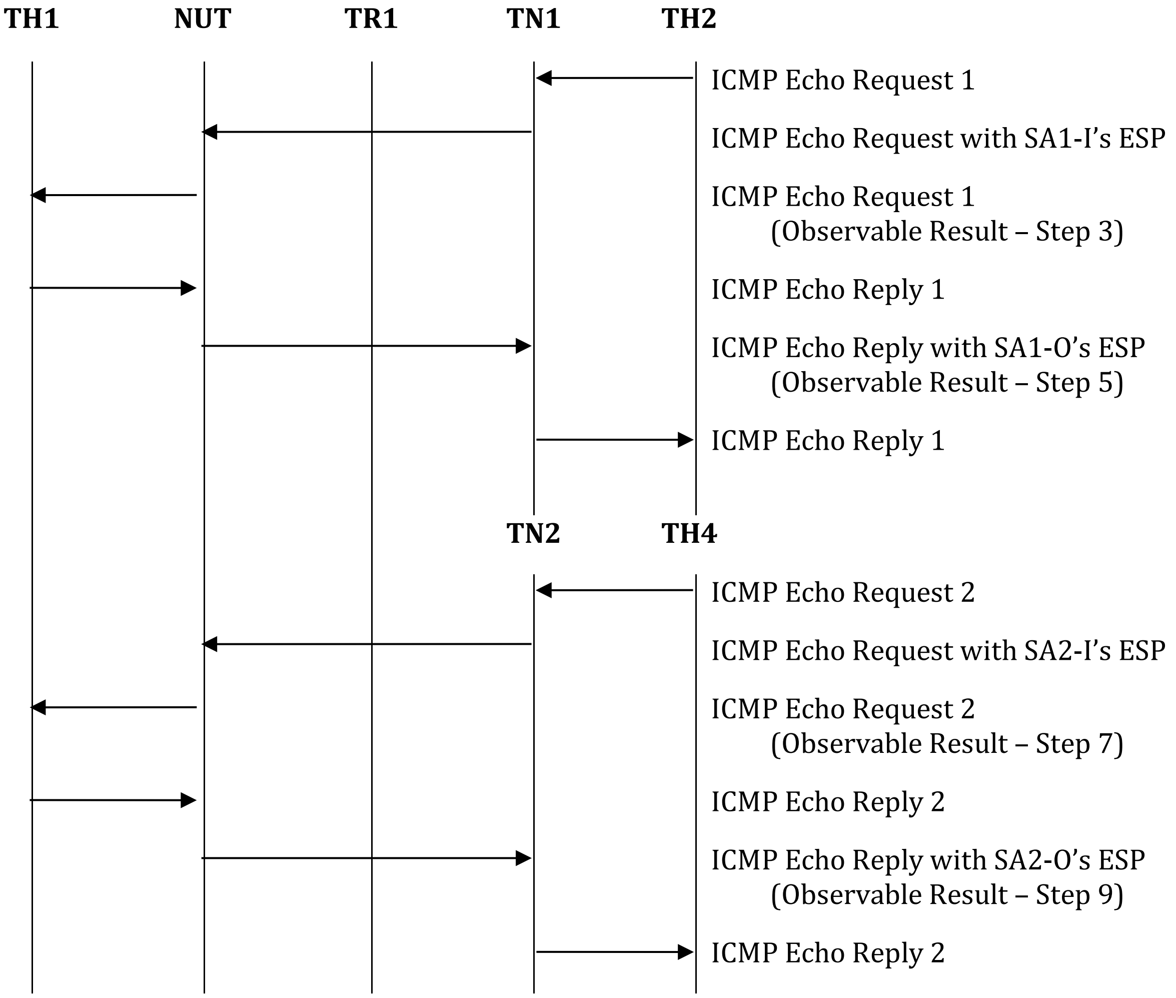
IP Header	Source Address	TH1_Network0
	Destination Address	TH4_Network4
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply 2

IP Header	Source Address	NUT_Network1
	Destination Address	TN2_Network2
ESP	SPI	Dynamic4 or 0x4000
	Sequence	1
	Encrypted Data/ICV	SA2-O
IP Header	Source Address	TH1_Network0
	Destination Address	TH4_Network4
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA2-O's ESP

Procedure:


TH1
NUT
TR1
TN1
TH2
ICMP Echo Request 1
ICMP Echo Request with SA1-I's ESP
ICMP Echo Request 1
(Observable Result – Step 3)
ICMP Echo Reply 1
ICMP Echo Reply with SA1-O's ESP
(Observable Result – Step 5)
ICMP Echo Reply 1
TN2
TH4
ICMP Echo Request 2
ICMP Echo Request with SA2-I's ESP
ICMP Echo Request 2
(Observable Result – Step 7)
ICMP Echo Reply 2
ICMP Echo Reply with SA2-O's ESP
(Observable Result – Step 9)
ICMP Echo Reply 2

Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits *ICMP Echo Request with SA1-I's ESP* (originally from TH2)	
3.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request 1*
4.	TH1 sends *ICMP Echo Reply 1*	
5.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Reply with SA1-O's ESP*
6.	TN2 transmits *ICMP Echo Request with SA2-I's ESP* (originally from TH4)	
7.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request 2*
8.	TH1 sends *ICMP Echo Reply 2*	
9.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Reply with SA2-O's ESP*

Possible Problems:

None

Ipsec.Conf.2.1.2. Select SPD (2 Hosts behind same Peer)

Purpose:

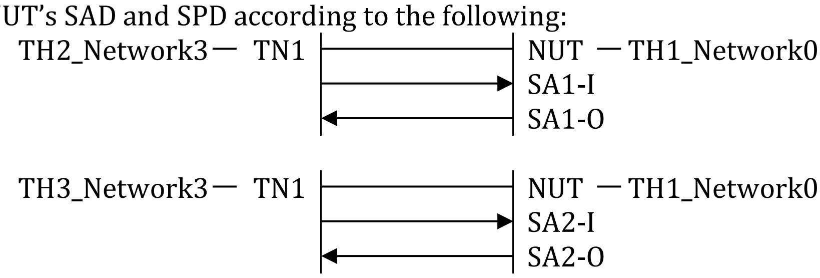
Verify that a NUT (SGW) selects appropriate SPD, for 2 Hosts behind 1 SGW

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 4
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	TH2_Network3
Local Traffic Selector	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Policy 2	
Peer	TN2_Network1
Mode	Tunnel
Remote Address	TH3_Network3
Local Address	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA2-I
Outgoing SA	SA2-O

Packets:

IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 1

IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA1-I
IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA1-I's ESP

IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply 1

IP Header	Source Address	NUT_Network1
	Destination Address	TN1_Network2
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA1-O
IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA1-O's ESP

IP Header	Source Address	TH3_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 2

IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic3 or 0x3000
	Sequence	1
	Encrypted Data/ICV	SA2-I
IP Header	Source Address	TH3_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA2-I's ESP

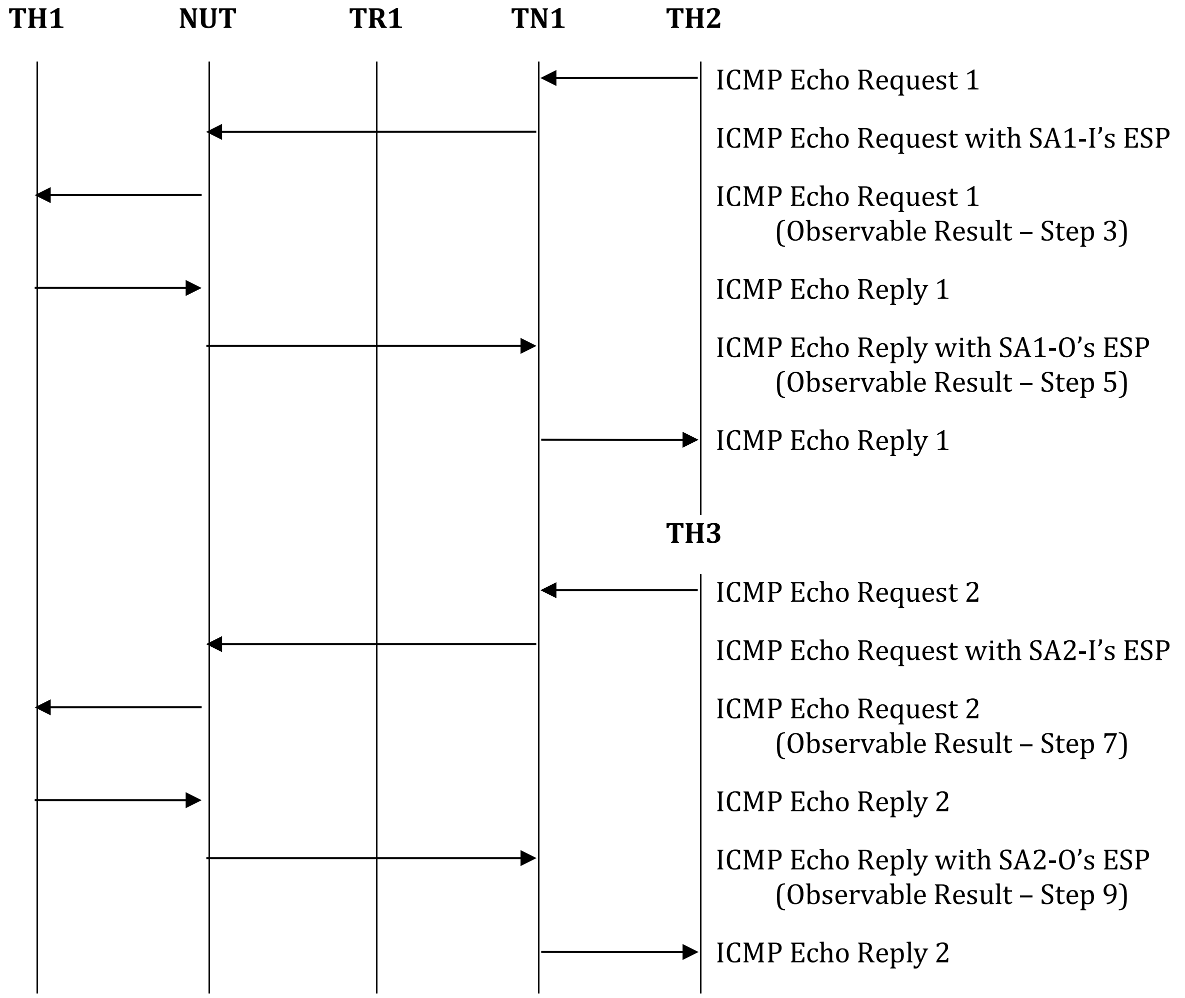
IP Header	Source Address	TH1_Network0
	Destination Address	TH3_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply 2

IP Header	Source Address	NUT_Network1
	Destination Address	TN1_Network2
ESP	SPI	Dynamic4 or 0x4000
	Sequence	1
	Encrypted Data/ICV	SA2-O
IP Header	Source Address	TH1_Network0
	Destination Address	TH3_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA2-O's ESP

Procedure:



TH1
NUT
TR1
TN1
TH2
ICMP Echo Request 1
ICMP Echo Request with SA1-I's ESP
ICMP Echo Request 1
(Observable Result – Step 3)
ICMP Echo Reply 1
ICMP Echo Reply with SA1-O's ESP
(Observable Result – Step 5)
ICMP Echo Reply 1
TH3
ICMP Echo Request 2
ICMP Echo Request with SA2-I's ESP
ICMP Echo Request 2
(Observable Result – Step 7)
ICMP Echo Reply 2
ICMP Echo Reply with SA2-O's ESP
(Observable Result – Step 9)
ICMP Echo Reply 2

Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits *ICMP Echo Request with SA1-I's ESP* (originally from TH2)	
3.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request 1*
4.	TH1 sends *ICMP Echo Reply 1*	
5.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Reply with SA1-O's ESP*
6.	TN1 sends *ICMP Echo Request with SA2-I's ESP* (originally from TH3)	
7.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request 2*
8.	TH1 sends *ICMP Echo Reply 2*	
9.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Reply with SA2-O's ESP*

Possible Problems:

None

Ipsec.Conf.2.1.3. Sequence Number Increment

Purpose:

Verify that a NUT (SGW) increases sequence number correctly, starting with 1.

Initialization:

- Network Topology
  - o Connect the devices according to Common Topology 4
- Configuration
  - o Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:

TH2_Network3 — TN1 ―――――― NUT — TH1_Network0

SA1-I →

← SA1-O

Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	Network3
Local Traffic Selector	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

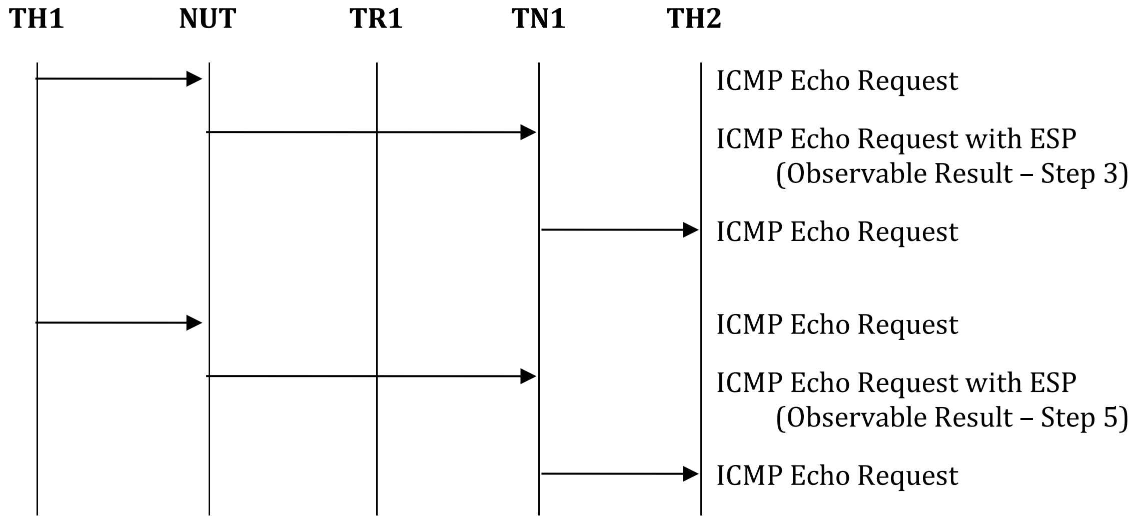
IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	NUT_Network1
	Destination Address	TN1_Network2
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1st = 1, 2nd = 2
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TH1 sends *ICMP Echo Request*	
3.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits an *ICMP Echo Request with ESP* with an ESP Sequence number of 1
4.	TH1 sends *ICMP Echo Request*	
5.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits an *ICMP Echo Request with ESP* with an ESP Sequence number of 2

Possible Problems:

None

Ipsec.Conf.2.1.4. Packet Too Big Transmission

Purpose:

Verify that a NUT (SGW) transmits the ICMP Error Message (Packet Too Big) correctly

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 4
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:

TH2_Network3 — TN1 —— NUT — TH1_Network0

TN1 → NUT: SA1-I

NUT → TN1: SA1-O

Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	Network3
Local Traffic Selector	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
	Payload Length	1460
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	NUT_Network0
	Destination Address	TH1_Network0
ICMP	Type	2 (Packet Too Big)
	MTU	1280 <= n <= 1430 (e.g., 1280)
	Data	1232Byte of *ICMP Echo Request*

ICMP Error Message (Packet Too Big)

IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
	Payload Length	*1stPL*(=MTU-40) (e.g., 1240)
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request 1

IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
	Payload Length	*2ndPL*(=1476-1stPL)
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of *ICMP Echo Request*

Fragmented ICMP Echo Request 2

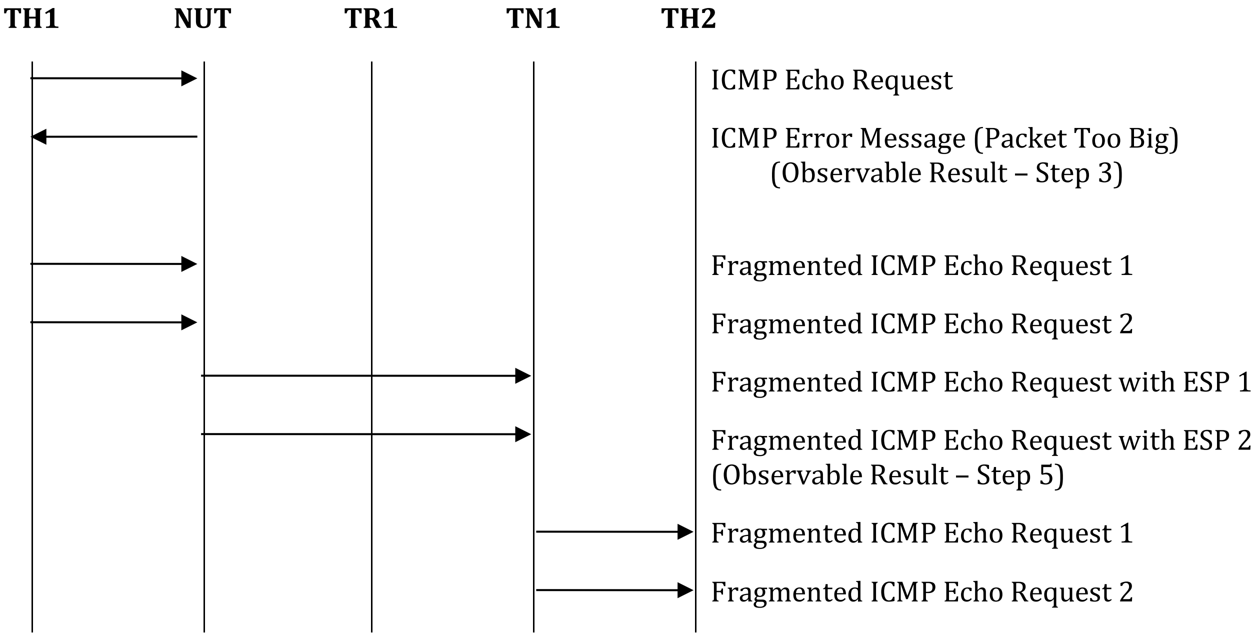
IP Header	Source Address	NUT_Network1
	Destination Address	TN1_Network2
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
	Payload Length	*1stPL*
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP 1

IP Header	Source Address	NUT_Network1
	Destination Address	TN1_Network2
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
	Payload Length	*2ndPL*
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of *ICMP Echo Request*

Fragmented ICMP Echo Request with ESP 2

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TH1 sends *ICMP Echo Request*	
3.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Error Message (Packet Too Big)*
4.	TH1 sends *Fragmented ICMP Echo Request 1* and *Fragmented ICMP Echo Request 2*	
5.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *Fragmented ICMP Echo Request with ESP 1* and *Fragmented ICMP Echo Request with ESP 2*

Possible Problems:

None

Ipsec.Conf.2.1.5. Packet Too Big Forwarding

Purpose:

Verify that a NUT (SGW) forwards the ICMP Error Message (Packet Too Big) correctly when the original Host cannot be determined

Initialization:

- Network Topology
    - Connect the devices according to Common Topology 4
- Configuration
    - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:

TH2_Network3 — TN1 ——— NUT — TH1_Network0

TN1 → NUT: SA1-I

NUT → TN1: SA1-O

Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	Network3
Local Traffic Selector	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
	Payload Length	1360
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	NUT_Network1
	Destination Address	TN1_Network2
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
	Payload Length	1360
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	TR1_Network2
	Destination Address	NUT_Network1
ICMP	Type	2 (Packet Too Big)
	MTU	1356
	Data	1232Byte of *ICMP Echo Request*

ICMP Error Message to NUT (Packet Too Big)

IP Header	Source Address	TR1_Network2 or NUT_Network1
	Destination Address	TH1_Network0
ICMP	Type	2 (Packet Too Big)
	MTU	1280 – 1286
	Data	1232Byte of *ICMP Echo Request*

ICMP Error Message to TH1 (Packet Too Big)

IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
	Payload Length	1240
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request 1

IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
	Payload Length	136
Fragment	Offset	154
	More Flag	0

Data	Data	Rest of *ICMP Echo Request*

Fragmented ICMP Echo Request 2

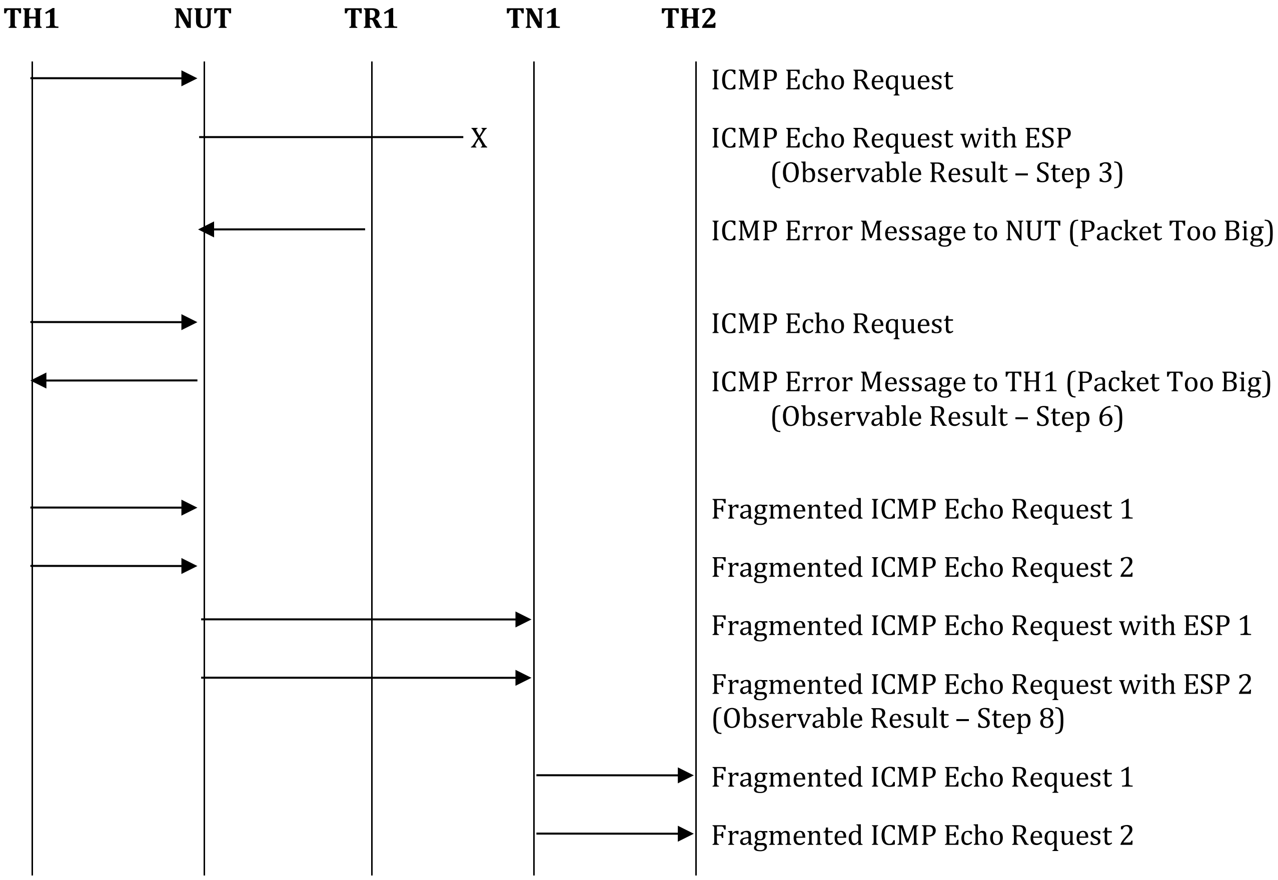
IP Header	Source Address	NUT_Network1
	Destination Address	TN1_Network2
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
	Payload Length	1240
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP 1

IP Header	Source Address	NUT_Network1
	Destination Address	TN1_Network2
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
	Payload Length	136
Fragment	Offset	154
	More Flag	0
Data	Data	Rest of *ICMP Echo Request*

Fragmented ICMP Echo Request with ESP 2

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TH1 sends *ICMP Echo Request*	
3.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request with ESP*
4.	TR1 sends *ICMP Error Message to NUT (Packet Too Big)*	
5.	TH1 sends *ICMP Echo Request*	The NUT transmits *Fragmented ICMP Echo Request with ESP 1* and *Fragmented ICMP Echo Request with ESP 2*
6.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Error Message to TH1*

		(Packet Too Big)
7.	TH1 sends *Fragmented ICMP Echo Request 1 and Fragmented ICMP Echo Request 2*	
8.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *Fragmented ICMP Echo Request with ESP 1* and *Fragmented ICMP Echo Request with ESP 2*

Possible Problems:

The NUT (SGW) may choose to process the ICMPv6 Packet Too Big PMTU information on the ciphertext side of the interface. In this case, the NUT will not generate and send a Packet Too Big Message to TH1, but will instead fragment Ipv6 Packets from TH1 after tunneling and applying ESP. TH1 will continue to transmit whole packets. See RFC 4301 Section 2.1.

Ipsec.Conf.2.1.6. Receipt of No Next Header

Purpose:

Verify that a NUT (SGW) can process the dummy packet (the protocol value 59) correctly.

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 4
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:

TH2_Network3 — TN1 ———— NUT — TH1_Network0

TN1 → NUT: SA1-I

NUT → TN1: SA1-O

Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	Network3
Local Traffic Selector	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

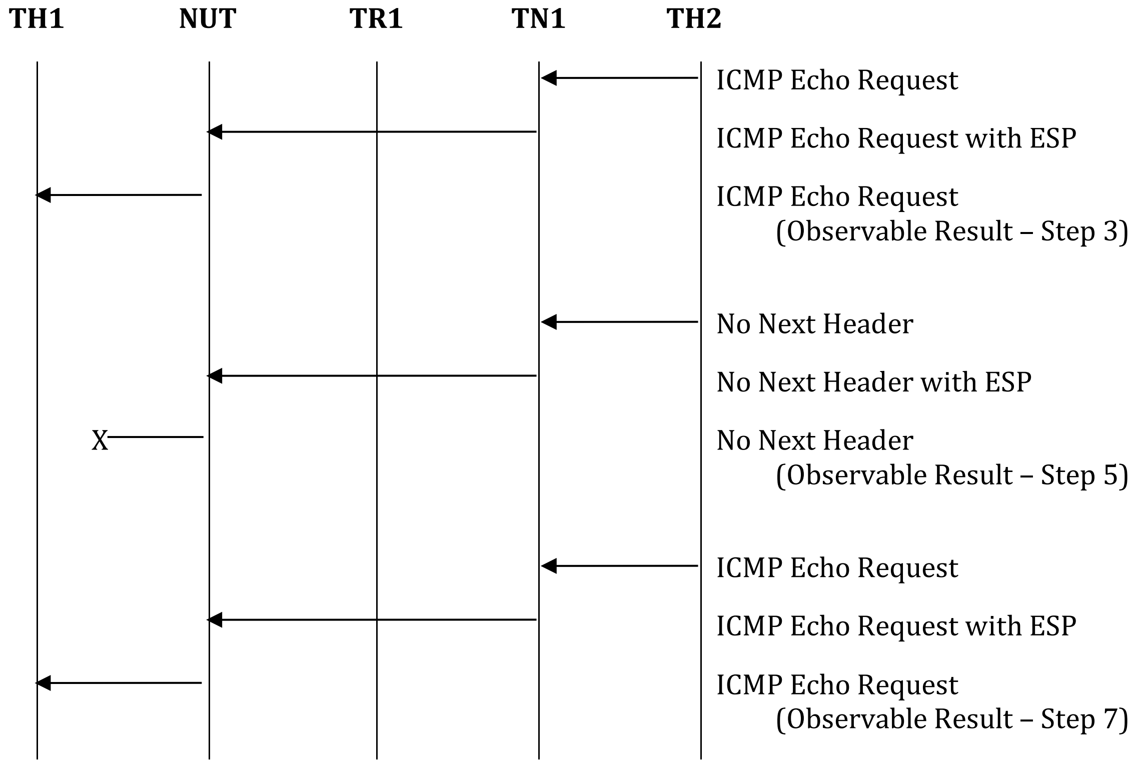
ICMP Echo Request with ESP

IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
	Next Header	no next header (59)
Upper Layer	Data	*See below*

No Next Header with ESP

Part A: No Next Header without TFC Padding	empty
Part B: No Next Header with TFC Padding	random bytes

Procedure:



Part A: No Next Header
Part B: No Next Header with TFC Padding
Use below steps for each part.

Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 sends ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request*
4.	TN1 sends No Next Header with ESP	
5.	The ESP sequence number must be 1 greater than the packet transmitted at step 2	
6.	Observe the packets transmitted on Network 0 and Network1	The NUT does not transmit any packets
7.	TN1 sends ICMP Echo Request with ESP	

8.	The ESP sequence number must be 1 greater than the packet transmitted at step 4	
9.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request*

Possible Problems:

None

Ipsec.Conf.2.1.7. Bypass Policy

Purpose:

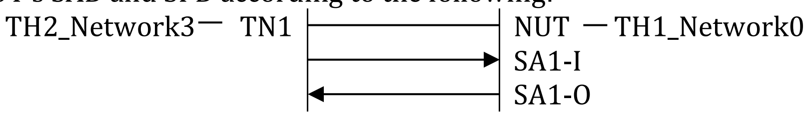
Verify that a NUT (End-Node) can utilize Bypass Policy

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 4
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	Network3
Local Traffic Selector	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Policy 2	
Peer	TN2_Network1
Mode	BYPASS
Remote Address	Network4
Local Address	NUT_Network0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 1

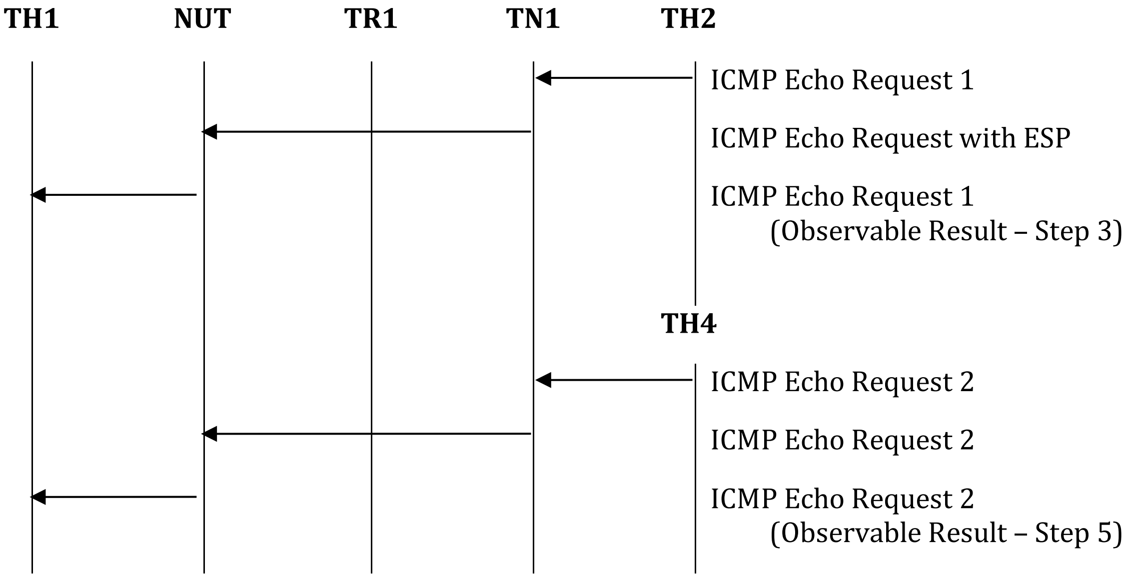
IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	TH4_Network4
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 2

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 sends ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request 1*
4.	TN2 forwards *ICMP Echo Request 2*	
5.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request 2*

Possible Problems:

Instead of specifying an address to bypass, a “bypass others by default” policy may also be enabled to discard address not covered by an Ipsec policy.

Ipsec.Conf.2.1.8. Discard Policy

Purpose:

Verify that a NUT (End-Node) can utilize Discard Policy

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 4
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:

TH2_Network3 — TN1 NUT — TH1_Network0
SA1-I
SA1-O

Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	Network3
Local Traffic Selector	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Policy 2	
Peer	TN2_Network1
Mode	DISCARD
Remote Address	Network4
Local Address	NUT_Network0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 1

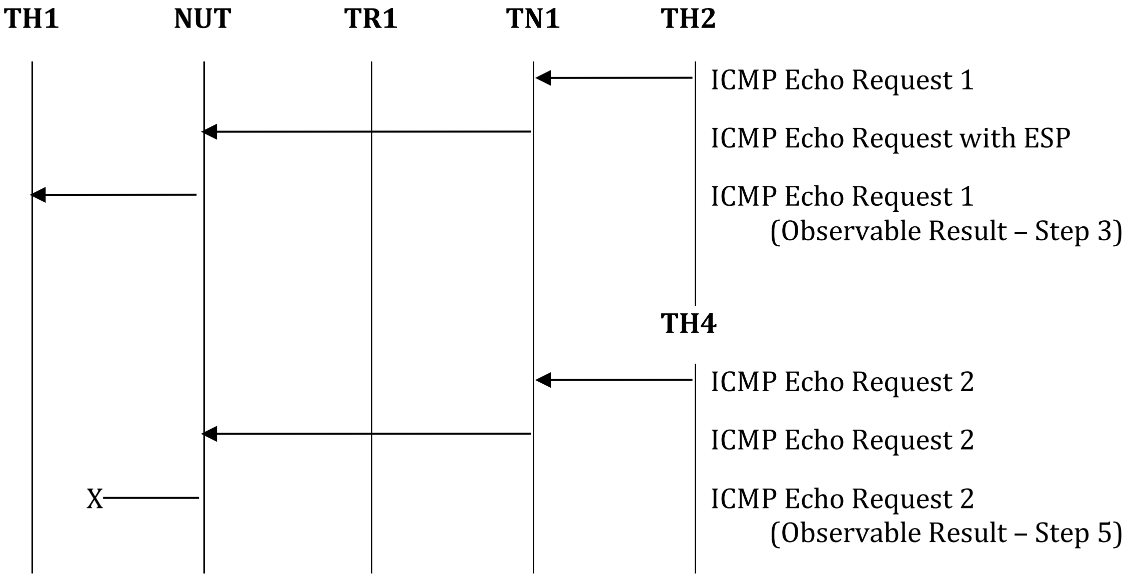
IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	TH4_Network4
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 2

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 sends ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request 1*
4.	TH4 sends ICMP Echo Request 2	
5.	Observe the packets transmitted on Network 0 and Network1	The NUT never transmits *ICMP Echo Request 2*

Possible Problems:

Instead of specifying an address to discard, a “discard others by default” policy may also be enabled to discard addresses not covered by an Ipsec policy.

Ipsec.Conf.2.1.9. Tunnel Mode Padding

Purpose:

Verify that a NUT (SGW) supports padding & padding byte handling

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 4
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:

TH2_Network3 — TN1 ——— NUT — TH1_Network0

TN1 ——→ NUT: SA1-I

TN1 ←—— NUT: SA1-O

Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	Network3
Local Traffic Selector	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Part A: Tunnel Mode Padding

Packets:

IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
	Padding	Sequential
	Padding Length	7+8n (0 <= n <= 31)
IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP

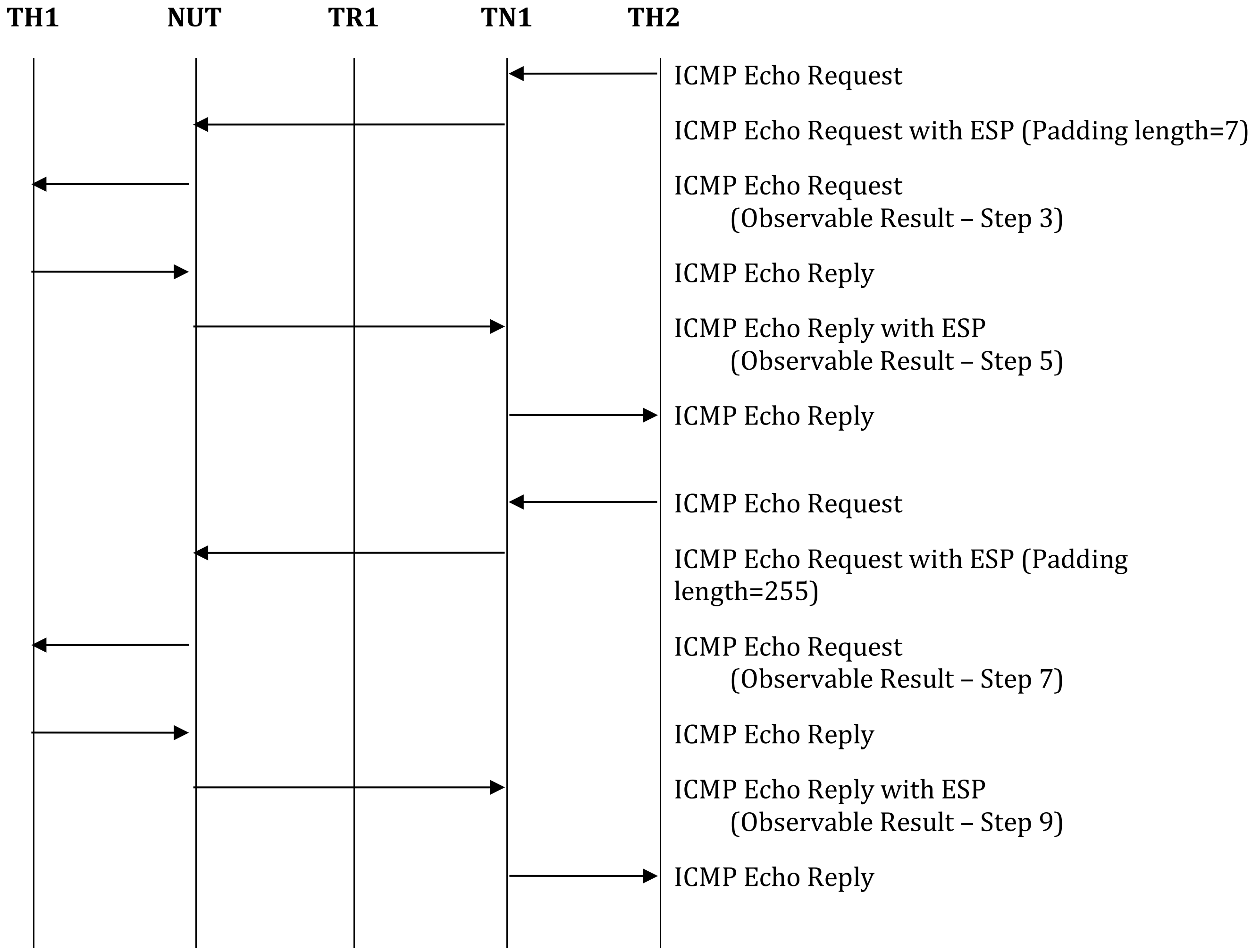
IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

IP Header	Source Address	NUT_Network1
	Destination Address	TN1_Network2
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
	Padding	Sequential
	Padding Length	7+8n (0 <= n <= 31)
IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
ICMP	Type	129 (Echo Reply)
	Data Length	7

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 sends ICMP Echo Request with ESP (Padding length=7)	
3.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request*
4.	TH1 sends ICMP Echo Reply	
5.	Observe the packet transmitted by NUT	The NUT transmits *ICMP Echo Reply with ESP*

6.	TN1 sends ICMP Echo Request with ESP (Padding length=255)	
7.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request*
8.	TH1 sends ICMP Echo Reply	
9.	Observe the packet transmitted by NUT	The NUT transmits *ICMP Echo Reply with ESP*

Part B: TFC enabled Tunnel Mode Padding

Packets:

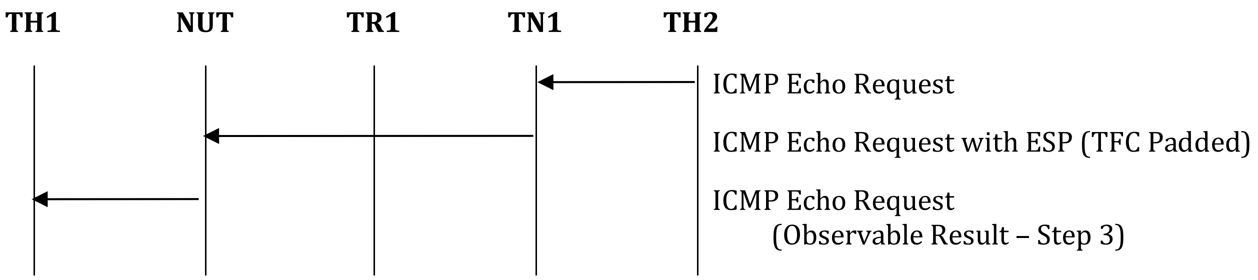
IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP (TFC Padded)

IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

Procedure:



Step	Action	Expected Result
10.	Initialize the NUT	
11.	TN1 sends *ICMP Echo Request with ESP (TFC Padded)*	

12.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request*

Possible Problems:

None

Ipsec.Conf.2.1.10. Invalid SPI

Purpose:

Verify that a NUT (End-Node) correctly processes an, otherwise valid, packet with an invalid SPI

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 4
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:

TH2_Network3 — TN1 — NUT — TH1_Network0
SA1-I
SA1-O

Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	Network3
Local Traffic Selector	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

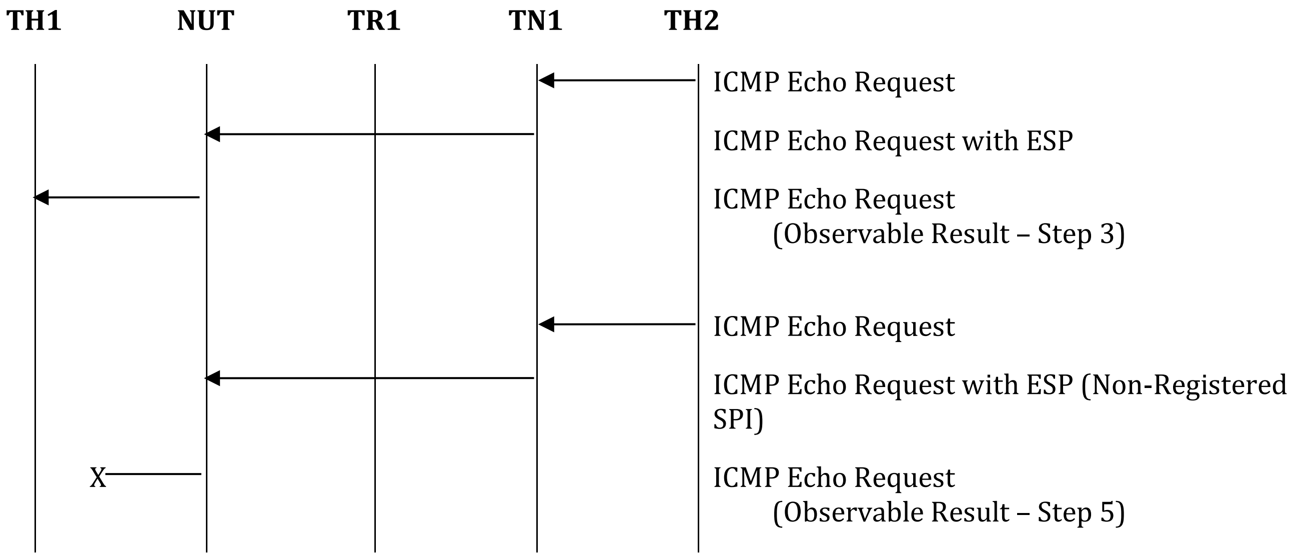
IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic1 or 0x1000
	Sequence Number	1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	0x9000 (different from SA-I's SPD)
	Sequence Number	1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP (Non-registered SPI)

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 sends *ICMP Echo Request with ESP*	
3.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request*
4.	TN1 sends *ICMP Echo Request with ESP (Non-registered SPI)*	
5.	Observe the packets transmitted on Network 0 and Network1	The NUT never transmits *ICMP Echo Request*

Possible Problems:

None

Ipsec.Conf.2.1.11. Invalid ICV

Purpose:

Verify that a NUT (End-Node) correctly processes an, otherwise valid, packet with an invalid SPI

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 4
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:

TH2_Network3 — TN1 —— NUT — TH1_Network0

SA1-I

SA1-O

Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	Network3
Local Traffic Selector	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)
	Data	"PadLen is zero"

ICMP Echo Request

IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

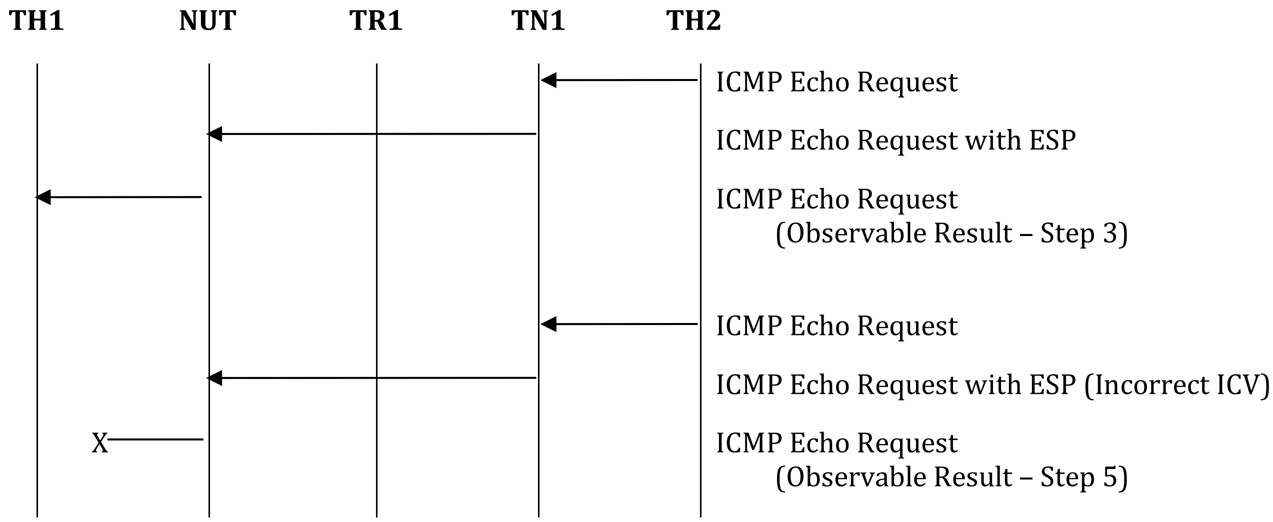
	Data	“PadLen is zero”

ICMP Echo Request with ESP

IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic1 or 0x1000
	Sequence	2
	Encrypted Data/ICV	SA-I
	ICV	aaaaaaaaa........
IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)
	Data	“cracked”

ICMP Echo Request with ESP (Incorrect ICV)

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 sends *ICMP Echo Request with ESP*	
3.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request*
4.	TN1 sends *ICMP Echo Request with ESP (Incorrect ICV)*	

5.	Observe the packets transmitted on Network 0 and Network1	The NUT never transmits *ICMP Echo Request*

Possible Problems:

None

Ipsec.Conf.2.1.12. Tunnel Mode with End-Node

Purpose:

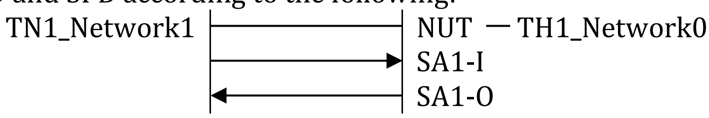
Verify that a NUT (SGW) can build Ipsec tunnel mode with End-Node correctly

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 3
- Configuration
  - Use Global Security Associations

Databases:

Set NUT's SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	TN1_Network1
Local Traffic Selector	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Network2
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	TN1_Network2
	Destination Address	TH1_Network0

ICMP	Type	128 (Echo Request)

ICMP Echo Request

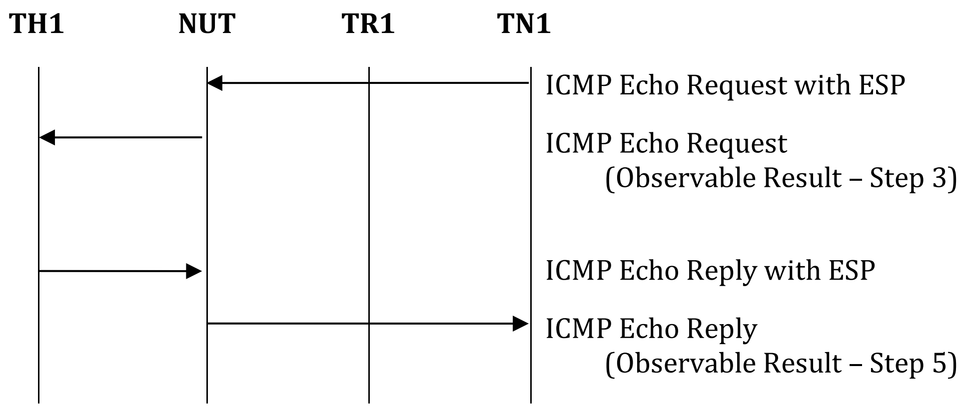
IP Header	Source Address	TH1_Network0
	Destination Address	TN1_Network2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

IP Header	Source Address	NUT_Network1
	Destination Address	TN1_Network2
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	TH1_Network0
	Destination Address	TN1_Network2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits *ICMP Echo Request with ESP*	
3.	Observe the packets transmitted on Network 0	The NUT transmits *ICMP Echo Request*
4.	TH1 transmits *ICMP Echo Reply*	
5.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Reply with ESP*

Possible Problems:

None

Section 3: ESP

This Chapter reviews the test cases for ESP, and in particular, the algorithms that use ESP.

Both End-Node and SGW devices should execute these test cases. The test cases are written to be agnostic towards device type. For each test, a given device should refer to the topology, packets, and detailed procedure, specific to its type.

4.1. ESP Algorithms

Scope:

The following test cases verify a device correctly utilizes ESP for different algorithms.

Overview:

Tests in this section verify that a node properly process and transmit based on the Algorithms and Security Policy Database and Security Association Database.

ESP Common Configurations

Algorithm List

The test case parts itemized below are used in this section, and are referred to by each test case.

Part	Encryption Algorithm	Integrity Algorithm	Keying
A	ENCR_NULL	AUTH_HMAC_SHA2_256_128	IKEv2 or Manual
B	ENCR_AES_CBC (128-bit)	AUTH_HMAC_SHA1_96	IKEv2 or Manual
C	ENCR_AES_CBC (128-bit)	AUTH_HMAC_SHA2_256_128	IKEv2 or Manual
D	ENCR_AES_CBC (256-bit)	AUTH_HMAC_SHA2_256_128	IKEv2 or Manual
E	ENCR_AES_CBC (256-bit)	AUTH_HMAC_SHA2_512_256	IKEv2 or Manual
F	ENCR_NULL	AUTH_AES_XCBC_96	IKEv2 or Manual
G	ENCR_AES_CCM_8 (128-bit)	N/A	IKEv2
H	ENCR_AES_GCM_16 (128-bit)	N/A	IKEv2
I	ENCR_AES_GCM_16 (256-bit)	N/A	IKEv2
J	ENCR_NULL_AUTH_AES_GMAC (128-bit)	N/A	IKEv2
K	ENCR_NULL_AUTH_AES_GMAC (256-bit)	N/A	IKEv2
L	ENCR_CHACHA20_POLY1305	N/A	IKEv2

Manual Key Settings

Part	SA	Direction	SPI		Keys
A	SA1-I	IN	0x1000	E	N/A
				A	ipv6readylogoph2ipsecsha2256in01
	SA1-O	OUT	0x2000	E	N/A
				A	ipv6readylogoph2ipsecsha2256out1
B	SA1-I	IN	0x1000	E	ipv6readaescin01
				A	ipv6readylogsha1in01
	SA1-O	OUT	0x2000	E	ipv6readaescout1
				A	ipv6readylogsha1out1
C	SA1-I	IN	0x1000	E	ipv6readaescin01
				A	ipv6readylogoph2ipsecsha2256in01
	SA1-O	OUT	0x2000	E	ipv6readaescout1
				A	ipv6readylogoph2ipsecsha2256out1
D	SA1-I	IN	0x1000	E	ipv6readylogoph2ipsecaesc256in01
				A	ipv6readylogoph2ipsecsha2256in01
	SA1-O	OUT	0x2000	E	ipv6readylogoph2ipsecaesc256out1
				A	ipv6readylogoph2ipsecsha2256out1
E	SA1-I	IN	0x1000	E	ipv6readylogoph2ipsecaesc256in01
				A	ipvsixreadylogophasetwoipsecconformancealghmacsha2fiveonetwoin01
	SA1-O	OUT	0x2000	E	ipv6readylogoph2ipsecaesc256out1
				A	ipvsixreadylogophasetwoipsecconformancealghmacsha2fiveonetwoout1

See appendix for notes regarding tests for which Manual Keys are disallowed.

IPsec.Conf.3.1.1. End-Node ESP Algorithms (Transport Mode)

Purpose:

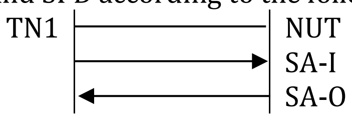
Verify that an End-Node device can correctly utilize various algorithms in Transport Mode

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - Use ESP Common Configurations combined with the below configurations
  - In addition, use the algorithms specified in each part, using Manual Keys only if IKEv2 is unsupported

Databases:

Set NUT's SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Transport
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

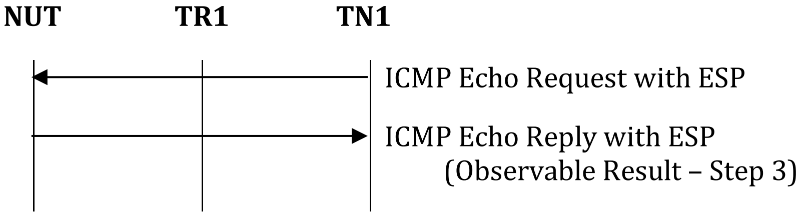
IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	*Dynamic1 or 0x1000*
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	*Dynamic2 or 0x2000*
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



All Parts: Algorithms

Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Network 0	The NUT transmits ICMP Echo Reply with ESP

Possible Problems:

None

IPsec.Conf.3.1.2. End-Node ESP Algorithms (Tunnel Mode)

Purpose:

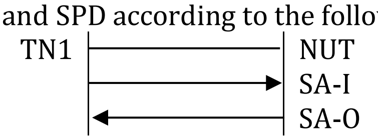
Verify that an End-Node device can correctly utilize various algorithms in Tunnel Mode

Initialization:

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - Use ESP Common Configurations combined with the below configurations
  - In addition, use the algorithms specified in each part, using Manual Keys only if IKEv2 is unsupported

Databases:

Set NUT's SAD and SPD according to the following:



Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Address	TN1_Network1
Local Address	NUT_Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

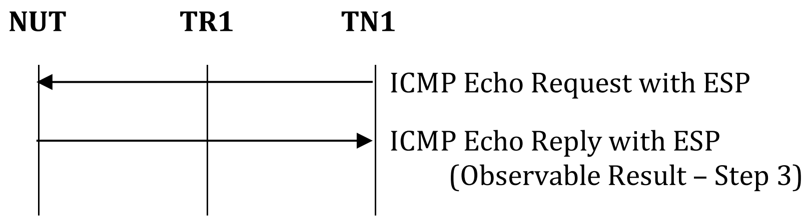
IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ESP	SPI	*Dynamic1 or 0x1000*
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Network1
	Destination Address	NUT_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ESP	SPI	*Dynamic2 or 0x2000*
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Network0
	Destination Address	TN1_Network1
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



All Parts: Algorithms

Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Network 0	The NUT transmits ICMP Echo Reply with ESP

Possible Problems:

None

IPsec.Conf.3.1.3. SGW ESP Algorithms

Purpose:

Verify that an SGW device can correctly utilize various algorithms

Initialization:

- Network Topology
    - Connect the devices according to Common Topology 4
- Configuration
    - Use ESP Common Configurations combined with the below configurations
    - In addition, use the algorithms specified in each part, using Manual Keys only if IKEv2 is unsupported

Databases:

Set NUT's SAD and SPD according to the following:

TH2_Network3— TN1 NUT — TH1_Network0
SA1-I
SA1-O

Policy 1	
Peer	TN1_Network1
Mode	Tunnel
Remote Traffic Selector	Network3
Local Traffic Selector	Network0
Protocol/Port	ANY/ANY
If using Manual Keys include:	
Incoming SA	SA1-I
Outgoing SA	SA1-O

Packets:

IP Header	Source Address	TN1_Network2
	Destination Address	NUT_Network1
ESP	SPI	Dynamic1 or 0x1000
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	TH2_Network3
	Destination Address	TH1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

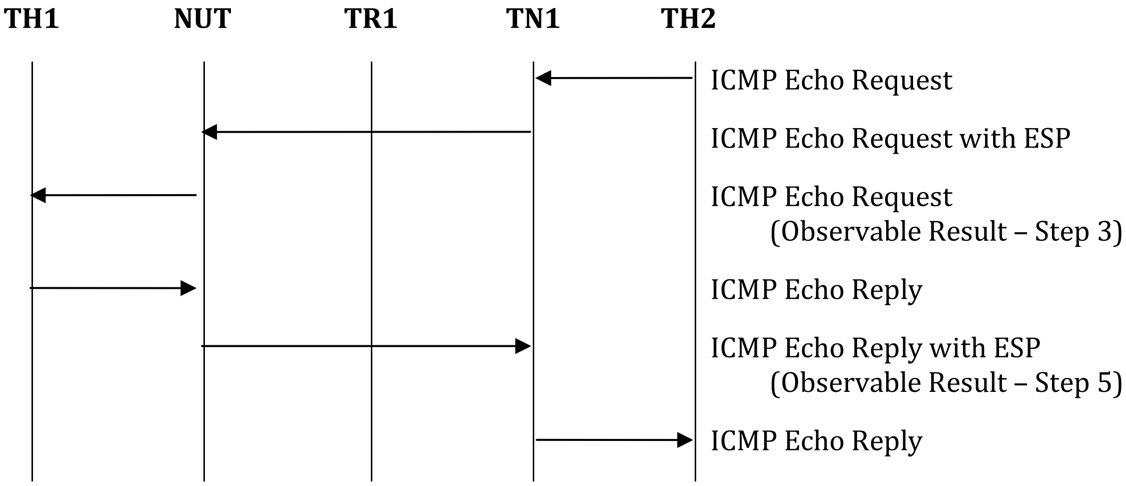
IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

IP Header	Source Address	NUT_Network1
	Destination Address	TN1_Network2
ESP	SPI	Dynamic2 or 0x2000
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	TH1_Network0
	Destination Address	TH2_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



All Parts: Algorithms

Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 transmits *ICMP Echo Request with ESP*	
3.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Request*
4.	TH1 transmits *ICMP Echo Reply*	
5.	Observe the packets transmitted on Network 0 and Network1	The NUT transmits *ICMP Echo Reply with ESP*

Possible Problems:

None

Appendix A: Annex-5.1.2 for the Passive Node

This appendix describes alternative methods to perform Test 5.1.2 on the passive node that doesn't have the application to send ICMPv6 Echo Request.

Using UDP application to invoke ICMPv6 Destination Unreachable (Port unreachable)

Requirements:

- Must respond to ICMPv6 Echo Request with ICMPv6 Echo Reply
- Must respond to UDP packet toward the closed port with ICMPv6 Destination Unreachable (Port unreachable)

Initialization:

Use common topology described as Fig.1

Set NUT's SAD and SPD according to the following:

```
                                              (passive node)
TH1 --------- transport ---------- NUT

        ------- spi=0x1000 ------> SA1-In    ICMPv6 Echo Request
        <------ spi=0x2000 ------- SA2-Out   ICMPv6 Echo Reply
        <------ spi=0x3000 ------- SA3-O     ICMPv6 Destination Unreachable
                                                  (Port unreachable)
```

- SA1-In

Security Association Database (SAD)

source address	TH1_Network1
destination address	NUT_Network0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD)

source address	TH1_Network1
destination address	NUT_Network0
upper spec	ICMPv6 Echo Request
direction	inbound
protocol	ESP
mode	transport

- SA2-Out

Security Association Database (SAD)

source address	NUT_Network0
destination address	TH1_Network1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout2
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out2

Security Policy Database (SPD)

source address	NUT_Network0
destination address	TH1_Network1
upper spec	ICMPv6 Echo Reply
direction	outbound
protocol	ESP
mode	transport

- SA3-O

Security Association Database (SAD)

source address	NUT_Network0
destination address	TH1_Network1
SPI	0x3000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout3
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out3

Security Policy Database (SPD)

source address	NUT_Network0
destination address	TH1_Network1
upper spec	ICMPv6 Destination Unreachable
direction	outbound
protocol	ESP
mode	transport

Packets:

ICMPv6 Echo Request with ESP1

IPv6	Source Address	TH1_Network1
	Destination Address	NUT_Network0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMPv6	Type	128 (Echo Request)

ICMPv6 Echo Reply with ESP2

IPv6	Source Address	NUT_Network0
	Destination Address	TH1_Network1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcout2
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out2
ICMPv6	Type	129 (Echo Reply)

UDP packet toward closed port

IPv6	Source Address	TH1_Network1
	Destination Address	NUT_Network0
UDP	Source Port	Any unused port on TH1
	Destination Port	Any closed port on NUT

ICMPv6 Destination Unreachable with ESP3

IPv6	Source Address	NUT_Network0
	Destination Address	TH1_Network1
ESP	SPI	0x3000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcout3
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out3
ICMPv6	Type	1 (Destination Unreachable)
	Code	4 (Port unreachable)

Procedure:

```
                              (passive node)
TH1_Network1                     NUT_Network0
     (TN)                            (NUT)
      |                               |
      |------ ciphertext ---->| ICMPv6 Echo Request with ESP1
      |<----- ciphertext -----| ICMPv6 Echo Reply with ESP2
      |                               |     (Observable Result #1)
      |                               |
      |------ plaintext ----->| UDP packet toward closed port
      |<----- ciphertext -----| ICMPv6 Destination Unreachable with ESP3
      |                               |     (Observable Result #2)
      |                               |
      V                               V
```

Part A (ADVANCED):

1. TH1_Network1 sends *"ICMPv6 Echo Request with ESP1"* to NUT_Network0
2. Observe the packet transmitted by NUT_Network0
3. TH1_Network1 sends *"UDP packet toward closed port"* to NUT_Network0
4. Observe the packet transmitted by NUT_Network0

Observable Results:

Part A:
Step-2 (Observable Result #1):
NUT_Network0 transmits *"ICMPv6 Echo Reply with ESP2"*
Step-4 (Observable Result #2):
NUT_Network0 transmits *"ICMPv6 Destination Unreachable with ESP3"*

Possible Problems:

None.

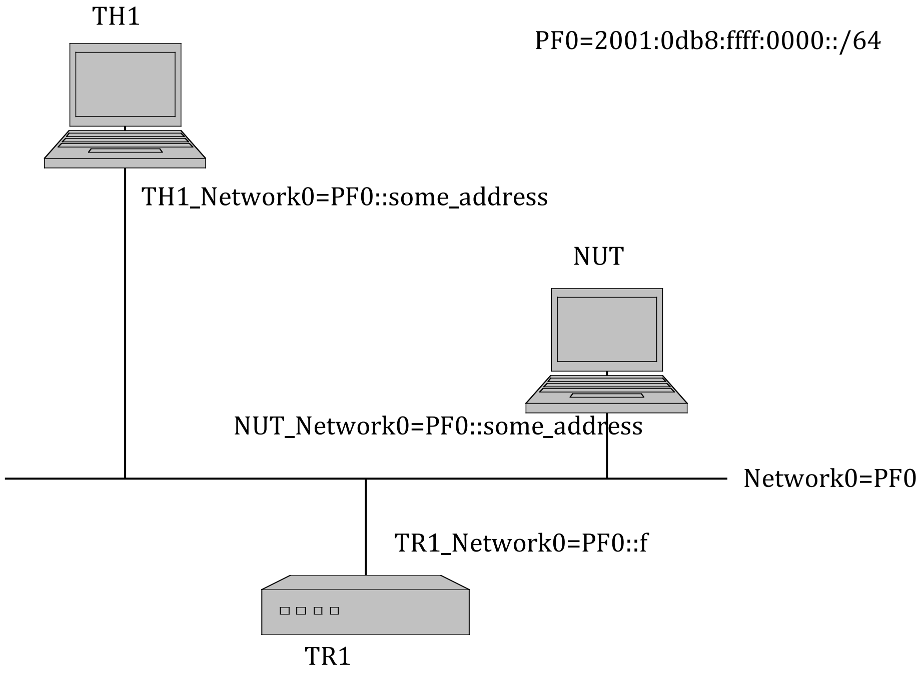
Invoking Neighbor Unreachability Detection

Requirements:

- Must respond to ICMPv6 Echo Request with ICMPv6 Echo Reply

Initialization:

Use following topology



Reboot NUT making sure it has cleared its neighbor cache. Allow time for all devices on Network 0 to perform Stateless Address Autoconfiguration and Duplicate Address Detection.

1. Set the global address (NUT_Network0) to NUT by RA if NUT is the Host. Otherwise set the global address (NUT_Network0) to NUT manually
2. Set MTU (1500 bytes for Network 0) to NUT by RA if NUT is the Host. Otherwise set MTU (1500 bytes for Network 0) to NUT manually.
3. Set NUT's SAD and SPD according to the following:

```
                                        (passive node)
TH1 --------- transport ---------- NUT

            ------- spi=0x1000 ------> SA1-In    ICMPv6 Echo Request
            <------ spi=0x2000 ------- SA2-Out   ICMPv6 Echo Reply
            <------ spi=0x3000 ------- SA3-O     ICMPv6 Neighbor Solicitation
            ------- spi=0x4000 ------> SA4-I     ICMPv6 Neighbor Advertisement
```

- SA1-In

Security Association Database (SAD)

source address	TH1_Network0
destination address	NUT_Network0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD)

source address	TH1_Network0
destination address	NUT_Network0
upper spec	ICMPv6 Echo Request
direction	inbound
protocol	ESP
mode	transport

- SA2-Out

Security Association Database (SAD)

source address	NUT_Network0
destination address	TH1_Network0
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout2
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out2

Security Policy Database (SPD)

source address	NUT_Network0
destination address	TH1_Network0
upper spec	ICMPv6 Echo Reply
direction	outbound
protocol	ESP
mode	transport

- SA3-O

Security Association Database (SAD)

source address	NUT_Network0
destination address	TH1_Network0
SPI	0x3000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout3
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out3

Security Policy Database (SPD)

source address	NUT_Network0
destination address	TH1_Network0
upper spec	ICMPv6 Neighbor Solicitation
direction	ipv6readylogo3descbcin01outbound
protocol	ESP
mode	transport

- SA4-I

Security Association Database (SAD)

source address	TH1_Network0
destination address	NUT_Network0
SPI	0x4000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin04
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in04

Security Policy Database (SPD)

source address	TH1_Network0
destination address	NUT_Network0
upper spec	ICMPv6 Neighbor Advertisement
direction	inbound
protocol	ESP
mode	transport

Packets:

ICMPv6 Neighbor Solicitation (multicast)

IPv6	Hop Limit	255
	Source Address	TH1_Network0
	Destination Address	NUT_Network0 (solicited-node multicast address)
ICMPv6	Type	135 (Neighbor Solicitation)
	Target Address	NUT_Network0
	Source Network-layer address Option Network-Layer Address: TH1_Network0 MAC address	

ICMPv6 Neighbor Advertisement

IPv6	Hop Limit	255
	Source Address	NUT_Network0
	Destination Address	TH1_Network0
ICMPv6	Type	136 (Neighbor Advertisement)
	R	false (if NUT is the Host) true (if NUT is the router)
	S	true
	O	true
	Target Address	NUT_Network0
	Target Network-layer address Option Network-Layer Address: NUT_Network0 MAC address	

ICMPv6 Echo Request with ESP1

IPv6	Source Address	TH1_Network0
	Destination Address	NUT_Network0
ESP	SPI	0x1000
	Sequence Number	1
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMPv6	Type	128 (Echo Request)

ICMPv6 Echo Reply with ESP2

IPv6	Source Address	NUT_Network0
	Destination Address	TH1_Network0
ESP	SPI	0x2000
	Sequence Number	1
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcout2
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out2
ICMPv6	Type	129 (Echo Reply)

ICMPv6 Neighbor Solicitation with ESP3

IPv6	Hop Limit	255

	Source Address	NUT_Network0
	Destination Address	TH1_Network0
ESP	SPI	0x3000
	Sequence Number	1
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcout3
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out3
ICMPv6	Type	135 (Neighbor Solicitation)
	Target Address	TH1_Network0
	Source Network-layer address Option Network-Layer Address: NUT_Network0 MAC address	

ICMPv6 Neighbor Advertisement with ESP4

IPv6	Hop Limit	255
	Source Address	TH1_Network0
	Destination Address	NUT_Network0
ESP	SPI	0x4000
	Sequence Number	1
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcin04
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in04
ICMPv6	Type	136 (Neighbor Advertisement)
	R	false
	S	true
	O	true
	Target Address	TH1_Network0
	Target Network-layer address Option Network-Layer Address: TH1_Network0 MAC address	

Procedure:

```
                                  (passive node)
TH1_Network0                        NUT_Network0
    (TN)                               (NUT)
     |                                   |
   [NONE]                              [NONE]
     |                                   |
[INCOMPLETE]                             |
     |                                   |
     |------ plaintext ----->|  ICMPv6 Neighbor Solicitation (multicast)
     |                                   |
     |                                [STALE]
     |                                   |
     |<----- plaintext ------|  ICMPv6 Neighbor Advertisement
     |                                   |     (Observable Result #1)
     |                                   |
[REACHABLE]                          [DELAY]
     |                                   |
     |------ ciphertext ---->|  ICMPv6 Echo Request with ESP1
     |<----- ciphertext -----|  ICMPv6 Echo Reply with ESP2
     |                                   |     (Observable Result #2)
     |                                   |
     |                                   * wait DELAY_FIRST_PROBE_TIME (5) seconds
     |                                   |
     |                                [PROBE]
     |                                   |
     |<----- ciphertext -----|  ICMPv6 Neighbor Solicitation with ESP3
     |                                   |     (Observable Result #3)
     |------ ciphertext ---->|  ICMPv6 Neighbor Advertisement with ESP4
     |                                   |
     |                              [REACHABLE]
     |                                   |
     V                                   V
```

Part A (ADVANCED):

1. TH1_Network0 sends *"ICMPv6 Neighbor Solicitation (multicast)"* to NUT_Network0
2. Observe the packet transmitted by NUT_Network0
3. TH1_Network0 sends *"ICMPv6 Echo Request with ESP1"* to NUT_Network0
4. Observe the packet transmitted by NUT_Network0
5. Observe the packet transmitted by NUT_Network0 for DELAY_FIRST_PROBE_TIME (5) seconds
6. TH1_Network0 sends *"ICMPv6 Neighbor Advertisement with ESP4"* to NUT_Network0

Observable Results:

Part A:

Step-2 (Observable Result #1):

NUT_Network0 transmits *"ICMPv6 Neighbor Advertisement"*

Step-4 (Observable Result #2):

NUT_Network0 transmits *"ICMPv6 Echo Reply with ESP2"*

Step-5 (Observable Result #3):

NUT_Network0 transmits *"ICMPv6 Neighbor Solicitation with ESP3"*

Possible Problems:

None

Appendix B: Manual Settings Disallowed

The below algorithms are inherently insecure when used with static keys. The quotes below reference the applicable sections describing this for each algorithm.

AES-CCM

According to RFC 4309, Section 2:

> AES CCM employs counter mode for encryption. As with any stream cipher, reuse of the same IV value with the same key is catastrophic. An IV collision immediately leaks information about the plaintext in both packets. For this reason, it is inappropriate to use this CCM with statically configured keys. Extraordinary measures would be needed to prevent reuse of an IV value with the static key across power cycles. To be safe, implementations MUST use fresh keys with AES CCM. The Internet Key Exchange (IKE) [IKE] protocol or IKEv2 [IKEv2] can be used to establish fresh keys.

Therefore, Manual Keys MUST NOT be used with this algorithm, and devices that do not support IKEv2 will FAIL this test case.

AES-GCM

According to RFC4106, Section 2:

> Because reusing an nonce/key combination destroys the security guarantees of AES-GCM mode, it can be difficult to use this mode securely when using statically configured keys. For safety's sake, implementations MUST use an automated key management system, such as the Internet Key Exchange (IKE) [RFC2409], to ensure that this requirement is met.

Therefore, Manual Keys MUST NOT be used with this algorithm, and devices that do not support IKEv2 will FAIL this test case

AES-GMAC

According to RFC4106, Section 2:

> Because reusing an nonce/key combination destroys the security guarantees of AES-GCM mode, it can be difficult to use this mode securely when using statically configured keys. For safety's sake, implementations MUST use an automated key management system, such as the Internet Key Exchange (IKE) [RFC2409], to ensure that this requirement is met.

Therefore, Manual Keys MUST NOT be used with this algorithm, and devices that do not support IKEv2 will FAIL this test case.

ChaCha20-Poly1305

According to RFC7634, Section 2:

> The Internet Key Exchange Protocol generates a bitstring called KEYMAT using a pseudorandom function (PRF). That KEYMAT is divided into keys for encryption, message authentication, and whatever else is needed. The KEYMAT requested for each ChaCha20-Poly1305 key is 36 octets. The first 32 octets are the 256-bit ChaCha20 key, and the remaining 4 octets are used as the Salt value in the nonce.

Also, from Section 5:

> The most important security consideration in implementing this document is the uniqueness of the nonce used in ChaCha20. The nonce should be selected uniquely for a particular key, but unpredictability of the nonce is not required. Counters and LFSRs are both acceptable ways of generating unique nonces.

Therefore, Manual Keys MUST NOT be used with this algorithm, and devices that do not support IKEv2 will FAIL this test case.