IPv6 Ready Logo

Phase-2 Interoperability
Test Scenario
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	Renumber and reorganized Test Sections Create Common Configurations 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 ESP=AES-CBC(128-bit) HMAC-SHA-256 (Section 5.1.13, 5.2.13, 5.3.13, 5.4.13) Added test cases for ESP=AES-CBC HMAC-SHA-384 (Section 5.1.14, 5.2.14, 5.3.14, 5.4.14) Added test cases for ESP=AES-CBC(256-bit) HMAC-SHA-512 (Section 5.1.15, 5.2.15, 5.3.15, 5.4.15) Added test cases for ESP=AES-GCM NULL (Section 5.1.16, 5.2.16, 5.3.16, 5.4.16), RFC 4106 "The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)" Added test cases for ESP=NULL AES-GMAC (Section 5.1.17, 5.2.17, 5.3.17, 5.4.17), RFC 4543 "The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH Modified formatting and fixed typos
1.11.0	2011-05-10	Timothy Carlin	Change test sequence of Section 5.3.11 (Section 5.3.11 uses new test topology for End-Node vs. SGW Tunnel Mode Test 2) Removed NULL Integrity tests Typos and Bug fixes
1.10.0	2010-05-10		Support Integrity Algorithm HMAC-SHA-256 in RFC 4868 (Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec) (Section 5.1.12, 5.2.12, 5.3.12, 5.4.12)
1.9.2	2017-03-11		Add Fragmentation test cases (Section 5.1.11, 5.2.11, 5.3.11, 5.4.11) Editorial fix at Appendix-A Section 1.2 Added the description of keying information file at Appendix-A Required Data Added file lists needed to be submitted at Appendix-A Section 1.3 Clarified the interoperable device requirement at REQUIREMENTS section
1.9.1	2009-01-06		Support the passive node which doesn't have ping6 application (as Possible Problems in Section 5.1.8, 5.3.8, 5.4.8)

1.9.0	2008-12-09	Support RFC 4312 (The Camellia Cipher Algorithm and Its Use With IPsec) (Section 5.1.7, 5.2.7, 5.3.7, 5.4.7) Use IPv6 prefix defined in RFC 3849 for the documentation
1.5.2	2007-10-11	Remove ESN test cases (Section 5.1.8, 5.2.8, 5.3.8, 5.4.8)
1.5.1	2007-06-19	Correct subsection in Section 5.3
1.5.0	2007-05-27	Support IPsec v3
1.4.3	2005-10-06	Update Appendix
1.4.2	2005-09-30	Change ping direction for tunnel tests between END-Nodes
1.4.1	2005-09-22	Editorial fix
1.4	2005-03-01	Change Keys
1.3	2004-12-21	Correct Require table
1.2	2004-11-29	Add concept of End-Node rather than Host Add criteria Editorial fix
1.1	2004-09-30	
1.0	2004-09-24	

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

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.
- Passive Node
  - If your device is an End-Node and cannot send ICMP Echo Request, it must play the role of TAR-EN1. Otherwise, it can play either role. In either case choose a device which can send ICMP Echo Request as TAR-EN2.

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.

Interoperable Device Requirements

IPv6 Logo Committee requires interoperable devices to obtain the IPv6 Ready Logo Phase-2 as following.

End-Node

- Transport Mode (BASIC): Test 5.1.X is required
    - 2 End-Node devices from different vendors
- Tunnel Mode (ADVANCED): Test 5.3.X and Test 5.4.X are required
    - 2 End-Node or SGW devices from different vendors

Test 5.1.X	Transport Mode	End-Node 1	Vendor A	BASIC
		End-Node 2	Vendor B	
Test 5.3.X	Tunnel Mode	SGW 1	Vendor C	ADVANCED
		SGW 2	Vendor D	
Test 5.4.X	Tunnel Mode	End-Node 1	Vendor A	ADVANCED
		End-Node 2	Vendor B	
		End-Node 3	Vendor C	
		End-Node 4	Vendor D	

SGW

- Tunnel Mode (BASIC): Test 5.2.X and Test 5.3.X are required
    - 2 SGW devices or 2 End-Node devices from different vendors

Test 5.2.X	Tunnel Mode	SGW 1	Vendor A	BASIC
		SGW 2	Vendor B	
Test 5.3.X	Tunnel Mode	End-Node 1	Vendor C	BASIC
		End-Node 2	Vendor D	

Required Tests

Test Case	Title	IPv6Ready Requirement
IPsec.IO.1.1.1	Transport Mode: ESP Algorithms	See IPsec.IO.1.1.1 Below
IPsec.IO.1.1.2	Transport Mode: Packet Too Big Processing	EN: Basic
IPsec.IO.1.1.3	Transport Mode: ICMPv6 Selectors	EN: Basic
IPsec.IO.2.1.1	Tunnel Mode: ESP Algorithms	See IPsec.IO.2.1.1 Below
IPsec.IO.2.1.2	Tunnel Mode: Packet Too Big Processing	EN: Basic
IPsec.IO.2.1.3	Tunnel Mode: ICMPv6 Selectors	EN: Basic
IPsec.IO.3.1.1	Tunnel Mode: ESP Algorithms	See IPsec.IO.3.1.1 Below
IPsec.IO.3.1.2	Tunnel Mode: Encrypted PTB	EN: Basic SGW: Basic
IPsec.IO.3.1.3	Tunnel Mode: Cleartext PTB	EN: Basic SGW: Basic
IPsec.IO.4.1.1	Tunnel Mode: ESP Algorithms	See IPsec.IO.4.1.1 Below
IPsec.IO.4.1.2	Tunnel Mode: Packet Too Big Processing	SGW: Basic
IPsec.IO.1.1.1 IPsec.IO.2.1.1 IPsec.IO.3.1.1 IPsec.IO.4.1.1 Part A	NULL/SHA256	EN: Basic SGW: Basic
IPsec.IO.1.1.1 IPsec.IO.2.1.1 IPsec.IO.3.1.1 IPsec.IO.4.1.1 Part B	AES128/SHA1	EN: Basic SGW: Basic
IPsec.IO.1.1.1 IPsec.IO.2.1.1 IPsec.IO.3.1.1 IPsec.IO.4.1.1 Part C	AES128/SHA256	EN: Basic SGW: Basic
IPsec.IO.1.1.1 IPsec.IO.2.1.1 IPsec.IO.3.1.1 IPsec.IO.4.1.1 Part D	AES256/SHA256	EN: Basic SGW: Basic
IPsec.IO.1.1.1 IPsec.IO.2.1.1 IPsec.IO.3.1.1 IPsec.IO.4.1.1 Part E	AES256/SHA512	EN: Advanced SGW: Advanced
IPsec.IO.1.1.1 IPsec.IO.2.1.1 IPsec.IO.3.1.1 IPsec.IO.4.1.1 Part F	AESCCM128/AESXCBC	EN: Advanced SGW: Advanced
IPsec.IO.1.1.1 IPsec.IO.2.1.1 IPsec.IO.3.1.1 IPsec.IO.4.1.1 Part G	AESCCM256/AESXCBC	EN: Advanced SGW: Advanced
IPsec.IO.1.1.1 IPsec.IO.2.1.1 IPsec.IO.3.1.1 IPsec.IO.4.1.1 Part H	AESGCM128	EN: Basic SGW: Basic
IPsec.IO.1.1.1	AESGCM256	EN: Basic

IPsec.IO.2.1.1 ***IPsec.IO.3.1.1*** ***IPsec.IO.4.1.1*** ***Part I***		SGW: Basic
IPsec.IO.1.1.1 ***IPsec.IO.2.1.1*** ***IPsec.IO.3.1.1*** ***IPsec.IO.4.1.1*** ***Part J***	AESGMAC128	EN: Basic SGW: Basic
IPsec.IO.1.1.1 ***IPsec.IO.2.1.1*** ***IPsec.IO.3.1.1*** ***IPsec.IO.4.1.1*** ***Part K***	AESGMAC256	EN: Basic SGW: Basic
IPsec.IO.1.1.1 ***IPsec.IO.2.1.1*** ***IPsec.IO.3.1.1*** ***IPsec.IO.4.1.1*** ***Part L***	CHACHA20_POLY1305	EN: Advanced SGW: Advanced

1. If applicant's device is a SGW, then the "SGW vs. SGW (Tunnel)" AND "End-Node vs. SGW (Tunnel)" tests must be run. Applicants need to run tests with more than 2 implementations as a counterpart regardless equipment type.
2. If applicant's device is an End-Node then the “End-Node vs. SGW (Tunnel)” AND “End-Node vs. End-Node (Tunnel)” tests must be run. Applicans need to run tests with more than 2 implementations as a counterpart regardless equipment type.
3. This test should be run using ICMP.
4. This test should be run using UDP.

References

This test specification focus 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

Topology 1: End-Node vs. End-Node

1. Set global address to TAR-EN1_Network0 and TAR-EN2_Network1 by RA.
2. Make IPsec transport mode or tunnel mode between TAR-EN1 and TAR-EN2.

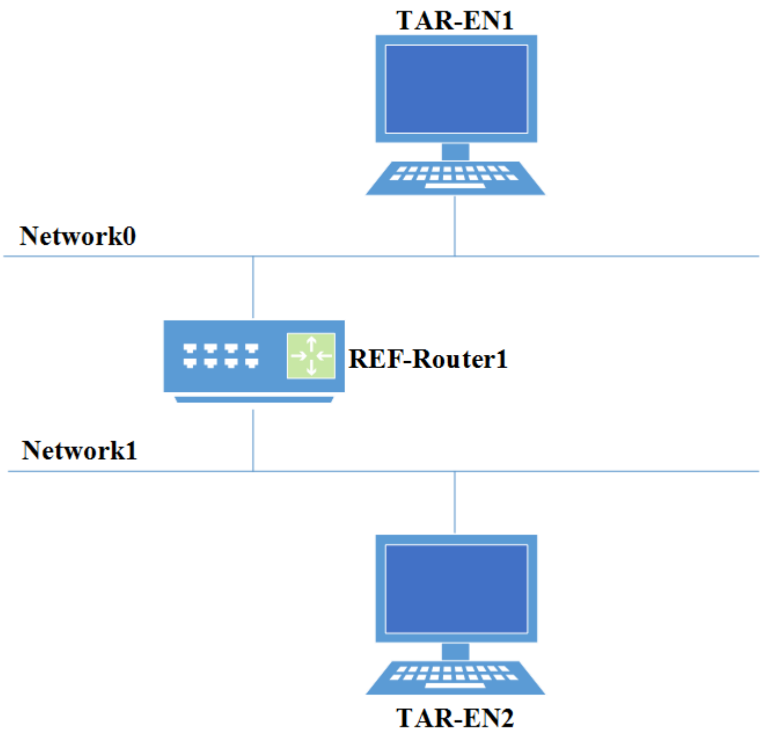


FIGURE 1 TOPOLOGY FOR END-NODE: TRANSPORT AND TUNNEL MODE WITH END-NODE

Addresses		
Network0:	TAR-EN1:	2001:0db8:ffff:0000:: [*interface id]*
	REF-Router1:	2001:0db8:ffff:0000::f
Network1:		2001:0db8:ffff:0001::f
	TAR-EN2:	2001:0db8:ffff:0001:: [*interface id]*

Topology 2: SGW vs. SGW

1. Set global address to REF-Host1_Network0 and REF-Host2_Network3 by RA.
2. Set global address to TAR-SGW1_Network0, TAR-SGW1_Network1, TAR-SGW2_Network2, TAR-SGW2_Network3, REF-Router1_Network1, REF-Router1_Network2 manually.
3. Set routing table to TAR-SGW1 (REF-Router1_Network1 for Network2 and Network3)
4. Set routing table to TAR-SGW2 (REF-Router1_Network2 for Network0 and Network1)
5. Set routing table to REF-Router1 (TAR-SGW1_Network1 for Network0, TAR-SGW2_Network2 for Network3)
6. Make IPsec tunnel mode between TAR-SGW1 and TAR-SGW2.

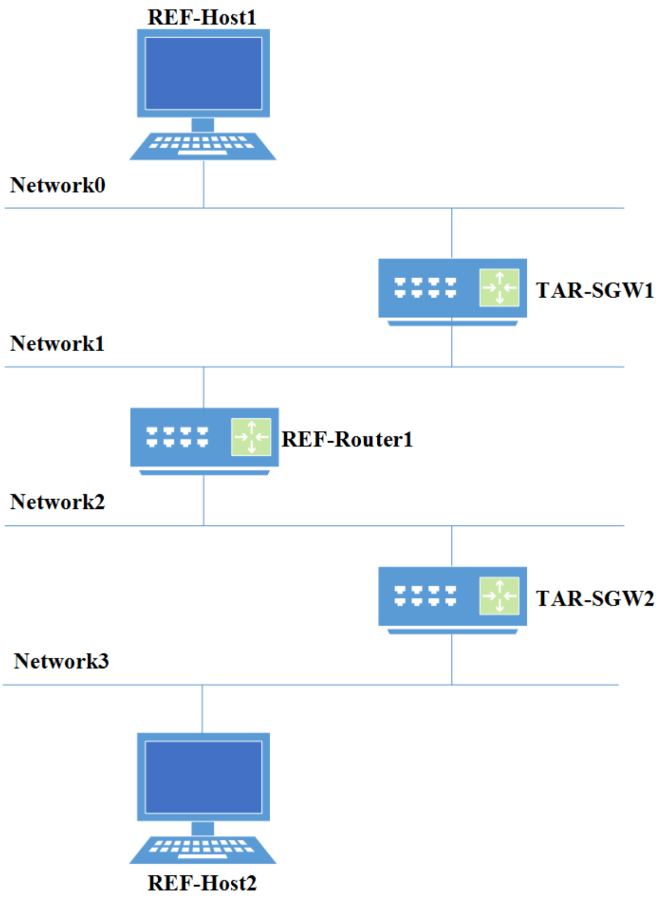


Figure 2 Topology for SGW: Tunnel mode with SGW

Addresses		
Network0:	REF-Host1:	2001:0db8:ffff:0000::[*interface id]*
	TAR-SGW1:	2001:0db8:ffff:0000::f
Network1:		2001:0db8:ffff:0001::f
	REF-Router1:	2001:0db8:ffff:0001::e
Network2:		2001:0db8:ffff:0002::e
	TAR-SGW2:	2001:0db8:ffff:0002::d
Network3:		2001:0db8:ffff:0003::d
	REF-Host2:	2001:0db8:ffff:0003:: [*interface id]*

Topology 3: End-Node vs. SGW

1. Set global address to TAR-EN1_Network0 and REF-Host1_Network2 by RA.
2. Set global address to TAR-SGW1_Network1 and TAR-SGW1_Network2 manually.
3. Set routing table to TAR-SGW1 (REF-Router1_Network1 for Network0)
4. Set routing table to REF-Router1 (TAR-SGW1_Network1 for Network2)
5. Make IPsec tunnel mode between TAR-EN1 and TAR-SGW1.

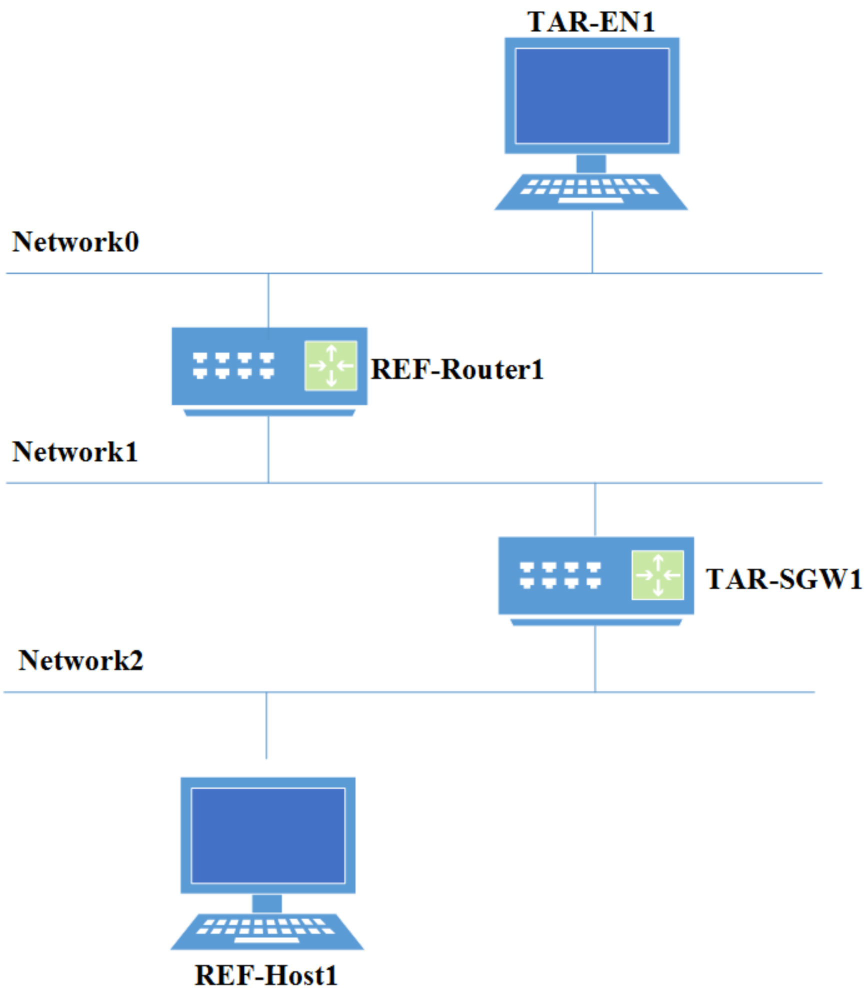


FIGURE 3 TOPOLOGY FOR END-NODE: TUNNEL MODE WITH SGW

Addresses		
Network0:	TAR-EN1:	2001:0db8:ffff:0000:: [*interface id]*
	REF-Router1:	2001:0db8:ffff:0000::f
Network1:		2001:0db8:ffff:0001::f
	TAR-SGW1:	2001:0db8:ffff:0001::e
Network2:		2001:0db8:ffff:0002::e
	REF-Host1:	2001:0db8:ffff:0002:: [*interface id]*

Topology 4: End-Node vs. SGW

1. Set global address to TAR-EN1_Network0 and REF-Host1_Network3 by RA.
2. Set global address to TAR-SGW1_Network1 and TAR-SGW1_Network2 manually.
3. Set routing table to TAR-SGW1 (REF-Router1_Network1 for Network0)
4. Set routing table to TAR-SGW1 (REF-Router2_Network2 for Network3)
5. Set routing table to REF-Router1 (TAR-SGW1_Network1 for Network2)
6. Set routing table to REF-Router1 (TAR-SGW1_Network1 for Network3)
7. Set routing table to REF-Router2 (TAR-SGW1_Network2 for Network0)
8. Set routing table to REF-Router2 (TAR-SGW1_Network2 for Network1)
9. Make IPsec tunnel mode between TAR-EN1 and TAR-SGW1.

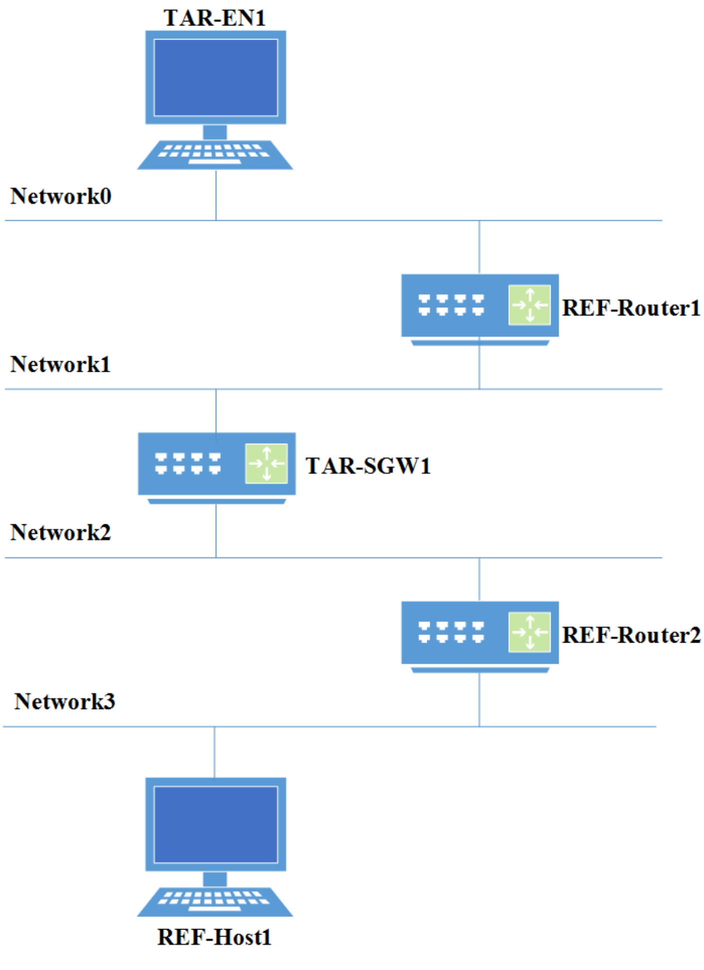


FIGURE 4 TOPOLOGY FOR END-NODE: TUNNEL MODE WITH SGW

Addresses		
Network0:	TAR-EN1:	2001:0db8:ffff:0000:: [*interface id]*
	REF-Router1:	2001:0db8:ffff:0000::f
Network1:		2001:0db8:ffff:0001::f
	TAR-SGW1:	2001:0db8:ffff:0001::e
Network2:		2001:0db8:ffff:0002::e
	REF-Router2:	2001:0db8:ffff:0002::d
Network3:		2001:0db8:ffff:0003::d
	REF-Host1:	2001:0db8:ffff:0003:: [*interface id]*

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

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

Common 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

ESP Algorithms

Algorithm List

Use the Global Security Associations for IKEv2. Run each part for each test case which references this section. Substitute the ESP configuration with the below algorithms in each part

The test case parts itemized below are used in this section, and are referred to by each test case. The Policy configuration is defined by the 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

Use the table below as needed to configure Manual Keys.

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.

Transport Mode: End-Node vs. End-Node

Configuration 1

Use the Global Security Associations, with the below policy configuration.

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

TAR-EN2 ——— TAR-EN1
TAR-EN2_SA-O → TAR-EN1_SA-I
TAR-EN2_SA-I ← TAR-EN1_SA-O

Policy 1	
Peer Left	TAR-EN2_Networ1
Peer Right	TAR-EN1_Network1
Mode	Transport
Traffic Selector Address Left	TAR-EN2_Networ1
Traffic Selector Address Right	TAR-EN1_Network1
Traffic Selector Protocol/Port	ANY/ANY
If using Manual Keys include:	
SA Left	SA1-I
SA Right	SA1-O

Configuration 2

Use the Global Security Associations, with the below policy configuration.

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

TAR-EN2	———	TAR-EN1	
TAR-EN2_SA1-O	→	TAR-EN1_SA1-I	ICMPv6 Echo Request
TAR-EN2_SA1-I	←	TAR-EN1_SA1-O	ICMPv6 Echo Request
TAR-EN2_SA2-O	→	TAR-EN1_SA2-I	ICMPv6 Echo Reply
TAR-EN2_SA2-I	←	TAR-EN1_SA2-O	ICMPv6 Echo Reply

Policy 1	
Peer Left	TAR-EN2_Networ1
Peer Right	TAR-EN1_Network1
Mode	Transport
Traffic Selector Address Left	TAR-EN2_Networ1
Traffic Selector Address Right	TAR-EN1_Network1
Traffic Selector Protocol/Port	ICMPv6/128 (Echo Request)
If using Manual Keys include:	
SA Left	SA1-I
SA Right	SA1-O

Policy 2	
Peer Left	TAR-EN2_Networ1
Peer Right	TAR-EN1_Network1
Mode	Transport
Traffic Selector Address Left	TAR-EN2_Networ1
Traffic Selector Address Right	TAR-EN1_Network1
Traffic Selector Protocol/Port	ICMPv6/129 (Echo Reply)
If using Manual Keys include:	
SA Left	SA2-I
SA Right	SA2-O

Tunnel Mode: SGW vs. SGW

Configuration 3

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

REF-Host2— TAR-SGW2 ——————— TAR-SGW1—REF-Host1
TAR-SGW2_SA1-O ———————→ TAR-SGW1_SA1-I
TAR-SGW2_SA1-I ←——————— TAR-SGW1_SA1-O

Policy 1	
Peer Left	TAR-SGW2_Network2
Peer Right	TAR-SGW1_Network1
Mode	Tunnel
Traffic Selector Address Left	Network3
Traffic Selector Address Right	Network0
Traffic Selector Protocol/Port	ANY/ANY
If using Manual Keys include:	
SA Left	SA1-I
SA Right	SA1-O

Tunnel Mode: End-Node vs. SGW

Configuration 4

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

TAR-EN1 ——— TAR-SGW1—REF-Host2
TAR-EN1_SA1-O ——→ TAR-SGW1_SA1-I
TAR-EN1_SA1-I ←—— TAR-SGW1_SA1-O

Policy 1	
Peer Left	TAR-EN1_Network0
Peer Right	TAR-SGW1_Network1
Mode	Tunnel
Traffic Selector Address Left	TAR-EN1_Network0
Traffic Selector Address Right	Network2
Traffic Selector Protocol/Port	ANY/ANY
If using Manual Keys include:	
SA Left	SA1-I
SA Right	SA1-O

Configuration 5

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

TAR-EN1 ——— TAR-SGW1—REF-Router2 - REF-Host2
TAR-EN1_SA1-O ——→ TAR-SGW1_SA1-I
TAR-EN1_SA1-I ←—— TAR-SGW1_SA1-O

Policy 1	
Peer Left	TAR-EN1_Network0
Peer Right	TAR-SGW1_Network1
Mode	Tunnel
Traffic Selector Address Left	TAR-EN1_Network0
Traffic Selector Address Right	Network2+Network3
Traffic Selector Protocol/Port	ANY/ANY
If using Manual Keys include:	
SA Left	SA1-I
SA Right	SA1-O

Tunnel Mode: End-Node vs. End-Node

Configuration 6

Use the Global Security Associations, with the below policy configuration.

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

TAR-EN2 ——— TAR-EN1
TAR-EN2_SA-O ——→ TAR-EN1_SA-I
TAR-EN2_SA-I ←—— TAR-EN1_SA-O

Policy 1	
Peer Left	TAR-EN2_Networ1
Peer Right	TAR-EN1_Network1
Mode	Tunnel
Traffic Selector Address Left	TAR-EN2_Networ1
Traffic Selector Address Right	TAR-EN1_Network1
Traffic Selector Protocol/Port	ANY/ANY
If using Manual Keys include:	
SA Left	SA1-I
SA Right	SA1-O

Section 1: Transport Mode: End-Node vs. End-Node

1.1: Transport Mode: End-Node vs. End-Node

Scope
The following tests focus on Transport Mode.

Overview
Tests in this section verify that a node properly processes and transmits the packets to which IPsec Transport Mode is applied between two End-Nodes.

IPsec.IO.1.1.1. ESP Algorithms

Purpose

Verify ESP Algorithms in Transport Mode between two End-Nodes

Initialization

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - In each part, configure the devices according to the ESP Algorithms, and Configuration 1

Packets

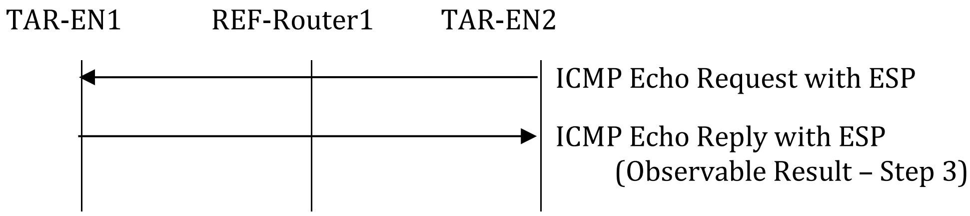
IP Header	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-I/TAR-EN2_SA-O
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-O/TAR-EN2_SA-I
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure



All Parts: Algorithms

Step	Action	Expected Result
1.	Initialize the Devices	
2.	Transmit *ICMP Echo Request with ESP* from TAR-EN2 to the Global unicast address of TAR-EN1	
3.	Observe the packets transmitted on Network 0	The NUT transmits ICMP Echo Reply with ESP

Possible Problems

None

IPsec.IO.1.1.2. Fragmentation

Purpose

Verify IPv6 Packet Too Big Processing, Fragmentation, and Reassembly in Transport Mode between two End-Nodes

Initialization

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - In each part, configure the devices according to the Global Security Associations, and Configuration 1

Packets

IP Header	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-I/TAR-EN2_SA-O
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP 1

IP Header	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_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 with ESP 2

IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-O/TAR-EN2_SA-I
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	REF-Router1
	Destination Address	TAR-EN1
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	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-O/TAR-EN2_SA-I
ICMP	Type	129 (Echo Reply)

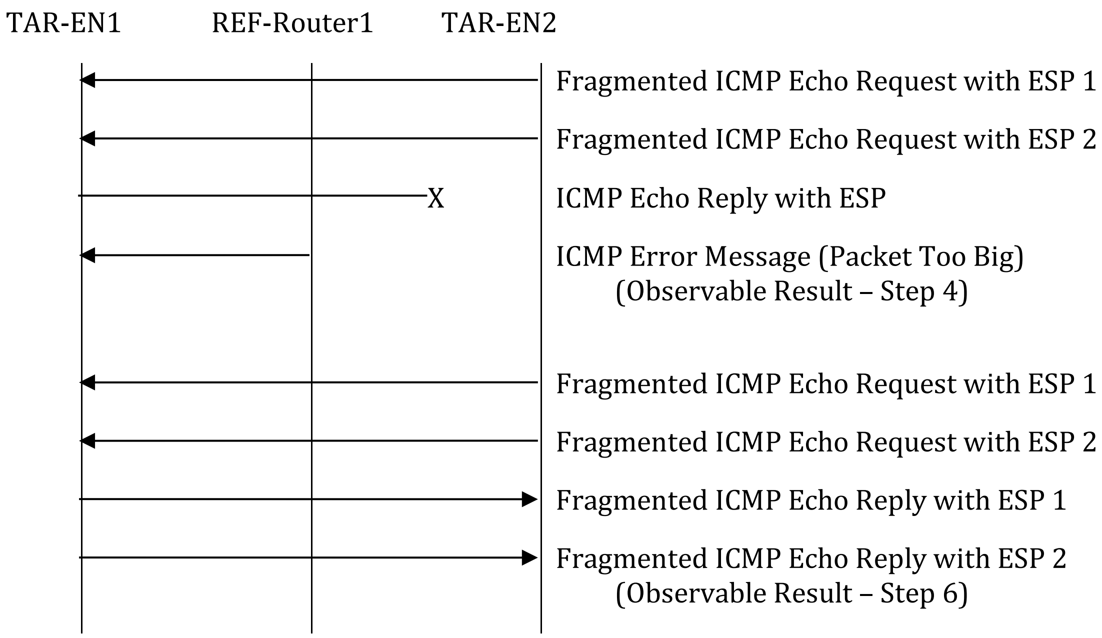
Fragmented ICMP Echo Reply with ESP 1

IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
	Payload Length	2ndPL (= 1476-1stPL)
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of ICMP Echo Reply

Fragmented ICMP Echo Reply with ESP 2

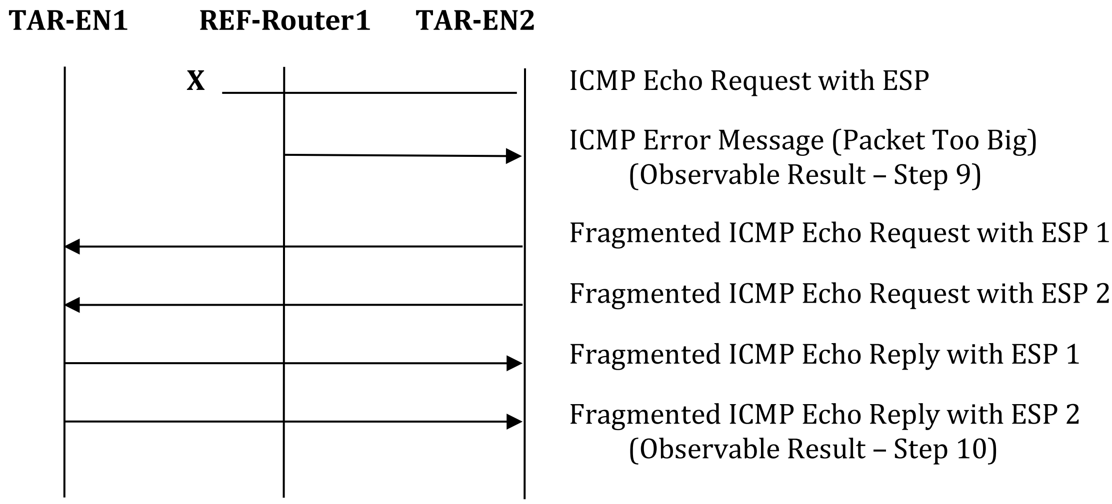
Procedure

Part A: TAR-EN1 Packet Too Big Processing



Step	Action	Expected Result
1.	Configure the Network1 interface of REF-Router1 with a path MTU of 1280 bytes	
2.	Initialize the Devices	
3.	Transmit Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 from TAR-EN2 to the Global unicast address of TAR-EN1	
4.	Observe the packets transmitted on Network 0 and Network 1	TAR-EN1 transmits ICMP Echo Reply with ESP REF-Router1 transmits ICMP Error Message (Packet Too Big) to TAR-EN1
5.	Transmit Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 from TAR-EN2 to the Global unicast address of TAR-EN1	TAR-EN1 transmits Fragmented ICMP Echo Reply with ESP 1 and Fragmented ICMP Echo Reply with ESP 2

Part B: TAR-EN2 Packet Too Big Processing



Step	Action	Expected Result
6.	Configure the Network0 interface of REF-Router1 with a path MTU of 1280 bytes	
7.	Initialize the Devices	
8.	Transmit ICMP Echo Request with ESP from TAR-EN2 to the Global unicast address of TAR-EN1	
9.	Observe the packets transmitted on Network 0 and Network 1	REF-Router1 transmits ICMP Error Message (Packet Too Big) to TAR-EN2.
10.	Transmit Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 from TAR-EN2 to the Global unicast address of TAR-EN1	TAR-EN1 transmits Fragmented ICMP Echo Reply with ESP 1 and Fragmented ICMP Echo Reply with ESP 2

Possible Problems

None

IPsec.IO.1.1.3. Transport Mode ICMPv6 Traffic Selectors

Purpose

Verify ICMPv6 Traffic Selectors with Transport Mode between two End-Nodes

Initialization

- Network Topology
    - Connect the devices according to Common Topology 1
- Configuration
    - In each part, configure the devices according to the Global Security Associations, and Configuration 2

Packets

IP Header	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA1-I/ TAR-EN2_SA1-O
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP 1

IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ESP	SPI	*Dynamic4 or* 0x4000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA2-O/ TAR-EN2_SA2-I
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP 1

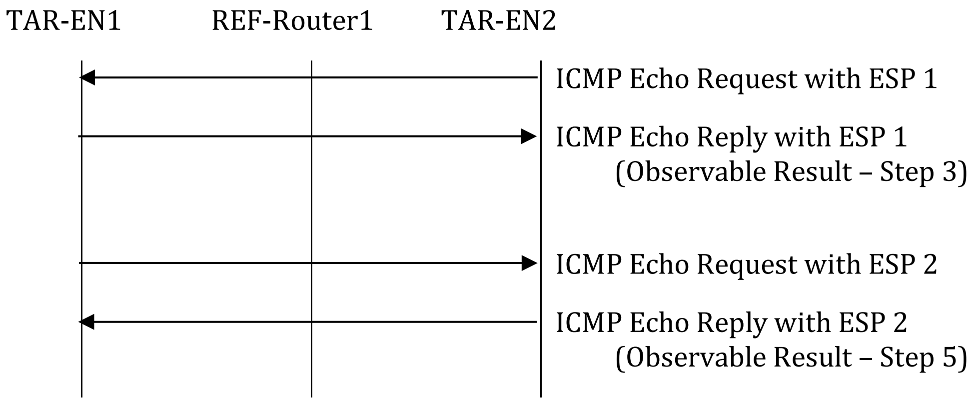
IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA1-O/ TAR-EN2_SA1-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP 2

IP Header	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	*Dynamic3 or* 0x3000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA2-I/ TAR-EN2_SA2-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP 2

Procedure



Step	Action	Expected Result
1.	Initialize the Devices	
2.	Transmit ICMP Echo Request with ESP 1 from TAR-EN2 to the Global unicast address of TAR-EN1	
3.	Observe the packets transmitted on Network 0	TAR-EN1 transmits ICMP Echo Reply with ESP 1
4.	Transmit ICMP Echo Request with ESP 2 from TAR-EN1 to the Global unicast address of TAR-EN2	
5.	Observe the packets transmitted on Network 0	TAR-EN1 transmits ICMP Echo Reply with ESP 2

Possible Problems

TAR-EN1 or TAR-EN2 may be a passive node which does not implement an application for sending Echo Requests. One of the following method to perform this test is required for the passive node.

- using UDP application to invoke ICMPv6 Destination Unreachable (Port unreachable) (see Appendix-B Section 1.1)
- invoking Neighbor Unreachability Detection (see Appendix-B Section 1.2)

Section 2: Tunnel Mode (End-Node vs. End-Node)

2.1: Tunnel Mode: End-Node vs. End-Node

Scope

The following tests focus on Tunnel Mode.

Overview

Tests in this section verify that a node properly processes and transmits the packets to which IPsec Tunnel Mode is applied between two End-Nodes.

IPsec.IO.2.1.1. Tunnel Mode ESP Algorithms

Purpose

Verify ESP Algorithms in Tunnel Mode between two End-Nodes

Initialization

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - In each part, configure the devices according to the ESP Algorithms, and Configuration 6

Packets

Packets:

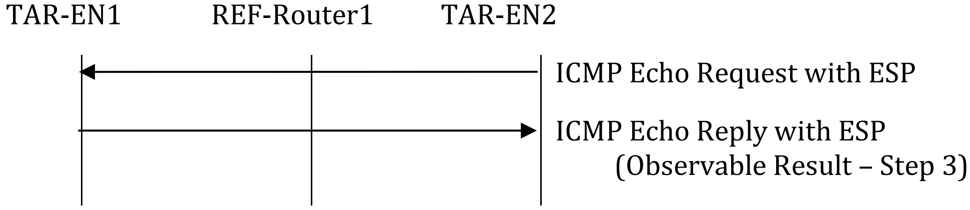
IP Header	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-I/ TAR-EN2_SA-O
IP Header	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-O/ TAR-EN2_SA-I
IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure



All Parts: Algorithms

Step	Action	Expected Result
1.	Initialize the Devices	
2.	Transmit *ICMP Echo Request with ESP* from TAR-EN2 to the Global unicast address of TAR-EN1	
3.	Observe the packets transmitted on Network 0	The NUT transmits ICMP Echo Reply with ESP

Possible Problems

None

IPsec.IO.2.1.2. Tunnel Mode Fragmentation

Purpose

Verify IPv6 Packet Too Big Processing, Fragmentation, and Reassembly in Tunnel Mode between two End-Nodes

Initialization

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - In each part, configure the devices according to the Global Security Associations, and Configuration 6

Packets

IP Header	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-I/TAR-EN2_SA-O
IP Header	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
	Payload Length	1stPL (= MTU-40) (e.g. 1240)
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP 1

IP Header	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-I/TAR-EN2_SA-O
IP Header	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_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 with ESP 2

IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-O/TAR-EN2_SA-I
IP Header	Source Address	TAR-EN1_Network0

	Destination Address	TAR-EN2_Network1
	Payload Length	1460
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	REF-Router1
	Destination Address	TAR-EN1
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	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-O/TAR-EN2_SA-I
IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
	Payload Length	1stPL (= MTU-40) (e.g. 1240)
Fragment	Offset	0
	More Flag	1
ICMP	Type	129 (Echo Reply)

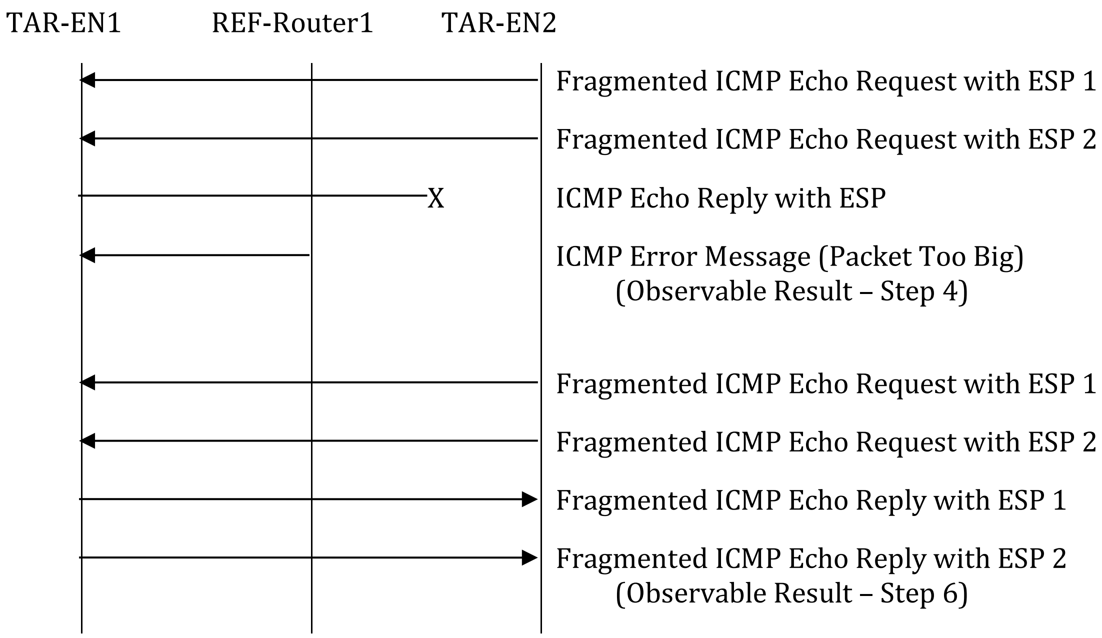
Fragmented ICMP Echo Reply with ESP 1

IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-O/TAR-EN2_SA-I
IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
	Payload Length	2ndPL (= 1476-1stPL)
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of ICMP Echo Reply

Fragmented ICMP Echo Reply with ESP 2

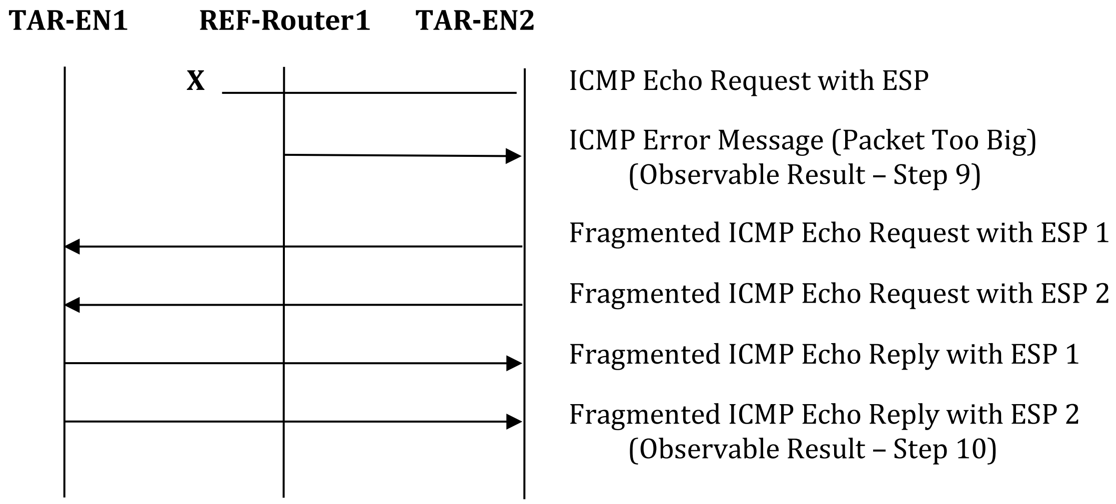
Procedure

Part A: TAR-EN1 Packet Too Big Processing



Step	Action	Expected Result
1.	Configure the Network1 interface of REF-Router1 with a path MTU of 1280 bytes	
2.	Initialize the Devices	
3.	Transmit Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 from TAR-EN2 to the Global unicast address of TAR-EN1	
4.	Observe the packets transmitted on Network 0 and Network 1	TAR-EN1 transmits ICMP Echo Reply with ESP REF-Router1 transmits ICMP Error Message (Packet Too Big) to TAR-EN1
5.	Transmit Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 from TAR-EN2 to the Global unicast address of TAR-EN1	TAR-EN1 transmits Fragmented ICMP Echo Reply with ESP 1 and Fragmented ICMP Echo Reply with ESP 2

Part B: TAR-EN2 Packet Too Big Processing



Step	Action	Expected Result
6.	Configure the Network0 interface of REF-Router1 with a path MTU of 1280 bytes	
7.	Initialize the Devices	
8.	Transmit ICMP Echo Request with ESP from TAR-EN2 to the Global unicast address of TAR-EN1	
9.	Observe the packets transmitted on Network 0 and Network 1	REF-Router1 transmits ICMP Error Message (Packet Too Big) to TAR-EN2.
10.	Transmit Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 from TAR-EN2 to the Global unicast address of TAR-EN1	TAR-EN1 transmits Fragmented ICMP Echo Reply with ESP 1 and Fragmented ICMP Echo Reply with ESP 2

Possible Problems

None

IPsec.IO.2.1.3. Tunnel Mode ICMPv6 Traffic Selectors

Purpose

Verify ICMPv6 Traffic Selectors with Tunnel Mode between two End-Nodes

Initialization

- Network Topology
  - Connect the devices according to Common Topology 1
- Configuration
  - In each part, configure the devices according to the Global Security Associations, and Configuration 6

Packets

IP Header	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA1-I/ TAR-EN2_SA1-O
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP 1

IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ESP	SPI	*Dynamic4 or* 0x4000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA2-O/ TAR-EN2_SA2-I
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP 1

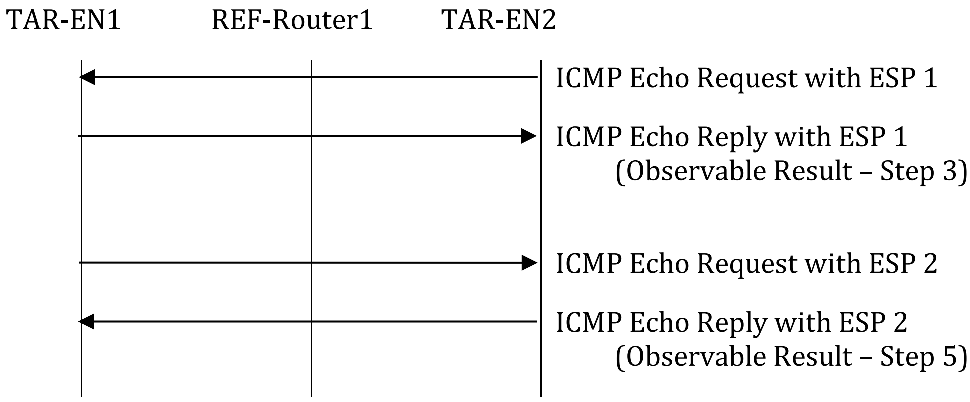
IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA1-O/ TAR-EN2_SA1-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP 2

IP Header	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	*Dynamic3 or* 0x3000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA2-I/ TAR-EN2_SA2-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP 2

Procedure



Step	Action	Expected Result
1.	Initialize the Devices	
2.	Transmit ICMP Echo Request with ESP 1 from TAR-EN2 to the Global unicast address of TAR-EN1	
3.	Observe the packets transmitted on Network 0	TAR-EN1 transmits ICMP Echo Reply with ESP 1
4.	Transmit ICMP Echo Request with ESP 2 from TAR-EN1 to the Global unicast address of TAR-EN2	
5.	Observe the packets transmitted on Network 0	TAR-EN1 transmits ICMP Echo Reply with ESP 2

Possible Problems

TAR-EN1 or TAR-EN2 may be a passive node which does not implement an application for sending Echo Requests. One of the following method to perform this test is required for the passive node.

- using UDP application to invoke ICMPv6 Destination Unreachable (Port unreachable) (see Appendix-B Section 1.1)
- invoking Neighbor Unreachability Detection (see Appendix-B Section 1.2)

Section 3: Tunnel Mode: (End-Node vs. SGW)

Scope
Following tests focus on Tunnel Mode between End-Node and SGW.

Overview
Tests in this section verify that a node properly processes and transmits the packets to which IPsec Tunnel Mode is applied between End-Node and SGWs.

3.1: Tunnel Mode: End-Node vs. SGW

Scope

The following tests focus on Tunnel Mode

Overview

Tests in this section verify that a node properly processes and transmits the packets to which IPsec Tunnel Mode is applied between an End-Node and SGW.

IPsec.IO.3.1.1. ESP Algorithms

Purpose

Verify ESP Algorithms in Tunnel Mode between End-node and SGW

Initialization

- Network Topology
    - Connect the devices according to Common Topology 3
- Configuration
    - In each part, configure the devices according to the ESP Algorithms, and Configuration 4

Packets

IP Header	Source Address	REF-Host2_Network2
	Destination Address	TAR-EN1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	TAR-SGW1_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-I/TAR-SGW1_SA-O
IP Header	Source Address	REF-Host2_Network2
	Destination Address	TAR-EN1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

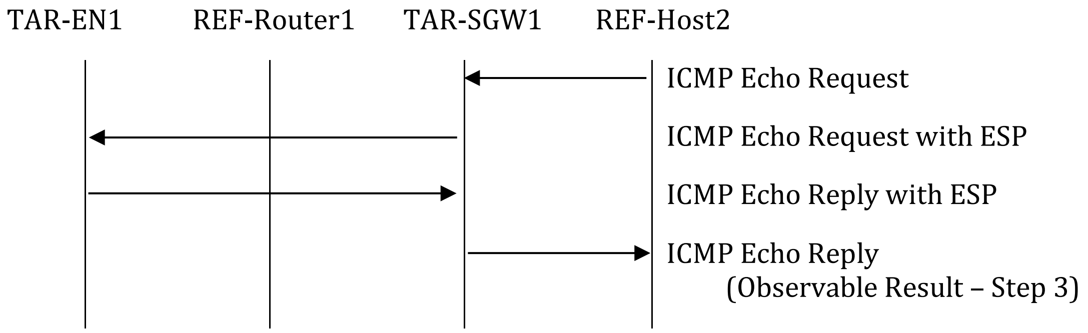
IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-SGW1_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-O/TAR-SGW1_SA-I
IP Header	Source Address	TAR-EN1_Network0
	Destination Address	REF-Host2_Network2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	TAR-EN1_Network0
	Destination Address	REF-Host2_Network2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

Procedure



IP Header	Source Address	TAR-SGW2_Network2
	Destination Address	TAR-SGW1_Network1
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1
	Encrypted Data/ICV	TAR-SGW1_SA-I/TAR-SGW2_SA-O
IP Header	Source Address	REF-Host2_Network3
	Destination Address	REF-Host1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	REF-Host2_Network3
	Destination Address	REF-Host1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	REF-Host1_Network0
	Destination Address	REF-Host2_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

IP Header	Source Address	TAR-SGW1_Network1
	Destination Address	TAR-SGW2_Network2
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-SGW1_SA-O/TAR-SGW2_SA-I
IP Header	Source Address	REF-Host1_Network0
	Destination Address	REF-Host2_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

All Parts: Algorithms

Step	Action	Expected Result
1.	Initialize the devices	
2.	Transmit ICMP Echo Request from REF-Host2 to the Global unicast address of TAR-EN1	
3.	Observe the packets transmitted on all networks	TAR-SGW1 transmits ICMP Echo Request with ESP TAR-EN1 transmits ICMP Echo Reply with ESP TAR-SGW1 transmits ICMP Echo Reply

Possible Problems

None

IPsec.IO.3.1.2. Fragmentation with Encrypted Packet Too Big

Purpose

Verify packet fragmentation and reassembly with an encrypted ICMPv6 Packet Too Big Message

Initialization

- Network Topology
  - Connect the devices according to Common Topology 4
- Configuration
  - In each part, configure the devices according to the Global Security Associations, and Configuration 5

Packets

IP Header	Source Address	REF-Host2_Network2
	Destination Address	TAR-EN1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	TAR-SGW1
	Destination Address	REF-Host2
ICMP	Type	2 (Packet Too Big)
	MTU	1280
	Data	1232Byte of ICMP Echo Request

ICMP Error Message to REF-Host2 (Packet Too Big)

IP Header	Source Address	REF-Host2_Network2
	Destination Address	TAR-EN1_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	REF-Host2_Network2
	Destination Address	TAR-EN1_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	TAR-SGW1_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1

	Encrypted Data/ICV	TAR-EN1_SA-I/TAR-SGW1_SA-O
IP Header	Source Address	REF-Host2_Network2
	Destination Address	TAR-EN1_Network0
	Payload Length	1stPL (= MTU-40) (e.g. 1240)
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP 1

IP Header	Source Address	TAR-SGW1_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-I/TAR-SGW1_SA-O
IP Header	Source Address	REF-Host2_Network2
	Destination Address	TAR-EN1_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 with ESP 2

IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-SGW1_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-O/TAR-SGW1_SA-I
IP Header	Source Address	TAR-EN1_Network0
	Destination Address	REF-Host2_Network2
	Payload Length	1460
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	REF-Router1
	Destination Address	TAR-EN1
ICMP	Type	2 (Packet Too Big)
	MTU	1280
	Data	1232Byte of ICMP Echo Reply with ESP

ICMP Error Message to TAR-EN1 (Packet Too Big)

IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-SGW1_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-O/TAR-SGW1_SA-I
IP Header	Source Address	TAR-EN1_Network0

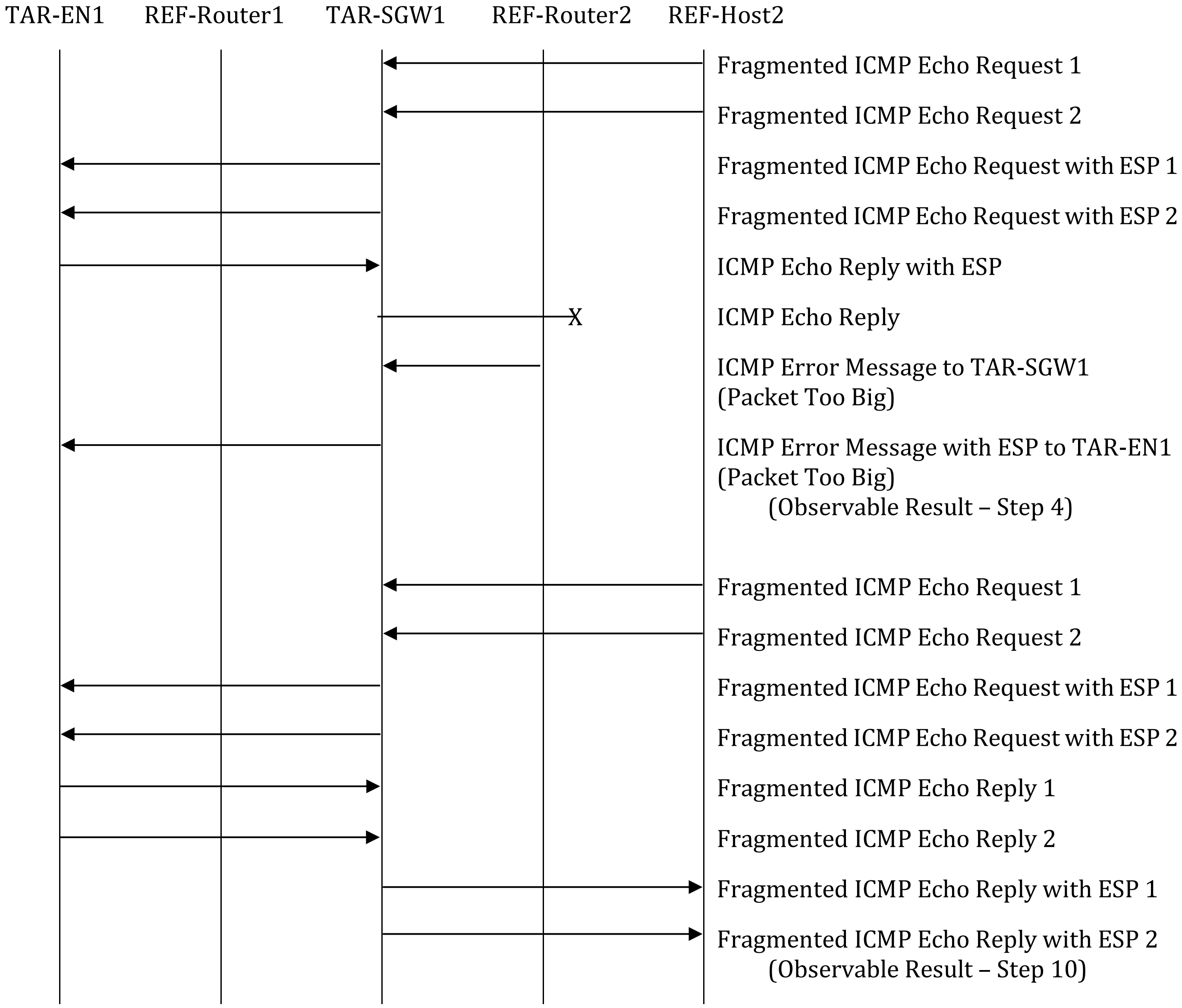
	Destination Address	REF-Host2_Network2
	Payload Length	1stPL (= MTU-40) (e.g. 1240)
Fragment	Offset	0
	More Flag	1
ICMP	Type	129 (Echo Reply)

Fragmented ICMP Echo Reply with ESP 1

IP Header	Source Address	TAR-EN1_Network0
	Destination Address	TAR-SGW1_Network1
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-EN1_SA-O/TAR-SGW1_SA-I
IP Header	Source Address	TAR-EN1_Network0
	Destination Address	REF-Host2_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 with ESP 2

Procedure


TAR-EN1
REF-Router1
TAR-SGW1
REF-Router2
REF-Host2
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
X
ICMP Echo Reply
ICMP Error Message to TAR-SGW1
(Packet Too Big)
ICMP Error Message with ESP to TAR-EN1
(Packet Too Big)
(Observable Result – Step 4)
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 1
Fragmented ICMP Echo Reply 2
Fragmented ICMP Echo Reply with ESP 1
Fragmented ICMP Echo Reply with ESP 2
(Observable Result – Step 10)

Step	Action	Expected Result
1.	Configure the Network3 interface of REF-Router2 with a path MTU of 1280 bytes	
2.	Initialize the devices	
3.	Transmit *Fragmented ICMP Echo Request 1* and *Fragmented ICMP Echo Request 2* from REF-Host2 to the Global unicast address of TAR-EN1	
4.	Observe the packets transmitted on all networks	TAR-SGW1 transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 TAR-EN1 transmits ICMP Echo Reply with ESP TAR-SGW1 transmits *ICMP Echo Reply* REF-Router2 transmits ICMP Error Message (Packet Too Big) to TAR-SGW1 TAR-SGW1 transmits *ICMP Error Message with ESP* to TAR-EN1
5.	Transmit *Fragmented ICMP Echo Request 1* and *Fragmented ICMP Echo Request 2* from REF-Host2 to the Global unicast address of TAR-EN1	
6.	Observe the packets transmitted on all networks	TAR-SGW1 transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 TAR-EN1 transmits Fragmented ICMP Echo Reply with ESP 1 and Fragmented ICMP Echo Reply with ESP 2 TAR-SGW1 transmits Fragmented ICMP Echo Reply 1 and Fragmented ICMP Echo Reply 2

Possible Problems

The link technology on Network1 may require fragmentation of the packet *Fragmented ICMP Echo Request with ESP 1*. In this case, TAR-SGW1 may further fragment this packet.

IPsec.IO.3.1.3. Fragmentation with Unprotected Packet Too Big

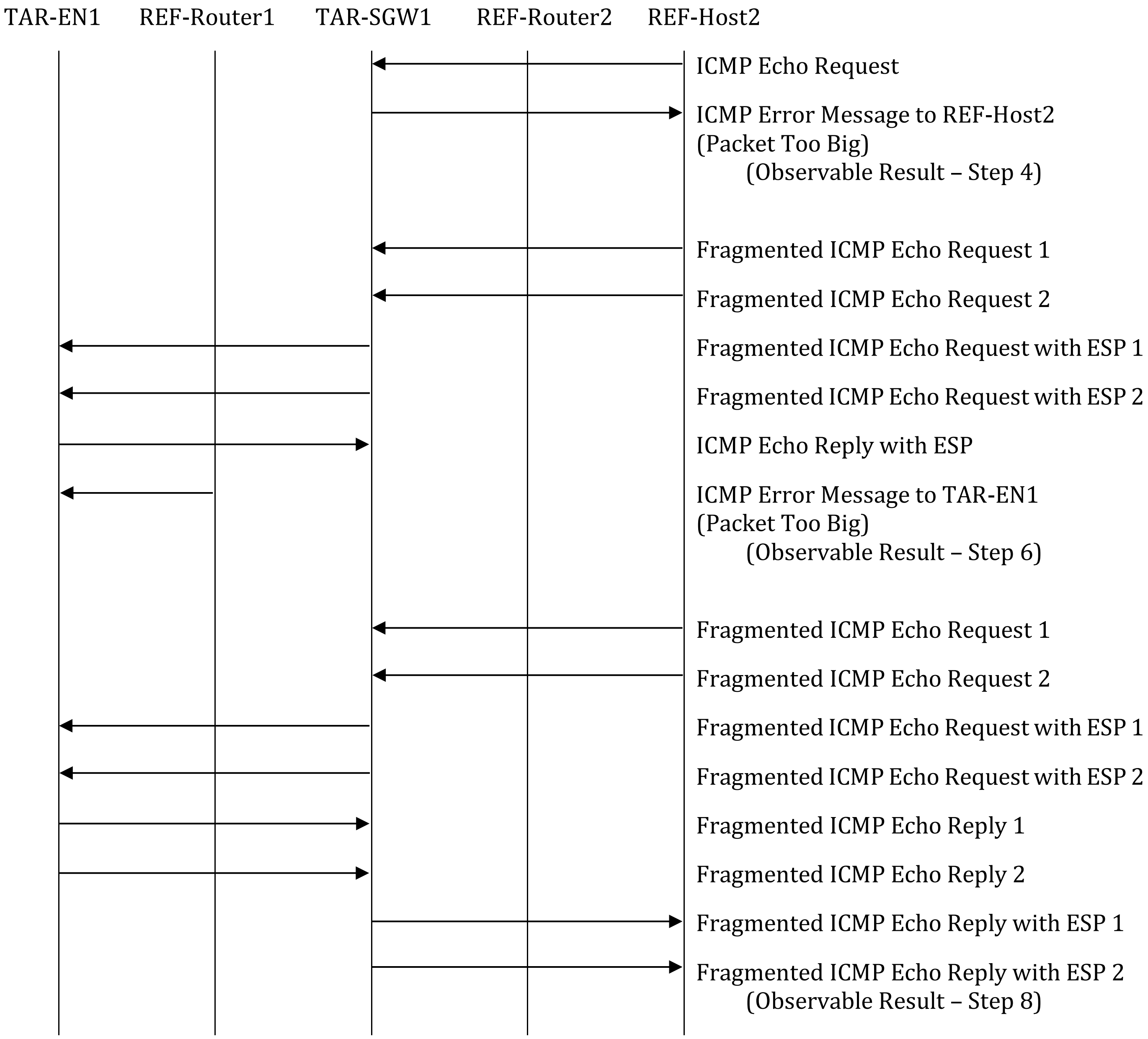
Purpose

Verify packet fragmentation and reassembly with an unprotected ICMPv6 Packet Too Big Message

Initialization

- Network Topology
  - Connect the devices according to Common Topology 4
- Configuration
  - In each part, configure the devices according to the Global Security Associations, and Configuration 5

Packets


TAR-EN1
REF-Router1
TAR-SGW1
REF-Router2
REF-Host2
ICMP Echo Request
ICMP Error Message to REF-Host2
(Packet Too Big)
(Observable Result – Step 4)
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
ICMP Error Message to TAR-EN1
(Packet Too Big)
(Observable Result – Step 6)
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 1
Fragmented ICMP Echo Reply 2
Fragmented ICMP Echo Reply with ESP 1
Fragmented ICMP Echo Reply with ESP 2
(Observable Result – Step 8)

Procedure

Step	Action	Expected Result
1.	Configure the Network1 interface of REF-Router1 and the Network1 interface of TAR-SGW1 with a path MTU of 1280 bytes	
2.	Initialize the devices	
3.	Transmit *ICMP Echo Request* from REF-Host2 to the Global unicast address of TAR-EN1	
4.	Observe the packets transmitted on Network0, Network1, Network2, and Network3	TAR-SGW1 transmits ICMP Error Message (Packet Too Big) to REF-Host2
5.	Transmit *Fragmented ICMP Echo Request 1* and *Fragmented ICMP Echo Request 2* from REF-Host2 to the Global unicast address of TAR-EN1	
6.	Observe the packets transmitted on Network0, Network1, Network2, and Network3	TAR-SGW1 transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 TAR-EN1 transmits ICMP Echo Reply with ESP REF-Router1 transmits ICMP Error Message (Packet Too Big) to TAR-EN1
7.	Transmit *Fragmented ICMP Echo Request 1* and *Fragmented ICMP Echo Request 2* from REF-Host2 to the Global unicast address of TAR-EN1	
8.	Observe the packets transmitted on Network0, Network1, Network2, and Network3	TAR-SGW1 transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 TAR-EN1 transmits Fragmented ICMP Echo Reply with ESP 1 and Fragmented ICMP Echo Reply with ESP 2 TAR-SGW1 transmits Fragmented ICMP Echo Reply 1 and Fragmented ICMP Echo Reply 2

Possible Problems

When transmitting the packet *Fragmented ICMP Echo Request with ESP 1 or Fragmented ICMP Echo Reply with ESP 1*, TAR-SGW1 or TAR-EN1 may further fragment these packets.

Section 4: Tunnel Mode (SGW vs. SGW)

4.1: Tunnel Mode: SGW vs. SGW

Scope

The following tests focus on Tunnel Mode

Overview

Tests in this section verify that a node properly processes and transmits the packets to which IPsec Tunnel Mode is applied between two SGWs.

IPsec.IO.4.1.1. ESP Algorithms

Purpose

Verify ESP Algorithms in Tunnel Mode between two SGWs

Initialization

- Network Topology
  - Connect the devices according to Common Topology 2
- Configuration
  - In each part, configure the devices according to the ESP Algorithms, and Configuration 3

Packets

IP Header	Source Address	TAR-SGW2_Network2
	Destination Address	TAR-SGW1_Network1
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1
	Encrypted Data/ICV	TAR-SGW1_SA-I/TAR-SGW2_SA-O
IP Header	Source Address	REF-Host2_Network3
	Destination Address	REF-Host1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	REF-Host2_Network3
	Destination Address	REF-Host1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

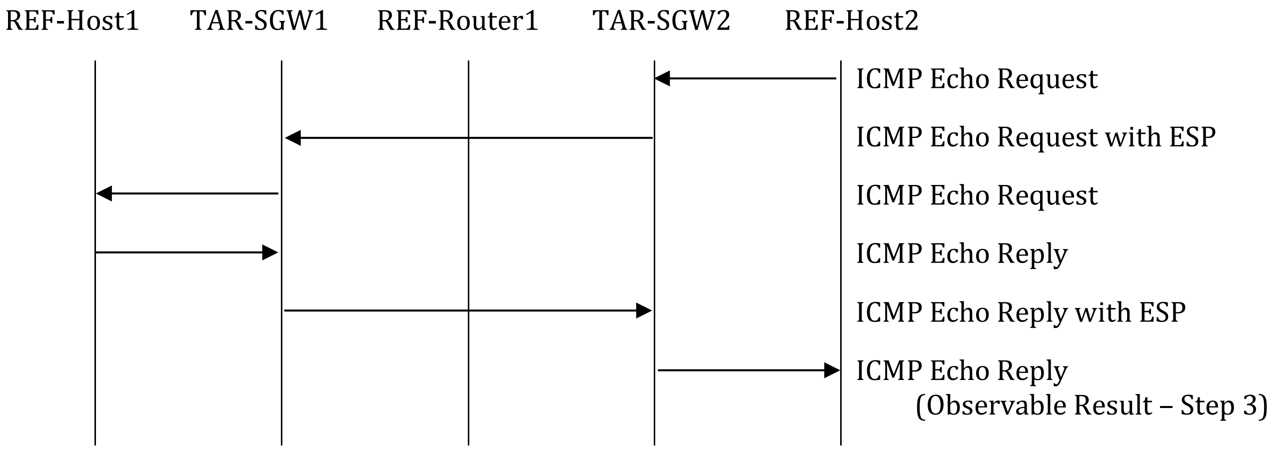
IP Header	Source Address	REF-Host1_Network0
	Destination Address	REF-Host2_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

IP Header	Source Address	TAR-SGW1_Network1
	Destination Address	TAR-SGW2_Network2
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-SGW1_SA-O/TAR-SGW2_SA-I
IP Header	Source Address	REF-Host1_Network0
	Destination Address	REF-Host2_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure



All Parts: Algorithms

Step	Action	Expected Result
1.	Initialize the devices	
2.	Transmit ICMP Echo Request from REF-Host2 to the Global unicast address of REF-Host1	
3.	Observe the packets transmitted on all networks	TAR-SGW2 transmits ICMP Echo Request with ESP TAR-SGW1 transmits *ICMP Echo Request* REF-Host1 transmits *ICMP Echo Reply* TAR-SGW1 transmits ICMP Echo Reply with ESP TAR-SGW2 transmits *ICMP Echo Reply*

Possible Problems

None

IPsec.IO.4.1.2. Fragmentation

Purpose

Verify IPv6 Packet Too Big Processing, Fragmentation, and Reassembly in Tunnel Mode between two SGWs

Initialization

- Network Topology
  - Connect the devices according to Common Topology 2
- Configuration
  - In each part, configure the devices according to the Global Security Associations, and Configuration 3

Packets

IP Header	Source Address	REF-Host2_Network3
	Destination Address	REF-Host1_Network0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	TAR-SGW2
	Destination Address	REF-Host2
ICMP	Type	2 (Packet Too Big)
	MTU	1280
	Data	1232Byte of ICMP Echo Request

ICMP Error Message to REF-Host2 (Packet Too Big)

IP Header	Source Address	REF-Host2_Network3
	Destination Address	REF-Host1_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	REF-Host2_Network3
	Destination Address	REF-Host1_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	TAR-SGW2_Network2
	Destination Address	TAR-SGW1_Network1
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1

	Encrypted Data/ICV	TAR-SGW1_SA-I/TAR-SGW2_SA-O
IP Header	Source Address	REF-Host2_Network3
	Destination Address	REF-Host1_Network0
	Payload Length	1stPL (= MTU-40) (e.g. 1240)
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP

IP Header	Source Address	TAR-SGW2_Network2
	Destination Address	TAR-SGW1_Network1
ESP	SPI	*Dynamic1 or* 0x1000
	Sequence	1
	Encrypted Data/ICV	TAR-SGW1_SA-I/TAR-SGW2_SA-O
IP Header	Source Address	REF-Host2_Network3
	Destination Address	REF-Host1_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 with ESP 2

IP Header	Source Address	REF-Host1_Network0
	Destination Address	REF-Host2_Network3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

IP Header	Source Address	TAR-SGW1_Network1
	Destination Address	TAR-SGW2_Network2
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-SGW1_SA-O/TAR-SGW2_SA-I
IP Header	Source Address	REF-Host1_Network0
	Destination Address	REF-Host2_Network3
	Payload Length	1460
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	REF-Router1
	Destination Address	TAR-SGW1
ICMP	Type	2 (Packet Too Big)
	MTU	1280
	Data	1232Byte of ICMP Echo Reply with ESP

ICMP Error Message to TAR-SGW1 (Packet Too Big)

IP Header	Source Address	TAR-SGW1
	Destination Address	REF-Host1_Network0
ICMP	Type	2 (Packet Too Big)
	MTU	1280
	Data	1232Byte of ICMP Echo Reply

ICMP Error Message to REF-Host1 (Packet Too Big)

IP Header	Source Address	REF-Host1_Network0
	Destination Address	REF-Host2_Network3
	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	REF-Host1_Network0
	Destination Address	REF-Host2_Network3
	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

IP Header	Source Address	TAR-SGW1_Network1
	Destination Address	TAR-SGW2_Network2
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-SGW1_SA-O/TAR-SGW2_SA-I
IP Header	Source Address	REF-Host1_Network0
	Destination Address	REF-Host2_Network3
	Payload Length	1stPL (= MTU-40) (e.g. 1240)
Fragment	Offset	0
	More Flag	1
ICMP	Type	129 (Echo Reply)

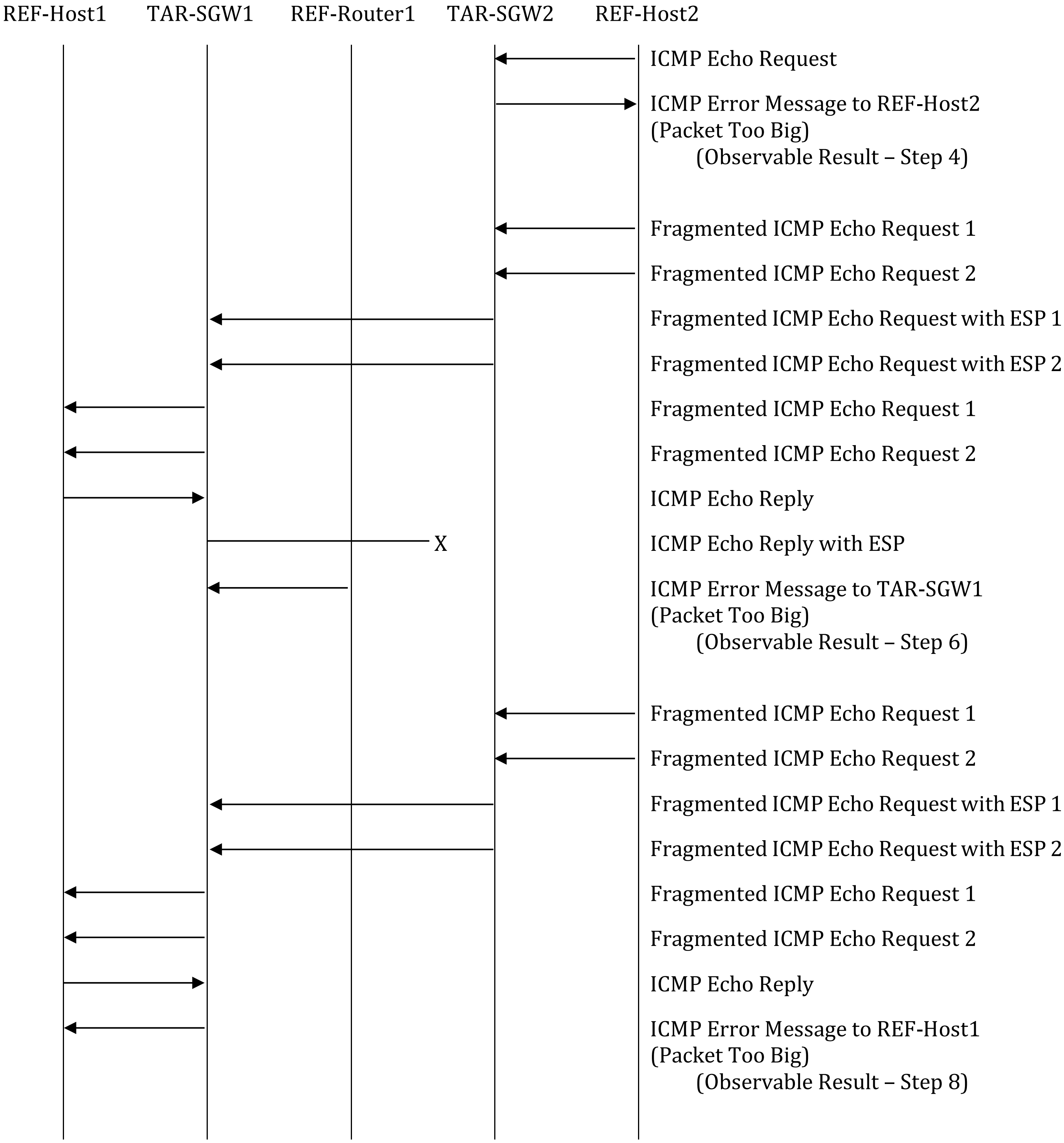
Fragmented ICMP Echo Reply with ESP 1

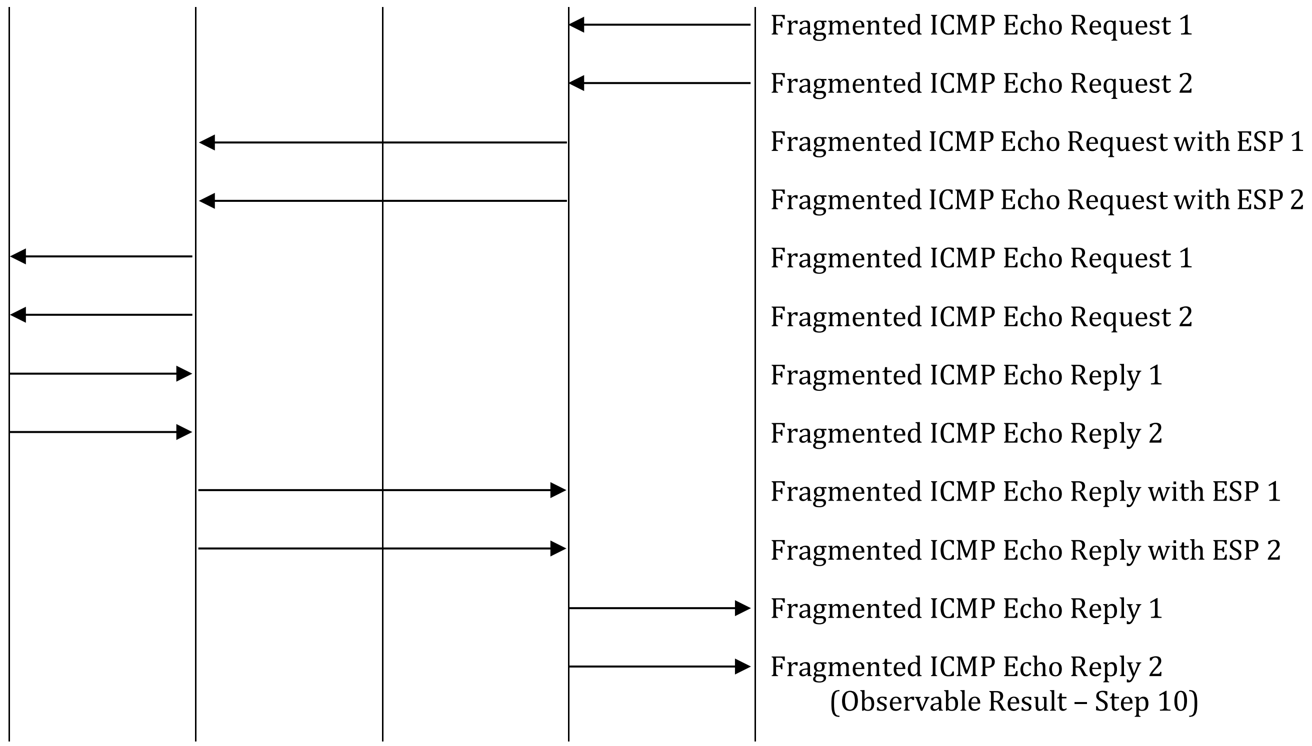
IP Header	Source Address	TAR-SGW1_Network1
	Destination Address	TAR-SGW2_Network2
ESP	SPI	*Dynamic2 or* 0x2000
	Sequence	1
	Encrypted Data/ICV	TAR-SGW1_SA-O/TAR-SGW2_SA-I
IP Header	Source Address	REF-Host1_Network0
	Destination Address	REF-Host2_Network3
	Payload Length	2ndPL (= 1476-1stPL)
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of ICMP Echo Reply

Fragmented ICMP Echo Reply with ESP 2

Procedure

Part A: TAR-SGW1 Packet Too Big Processing

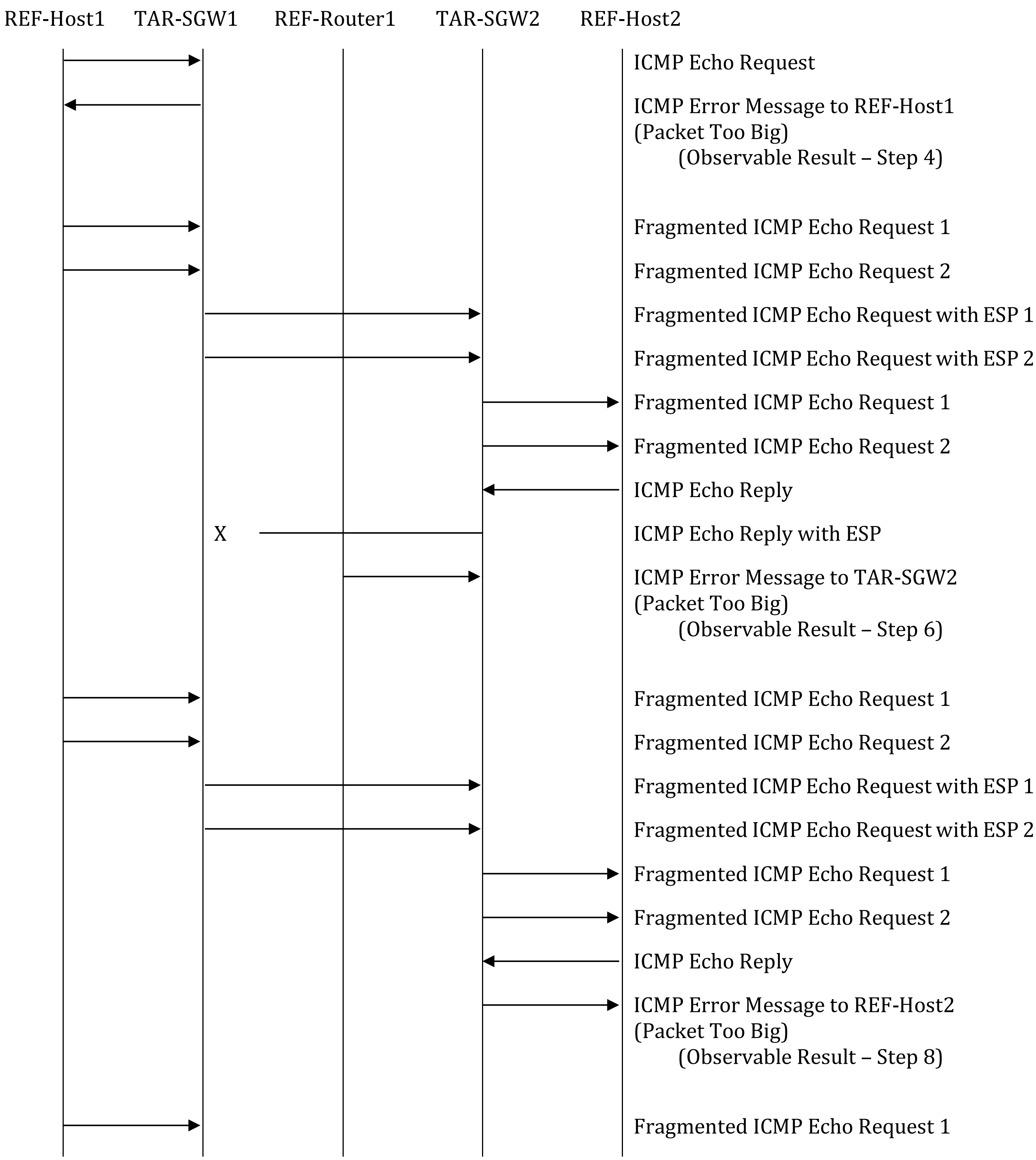


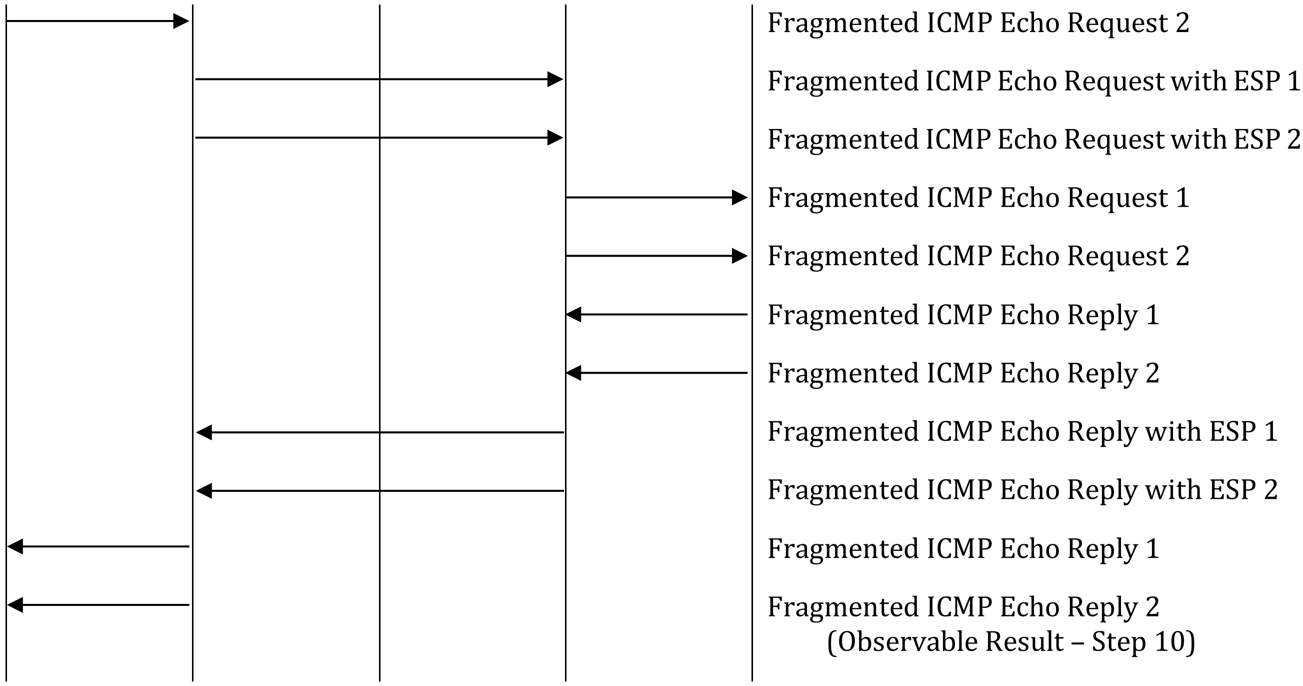

Fragmented ICMP Echo Request 1
Fragmented ICMP Echo Request 2
Fragmented ICMP Echo Request with ESP 1
Fragmented ICMP Echo Request with ESP 2
Fragmented ICMP Echo Request 1
Fragmented ICMP Echo Request 2
Fragmented ICMP Echo Reply 1
Fragmented ICMP Echo Reply 2
Fragmented ICMP Echo Reply with ESP 1
Fragmented ICMP Echo Reply with ESP 2
Fragmented ICMP Echo Reply 1
Fragmented ICMP Echo Reply 2
(Observable Result – Step 10)

Step	Action	Expected Result
1.	Configure the Network2 interface of REF-Router1 and the Network2 interface of TAR-SGW2 with a path MTU of 1280 bytes	
2.	Initialize the Devices	
3.	Transmit *ICMP Echo Request* from REF-Host2 to the Global unicast address of REF-Host1	
4.	Observe the packets transmitted on Network0, Network1, Network2, and Network3	TAR-SGW2 transmits ICMP Error Message (Packet Too Big) to REF-Host2
5.	Transmit *Fragmented ICMP Echo Request 1* and *Fragmented ICMP Echo Request 2* from REF-Host2 to the Global unicast address of REF-Host1	
6.	Observe the packets transmitted on Network0, Network1, Network2, and Network3	TAR-SGW2 transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 TAR-SGW1 transmits Fragmented ICMP Echo Request 1 and Fragmented ICMP Echo Request 2 REF-Host1 transmits *ICMP Echo Reply* TAR-SGW1 transmits ICMP Echo Reply with ESP REF-Router1 transmits ICMP Error Message (Packet Too Big) to TAR-SGW1
7.	Transmit *Fragmented ICMP Echo Request 1* and *Fragmented ICMP Echo Request 2* from REF-Host2 to the Global unicast address of REF-Host1	
8.	Observe the packets transmitted on Network0, Network1, Network2, and Network3	TAR-SGW2 transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 TAR-SGW1 transmits Fragmented ICMP Echo Request 1 and Fragmented ICMP Echo Request 2 REF-Host1 transmits *ICMP Echo Reply*

		TAR-SGW1 transmits ICMP Error Message (Packet Too Big) to REF-Host1
9.	Transmit *Fragmented ICMP Echo Request 1* and *Fragmented ICMP Echo Request 2* from REF-Host2 to the Global unicast address of REF-Host1	
10.	Observe the packets transmitted on Network0, Network1, Network2, and Network3	TAR-SGW2 transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 TAR-SGW1 transmits Fragmented ICMP Echo Request 1 and Fragmented ICMP Echo Request 2 REF-Host1 transmits Fragmented ICMP Echo Reply 1 and Fragmented ICMP Echo Reply 2 TAR-SGW1 transmits Fragmented ICMP Echo Reply with ESP 1 and Fragmented ICMP Echo Reply with ESP 2 TAR-SGW2 transmits Fragmented ICMP Echo Reply 1 and Fragmented ICMP Echo Reply 2

Part B: TAR-SGW2 Packet Too Big Processing




Fragmented ICMP Echo Request 2
Fragmented ICMP Echo Request with ESP 1
Fragmented ICMP Echo Request with ESP 2
Fragmented ICMP Echo Request 1
Fragmented ICMP Echo Request 2
Fragmented ICMP Echo Reply 1
Fragmented ICMP Echo Reply 2
Fragmented ICMP Echo Reply with ESP 1
Fragmented ICMP Echo Reply with ESP 2
Fragmented ICMP Echo Reply 1
Fragmented ICMP Echo Reply 2
(Observable Result – Step 10)

Step	Action	Expected Result
11.	Configure the Network1 interface of REF-Router1 and the Network1 interface of TAR-SGW1 with a path MTU of 1280 bytes	
12.	Initialize the Devices	
13.	Transmit *ICMP Echo Request* from REF-Host1 to the Global unicast address of REF-Host2	
14.	Observe the packets transmitted on Network0, Network1, Network2, and Network3	TAR-SGW1 transmits ICMP Error Message (Packet Too Big) to REF-Host1
15.	Transmit *Fragmented ICMP Echo Request 1* and *Fragmented ICMP Echo Request 2* from REF-Host1 to the Global unicast address of REF-Host2	
16.	Observe the packets transmitted on Network0, Network1, Network2, and Network3	TAR-SGW1 transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 TAR-SGW2 transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 REF-Host2 transmits *ICMP Echo Reply* TAR-SGW2 transmits ICMP Echo Reply with ESP REF-Router1 transmits ICMP Error Message (Packet Too Big) to TAR-SGW2
17.	Transmit *Fragmented ICMP Echo Request 1* and *Fragmented ICMP Echo Request 2* from REF-Host1 to the Global unicast address of REF-Host2	
18.	Observe the packets transmitted on Network0, Network1, Network2, and Network3	TAR-SGW1 transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 TAR-SGW2 transmits Fragmented ICMP Echo Request 1 and Fragmented ICMP Echo Request 2 REF-Host2 transmits *ICMP Echo*

		Reply Step-8: TAR-SGW2 transmits *ICMP Error Message (Packet Too Big)* to REF-Host2
19.	Transmit *Fragmented ICMP Echo Request 1* and *Fragmented ICMP Echo Request 2* from REF-Host1 to the Global unicast address of REF-Host2	
20.	Observe the packets transmitted on Network0, Network1, Network2, and Network3	TAR-SGW1 transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 TAR-SGW2 transmits Fragmented ICMP Echo Request 1 and Fragmented ICMP Echo Request 2 REF-Host2 transmits Fragmented ICMP Echo Reply 1 and Fragmented ICMP Echo Reply 2 TAR-SGW2 transmits Fragmented ICMP Echo Reply with ESP 1 and Fragmented ICMP Echo Reply with ESP 2 TAR-SGW1 transmits Fragmented ICMP Echo Reply 1 and Fragmented ICMP Echo Reply 2

Possible Problems

None

Appendix-A Required Data

When you apply for an IPv6 Ready Logo Phase-2(IPsec) you need to submit test logs. In this appendix the detail requirement for the test log is described.

Required Data Type

As "IPv6 Ready Logo Phase-2" the following interoperability test result data are required.

A) Topology map

Network topology figures or address list, with IPv6 addresses and MAC address of each attached interfaces, are required. Fig.4 is an example of topology figure.

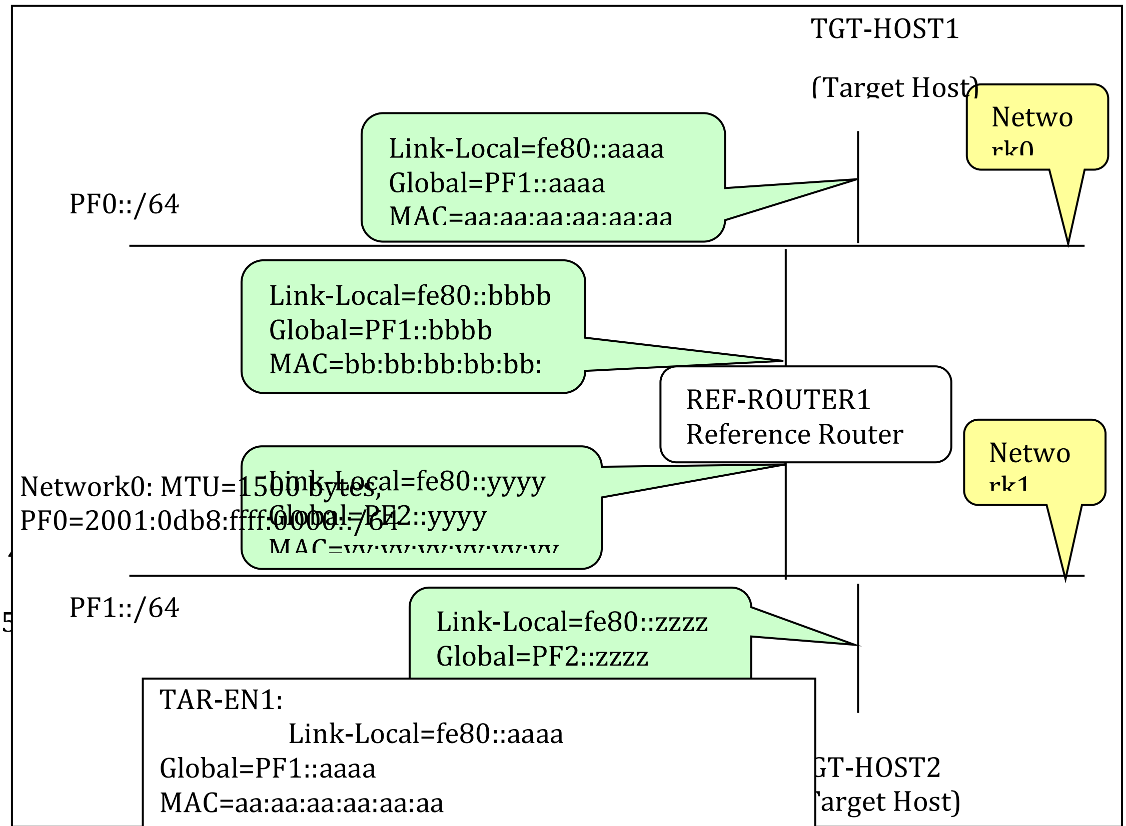


Fig. 4

Fig.5

TAR-EN1:
Link-Local=fe80::aaaa
Global=PF1::aaaa
MAC=aa:aa:aa:aa:aa:aa
REF-Router1 [Network0]:
Link-Local=fe80::bbbb
Global=PF1::bbbb
MAC=bb:bb:bb:bb:bb:bb
REF-Router1 [Network1]:
Link-Local=fe80::yyyy
Global=PF2::yyyy
MAC=yy:yy:yy:yy:yy:yy
TAR-EN2:
Link-Local=fe80::zzzz
Global=PF2::zzzz
MAC=zz:zz:zz:zz:zz:zz

Fig. 5 Address

B) Command Log

Ping is used as default application. When you run test with ping application, please save the command log into individual files.

We allow using other protocol than ICMP Echo Request and Reply. Even though you use other kind of application, please save the command log.

Save the command files for each test on each node.

C) Packet Capture File

Capture all packets on each link during the test with a device that is not part of the test.

Make individual tcpdump(pcap) format file for each test and link or put the packet dump in a readable HTML file.

If you run tcpdump, please specify packet size as 4096.

e.g.,) tcpdump -i if0 -s 4096 –w 5.1.A.VendorA.Network0.dump

D) Test Result Table

Collect all test result tables in a file and fill the tables as required. This file must contain a table where all passes are clearly marked.

E) Keying Information

Collect all SPD and SAD information. If you configure keying information manually, it is not required to submit keying information. Fig. 6 is an example of Keying Information.

```
TAR-EN1's SAD1:
              Source Address: TAR-EN1_Network0
Destination Address: TAR-EN2_Network1
SPI: 0x1000
mode: transport
protocol: ESP
ESP algorithm: 3DES-CBC
ESP key: ipv6readylogo3descbc1to2
ESP authentication: HMAC-SHA1
ESP authentication key: ipv6readylogsha11to2
TAR-EN1's SAD2:
              Source Address: TAR-EN2_Network1
Destination Address: TAR-EN1_Network0
SPI: 0x2000
mode: transport
protocol: ESP
ESP algorithm: 3DES-CBC
ESP key: ipv6readylogo3descbc2to1
ESP authentication: HMAC-SHA1
ESP authentication key: ipv6readylogsha12to1
TAR-EN1's SPD1:
              Source Address: TAR-EN1_Network0
Destination Address: TAR-EN2_Network1
upper spec: any
direction: out
protocol: ESP
mode: transport
TAR-EN1's SPD2:
              Source Address: TAR-EN1_Network0
Destination Address: TAR-EN2_Network1
upper spec: any
direction: in
protocol: ESP
mode: transport
```

Fig. 6 Keying In

Data file name syntax

Please use following syntax in the file name.

A) Topology Map

Syntax: Chapter.Section.Sub_section.ON.topology
For "ON", use the Node's vendor name which behaved as a Opposite side target Node(ON).
e.g.,)
5.1.1 Transport Mode ESP=3DES-CBC HMAC-SHA1
TAR-EN1 (Your Device):
End-Node [vendor: VendorX, model: rHost1, version: 1.0]
TAR-EN2 (Opposite side device):
End-Node [vendor: VendorA, model: rHost2, version: 2.0]

5.1.1.VendorA.topology

B) Command Results

Syntax: Chapter.Section.Sub_Section.SRC.DST.result
For "*SRC*", use the vendor name on which the commands were run. If SRC is a Reference Host, just specify REF_HOST*n* as SRC. For "*DST*", use the vendor name to which the commands were run, in other word, destination of ping command. If DST is a Reference Host, just specify REF_HOST*n* as DST
e.g.,)
Typical Naming sample are following.

5.1.1 Transport Mode ESP=3DES-CBC HMAC-SHA1
TAR-EN1: End-Node [vendor: VendorA, model: rHost1, version: 1.0]
TAR-EN2: End-Node [vendor: VendorB, model: rHost2, version: 2.0]

5.1.1.VendorB.VendorA.result

5.2.1 Tunnel Mode ESP=3DES-CBC HMAC-SHA1
TAR-SGW1: SGW [vendor: VendorA, model: rRouter1, version: 1.0]
TAR-SGW2: SGW [vendor: VendorB, model: rRouter2, version: 2.0]
REF-Host1: Host [vendor: VendorC, model: rHost1, version: 1.0]
REF-Host2: Host [vendor: VendorD, model: rHost2, version: 2.0]

5.2.1.REF-Host2.REF-Host1.result

C) Captured packet file

Syntax:Chapter.Section.Sub_Section.ON.Link.dump
For "*Link*", use the captured link name.
For "ON", use the Node's vendor name which behaved as a Opposite side target Node(ON).

Even if the command run on a Reference Node, you should list ON's vendor name rather than REF_HOST*n*.
e.g.,)
5.1.1 Transport Mode ESP=3DES-CBC HMAC-SHA1
TAR-EN1 (Your Device):
End-Node [vendor: VendorX, model: rHost1, version: 1.0]
TAR-EN2 (Opposite side device):
End-Node [vendor: VendorA, model: rHost2, version: 2.0]

5.1.1.VendorA.Network0.dump
5.1.1.VendorA.Network1.dump

D) Test Result Table

Syntax: Vendor.table
In this file you must make table for each sub-section.

For End-Node)

- Transport Mode (BASIC): Test 5.1.X is required.

For Test 5.1.X

	VendorA (End-Node)	VendorB (End-Node)
Applicants_name (End-Node)		

- Tunnel Mode (ADVANCED): Test 5.3.X or Test 5.4.X is required.

For Test 5.3.X

	VendorC (SGW)	VendorD (SGW)
Applicants_name (End Node)		

or

For Test 5.4.X

	VendorC (End Node)	VendorD (End Node)
Applicants_name (End Node)		

For SGW)

- Tunnel Mode (BASIC): Test 5.2.X or Test 5.3.X is required.

For Test 5.2.X

	VendorA (SGW)	VendorB (SGW)
Applicants_name (SGW)		

or

For Test 5.3.X

	VendorA (End-Node)	VendorB (End-Node)
Applicants_name (SGW)		

e.g.,)
Test result of following host.
TAR-EN1:
End-Node [vendor: VendorX, model: rHost1, version: 1.0]
VendorX.table

E) Keying Information

Syntax: Chapter.Section.Sub_Section.ON.key
For "ON", use the Node's vendor name which behaved as a Opposite side target Node(ON).
e.g.,)
5.1.1 Transport Mode ESP=3DES-CBC HMAC-SHA1
TAR-EN1 (Your Device):
End-Node [vendor: VendorX, model: rHost1, version: 1.0]
TAR-EN2 (Opposite side device):
End-Node [vendor: VendorA, model: rHost2, version: 2.0]

5.1.1.VendorA.key

Data Archive

Please organize your data as following directory structure.
$YourDeviceName_ver/
Conformance/
Interoperability/
Put all interoperability data file in "Interoperability" directory.
Put all conformance Self-Test results or conformance Lab test results in "Conformance" directory.
Make a tar.gz format archive file, and put all files under "$YourDeviceName_ver" in it.

1) File list for End-Node

1-1) In the case of supporting only transport mode

Test 5.1 is performed with following condition.
TAR-EN1:

Your device:	VendorX (End-Node)

TAR-EN2:

Counterpart End-Node 1:	VendorA
Counterpart End-Node 2:	VendorB

The file list is described below.

```
${Your_Device_ver}/
 | Conformance/
 |    | ...
 |    |
 | Interoperability/
 |    | YYYY_MM
 |    |    | Round1_VendorA
 |    |    |    | Captures
 |    |    |    |    | 5_1_[01-10/12]_VendorX_VendorA_Network0.pcap
 |    |    |    |    | 5_1_[01-10/12]_VendorX_VendorA_Network1.pcap
 |    |    |    |    | 5_1_11[A-B]_VendorX_VendorA_Network0.pcap
 |    |    |    |    | 5_1_11[A-B]_VendorX_VendorA_Network1.pcap
 |    |    |    | Results
 |    |    |    |    | 5_1_[01-07/09-10/12]_VendorA_VendorX.result
 |    |    |    |    | 5_1_08_VendorA_VendorX.result
 |    |    |    |    | 5_1_08_VendorX_VendorA.result
 |    |    |    |    | 5_1_11A_VendorA_VendorX.result
 |    |    |    |    | 5_1_11B_VendorX_VendorA.result
 |    |    | Round2_VendorB
 |    |    |    | Captures
 |    |    |    |    | 5_1_[01-10/12]_VendorX_VendorB_Network0.pcap
 |    |    |    |    | 5_1_[01-10/12]_VendorX_VendorB_Network1.pcap
 |    |    |    |    | 5_1_11[A-B]_VendorX_VendorB_Network0.pcap
 |    |    |    |    | 5_1_11[A-B]_VendorX_VendorB_Network1.pcap
 |    |    |    | Results
 |    |    |    |    | 5_1_[01-07/09-10/12]_VendorB_VendorX.result
```

```
|   |   |   |   | 5_1_08_VendorB_VendorX.result
|   |   |   |   | 5_1_08_VendorX_VendorB.result
|   |   |   |   | 5_1_11A_VendorB_VendorX.result
|   |   |   |   | 5_1_11B_VendorX_VendorB.result
```

1-2) In the case of supporting transport mode and tunnel mode

1-2-1) In the case of choosing SGW as the counterpart device

Test 5.1 is performed with following condition.
TAR-EN1:

Your device:	VendorX (End Node)

TAR-EN2:

Counterpart End Node 1:	VendorA
Counterpart End Node 2:	VendorB

Test 5.3 is performed with following condition.
TAR-EN1:

Your device:	VendorX (End Node)

TAR-SGW1:

Counterpart SGW 1:	VendorC
Counterpart SGW 2:	VendorD

The file list is described below.

```
${Your_Device_ver}/
 | Conformance/
 |   | ...
 |   |
 | Interoperability/
 |   | YYYY_MM
 |   |   | Round1_VendorA
 |   |   |   | Captures
 |   |   |   |   | 5_1_[01-10/12]_VendorX_VendorA_Network0.pcap
 |   |   |   |   | 5_1_[01-10/12]_VendorX_VendorA_Network1.pcap
 |   |   |   |   | 5_1_11[A-B]_VendorX_VendorA_Network0.pcap
 |   |   |   |   | 5_1_11[A-B]_VendorX_VendorA_Network1.pcap
 |   |   |   | Results
 |   |   |   |   | 5_1_[01-07/09-10/12]_VendorA_VendorX.result
 |   |   |   |   | 5_1_08_VendorA_VendorX.result
 |   |   |   |   | 5_1_08_VendorX_VendorA.result
 |   |   |   |   | 5_1_11A_VendorA_VendorX.result
 |   |   |   |   | 5_1_11B_VendorX_VendorA.result
 |   |   | Round2_VendorB
 |   |   |   | Captures
 |   |   |   |   | 5_1_[01-10/12]_VendorX_VendorB_Network0.pcap
 |   |   |   |   | 5_1_[01-10/12]_VendorX_VendorB_Network1.pcap
```

```
|   |   |   |   | 5_1_11[A-B]_VendorX_VendorB_Network0.pcap
|   |   |   |   | 5_1_11[A-B]_VendorX_VendorB_Network1.pcap
|   |   |   | Results
|   |   |   |   | 5_1_[01-07/09-10/12]_VendorB_VendorX.result
|   |   |   |   | 5_1_08_VendorB_VendorX.result
|   |   |   |   | 5_1_08_VendorX_VendorB.result
|   |   |   |   | 5_1_11A_VendorB_VendorX.result
|   |   |   |   | 5_1_11B_VendorX_VendorB.result
|   |   | Round3_VendorC
|   |   |   | Captures
|   |   |   |   | 5_3_[01-10/12]_VendorX_VendorC_Network0.pcap
|   |   |   |   | 5_3_[01-10/12]_VendorX_VendorC_Network1.pcap
|   |   |   |   | 5_3_[01-10/12]_VendorX_VendorC_Network2.pcap
|   |   |   |   | 5_3_11[A-B]_VendorX_VendorC_Network0.pcap
|   |   |   |   | 5_3_11[A-B]_VendorX_VendorC_Network1.pcap
|   |   |   |   | 5_3_11[A-B]_VendorX_VendorC_Network2.pcap
|   |   |   |   | 5_3_11[A-B]_VendorX_VendorC_Network3.pcap
|   |   |   | Results
|   |   |   |   | 5_3_[01-07/09-10/12]_REF-Host2_VendorX.result
|   |   |   |   | 5_3_08_REF-Host2_VendorX.result
|   |   |   |   | 5_3_08_VendorX_REF-Host2.result
|   |   |   |   | 5_3_11[A-B]_REF-Host2_VendorX.result
|   |   | Round4_VendorD
|   |   |   | Captures
|   |   |   |   | 5_3_[01-10/12]_VendorX_VendorD_Network0.pcap
|   |   |   |   | 5_3_[01-10/12]_VendorX_VendorD_Network1.pcap
|   |   |   |   | 5_3_[01-10/12]_VendorX_VendorD_Network2.pcap
|   |   |   |   | 5_3_11[A-B]_VendorX_VendorD_Network0.pcap
|   |   |   |   | 5_3_11[A-B]_VendorX_VendorD_Network1.pcap
|   |   |   |   | 5_3_11[A-B]_VendorX_VendorD_Network2.pcap
|   |   |   |   | 5_3_11[A-B]_VendorX_VendorD_Network3.pcap
|   |   |   | Results
|   |   |   |   | 5_3_[01-07/09-10/12]_REF-Host2_VendorX.result
|   |   |   |   | 5_3_08_REF-Host2_VendorX.result
|   |   |   |   | 5_3_08_VendorX_REF-Host2.result
|   |   |   |   | 5_3_11[A-B]_REF-Host2_VendorX.result
```

1-2-2) In the case of choosing End-Node as the counterpart device

Test 5.1 is performed with following condition.
TAR-EN1:

Your device:	VendorX (End-Node)

TAR-EN2:

Counterpart End Node 1 for the transport mode:	VendorA
Counterpart End Node 2 for the transport mode:	VendorB

Test 5.4 is performed with following condition.

TAR-EN1:

Your device:	VendorX (End-Node)

TAR-EN2:

Counterpart End-Node 3 for the tunnel mode:	VendorC
Counterpart End-Node 4 for the tunnel mode:	VendorD

The file list is described below.

```
${Your_Device_ver}/
 | Conformance/
 |    | ...
 |    |
 | Interoperability/
 |    | YYYY_MM
 |    |    | Round1_VendorA
 |    |    |    | Captures
 |    |    |    |    | 5_1_[01-10/12]_VendorX_VendorA_Network0.pcap
 |    |    |    |    | 5_1_[01-10/12]_VendorX_VendorA_Network1.pcap
 |    |    |    |    | 5_1_11[A-B]_VendorX_VendorA_Network0.pcap
 |    |    |    |    | 5_1_11[A-B]_VendorX_VendorA_Network1.pcap
 |    |    |    | Results
 |    |    |    |    | 5_1_[01-07/09-10/12]_VendorA_VendorX.result
 |    |    |    |    | 5_1_08_VendorA_VendorX.result
 |    |    |    |    | 5_1_08_VendorX_VendorA.result
 |    |    |    |    | 5_1_11A_VendorA_VendorX.result
 |    |    |    |    | 5_1_11B_VendorX_VendorA.result
 |    |    | Round2_VendorB
 |    |    |    | Captures
 |    |    |    |    | 5_1_[01-10/12]_VendorX_VendorB_Network0.pcap
 |    |    |    |    | 5_1_[01-10/12]_VendorX_VendorB_Network1.pcap
 |    |    |    |    | 5_1_11[A-B]_VendorX_VendorB_Network0.pcap
 |    |    |    |    | 5_1_11[A-B]_VendorX_VendorB_Network1.pcap
 |    |    |    | Results
 |    |    |    |    | 5_1_[01-07/09-10/12]_VendorB_VendorX.result
 |    |    |    |    | 5_1_08_VendorB_VendorX.result
 |    |    |    |    | 5_1_08_VendorX_VendorB.result
 |    |    |    |    | 5_1_11A_VendorB_VendorX.result
 |    |    |    |    | 5_1_11B_VendorX_VendorB.result
 |    |    | Round3_VendorC
 |    |    |    | Captures
 |    |    |    |    | 5_4_[01-10/12]_VendorX_VendorA_Network0.pcap
 |    |    |    |    | 5_4_[01-10/12]_VendorX_VendorA_Network1.pcap
 |    |    |    |    | 5_4_11[A-B]_VendorX_VendorA_Network0.pcap
 |    |    |    |    | 5_4_11[A-B]_VendorX_VendorA_Network1.pcap
 |    |    |    | Results
 |    |    |    |    | 5_4_[01-07/09-10/12]_VendorA_VendorX.result
 |    |    |    |    | 5_4_08_VendorA_VendorX.result
```

```
|    |    |    |    | 5_4_08_VendorX_VendorA.result
|    |    |    |    | 5_4_11A_VendorA_VendorX.result
|    |    |    |    | 5_4_11B_VendorX_VendorA.result
|    |    | Round4_VendorD
|    |    |    | Captures
|    |    |    |    | 5_4_[01-10/12]_VendorX_VendorB_Network0.pcap
|    |    |    |    | 5_4_[01-10/12]_VendorX_VendorB_Network1.pcap
|    |    |    |    | 5_4_11[A-B]_VendorX_VendorB_Network0.pcap
|    |    |    |    | 5_4_11[A-B]_VendorX_VendorB_Network1.pcap
|    |    |    | Results
|    |    |    |    | 5_4_[01-07/09-10/12]_VendorB_VendorX.result
|    |    |    |    | 5_4_08_VendorB_VendorX.result
|    |    |    |    | 5_4_08_VendorX_VendorB.result
|    |    |    |    | 5_4_11A_VendorB_VendorX.result
|    |    |    |    | 5_4_11B_VendorX_VendorB.result
```

2) File list for SGW

2-1) In the case of choosing SGW as the counterpart device

Test 5.2 is performed with following condition.

TAR-SGW1/TAR-SGW2:

Your device:	VendorX (SGW)

TAR-SGW1/TAR-SGW2:

Counterpart SGW 1:	VendorA
Counterpart SGW 2:	VendorB

The file list is described below.

```
${Your_Device_ver}/
| Conformance/
|    | ...
|    |
| Interoperability/
|    | YYYY_MM
|    |    | Round1_VendorA
|    |    |    | Captures
|    |    |    |    | 5_2_[01-10/12]_VendorX_VendorA_Network0.pcap
|    |    |    |    | 5_2_[01-10/12]_VendorX_VendorA_Network1.pcap
|    |    |    |    | 5_2_[01-10/12]_VendorX_VendorA_Network2.pcap
|    |    |    |    | 5_2_[01-10/12]_VendorX_VendorA_Network3.pcap
|    |    |    |    | 5_2_11[A-B]_VendorX_VendorA_Network0.pcap
|    |    |    |    | 5_2_11[A-B]_VendorX_VendorA_Network1.pcap
|    |    |    |    | 5_2_11[A-B]_VendorX_VendorA_Network2.pcap
|    |    |    |    | 5_2_11[A-B]_VendorX_VendorA_Network3.pcap
|    |    |    | Results
|    |    |    |    | 5_2_[01-07/09-10/12]_REF-Host2_REF-Host1.result
|    |    |    |    | 5_2_08_REF-Host2_REF-Host1.result
```

```
|    |    |    |    | 5_2_08_REF-Host1_REF-Host2.result
|    |    |    |    | 5_2_11A_REF-Host2_REF-Host1.result
|    |    |    |    | 5_2_11B_REF-Host1_REF-Host2.result
|    |    | Round2_VendorB
|    |    |    | Captures
|    |    |    |    | 5_2_[01-10/12]_VendorX_VendorB_Network0.pcap
|    |    |    |    | 5_2_[01-10/12]_VendorX_VendorB_Network1.pcap
|    |    |    |    | 5_2_[01-10/12]_VendorX_VendorB_Network2.pcap
|    |    |    |    | 5_2_[01-10/12]_VendorX_VendorB_Network3.pcap
|    |    |    |    | 5_2_11[A-B]_VendorX_VendorB_Network0.pcap
|    |    |    |    | 5_2_11[A-B]_VendorX_VendorB_Network1.pcap
|    |    |    |    | 5_2_11[A-B]_VendorX_VendorB_Network2.pcap
|    |    |    |    | 5_2_11[A-B]_VendorX_VendorB_Network3.pcap
|    |    |    | Results
|    |    |    |    | 5_2_[01-07/09-10/12]_REF-Host2_REF-Host1.result
|    |    |    |    | 5_2_08_REF-Host2_REF-Host1.result
|    |    |    |    | 5_2_08_REF-Host1_REF-Host2.result
|    |    |    |    | 5_2_11A_REF-Host2_REF-Host1.result
|    |    |    |    | 5_2_11B_REF-Host1_REF-Host2.result
```

2-2) In the case of choosing End-Node as the counterpart device

Test 5.3 is performed with following condition.
TAR-SGW1:

Your device:	VendorX (SGW)

TAR-EN1:

Counterpart End-Node 1:	VendorA
Counterpart End-Node 2:	VendorB

The file list is described below.

```
${Your_Device_ver}/
| Conformance/
|    | ...
|    |
| Interoperability/
|    | YYYY_MM
|    |    | Round1_VendorA
|    |    |    | Captures
|    |    |    |    | 5_3_[01-10/12]_VendorX_VendorA_Network0.pcap
|    |    |    |    | 5_3_[01-10/12]_VendorX_VendorA_Network1.pcap
|    |    |    |    | 5_3_[01-10/12]_VendorX_VendorA_Network2.pcap
|    |    |    |    | 5_3_11[A-B]_VendorX_VendorA_Network0.pcap
|    |    |    |    | 5_3_11[A-B]_VendorX_VendorA_Network1.pcap
|    |    |    |    | 5_3_11[A-B]_VendorX_VendorA_Network2.pcap
|    |    |    |    | 5_3_11[A-B]_VendorX_VendorA_Network3.pcap
|    |    |    | Results
```

```
|    |    |    |    | 5_3_[01-07/09-10/12]_REF-Host2_VendorA.result
|    |    |    |    | 5_3_08_REF-Host2_VendorA.result
|    |    |    |    | 5_3_08_VendorA_REF-Host2.result
|    |    |    |    | 5_3_11[A-B]_REF-Host2_VendorA.result
|    |    | Round2_VendorB
|    |    |    | Captures
|    |    |    |    | 5_3_[01-10/12]_VendorX_VendorB_Network0.pcap
|    |    |    |    | 5_3_[01-10/12]_VendorX_VendorB_Network1.pcap
|    |    |    |    | 5_3_[01-10/12]_VendorX_VendorB_Network2.pcap
|    |    |    |    | 5_3_11[A-B]_VendorX_VendorB_Network0.pcap
|    |    |    |    | 5_3_11[A-B]_VendorX_VendorB_Network1.pcap
|    |    |    |    | 5_3_11[A-B]_VendorX_VendorB_Network2.pcap
|    |    |    |    | 5_3_11[A-B]_VendorX_VendorB_Network3.pcap
|    |    |    | Results
|    |    |    |    | 5_3_[01-07/09-10/12]_REF-Host2_VendorB.result
|    |    |    |    | 5_3_08_REF-Host2_VendorB.result
|    |    |    |    | 5_3_08_VendorB_REF-Host2.result
|    |    |    |    | 5_3_11[A-B]_REF-Host2_VendorB.result
```

Appendix-B annex-1.1.3 for the passive node

This appendix describes alternative methods to perform Test 1.1.3 on the passive node which doesn't have the application to send ICMPv6 Echo Request.
In these method, only TAR-EN2 role can be the passive node. If TAR-EN1 is the passive node, switch the role such that TAR-EN2 is the passive node.

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

Requirements:

- TAR-EN1
  - Must support the application to send ICMPv6 Echo Request
  - Must support the application to send UDP packet (e.g., DNS lookup client)
- TAR-EN2 (passive node)
  - 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)
 TAR-EN1 ------- transport ------- TAR-EN2

HOST1_SA1-O ----- spi=0x1000 -----> HOST2_SA1-I   ICMPv6 Echo Request
HOST1_SA2-I <---- spi=0x2000 ------ HOST2_SA2-O   ICMPv6 Echo Reply
HOST1_SA3-I <---- spi=0x3000 ------ HOST2_SA3-O   ICMPv6 Destination Unreachable
                                                      (Port unreachable)
```

HOST1_SA1-O and HOST2_SA1-I

Security Association Database (SAD)

source address	TAR-EN1_Network0
destination address	TAR-EN2_Network1
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des1to2req
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha11to2req

Security Policy Database (SPD)

	HOST1_SA1-O	HOST2_SA1-I
source address	TAR-EN1_Network0	
destination address	TAR-EN2_Network1	
upper spec	ICMPv6 Echo Request	
direction	outbound	inbound
protocol	ESP	
mode	transport	

HOST1_SA2-I and HOST2_SA2-O

Security Association Database (SAD)

source address	TAR-EN2_Network1
destination address	TAR-EN1_Network0
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des2to1rep
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha12to1rep

Security Policy Database (SPD)

	HOST1_SA2-I	HOST2_SA2-O
source address	TAR-EN2_Network1	
destination address	TAR-EN1_Network0	
upper spec	ICMPv6 Echo Reply	
direction	inbound	outbound
protocol	ESP	
mode	transport	

HOST1_SA3-I and HOST2_SA3-O

Security Association Database (SAD)

source address	TAR-EN2_Network1
destination address	TAR-EN1_Network0
SPI	0x3000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des2to1dst
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha12to1dst

Security Policy Database (SPD)

	HOST1_SA3-I	HOST2_SA3-O
source address	TAR-EN2_Network1	
destination address	TAR-EN1_Network0	
upper spec	ICMPv6 Destination Unreachable	
direction	inbound	outbound
protocol	ESP	
mode	transport	

Packets:

ICMPv6 Echo Request with ESP1

IPv6	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des1to2req
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha11to2req
ICMPv6	Type	128 (Echo Request)

ICMPv6 Echo Reply with ESP2

IPv6	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des2to1rep
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha12to1rep
ICMPv6	Type	129 (Echo Reply)

UDP packet toward closed port

IPv6	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
UDP	Source Port	Any unused port on TAR-EN1
	Destination Port	Any closed port on TAR-EN2

ICMPv6 Destination Unreachable with ESP3

IPv6	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	0x3000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des2to1dst
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha12to1dst
ICMPv6	Type	1 (Destination Unreachable)
	Code	4 (Port unreachable)

Procedure:

```
                          (passive node)
TAR-EN1                    TAR-EN2
    |                          |
    |------ 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
```

1. TAR-EN1 sends "ICMPv6 Echo Request with ESP1" to TAR-EN2
2. Observe the packet transmitted by TAR-EN2
3. Save the command log on TAR-EN1
4. TAR-EN1 sends "UDP packet toward closed port" to TAR-EN2
5. Observe the packet transmitted by TAR-EN2
6. Save the command log on TAR-EN1

Observable Result:

Observable Result #1
Step-2: TAR-EN2 transmits "ICMPv6 Echo Reply with ESP2"
Observable Result #2
Step-5: TAR-EN2 transmits "ICMPv6 Destination Unreachable with ESP3"

Possible Problems:

None.

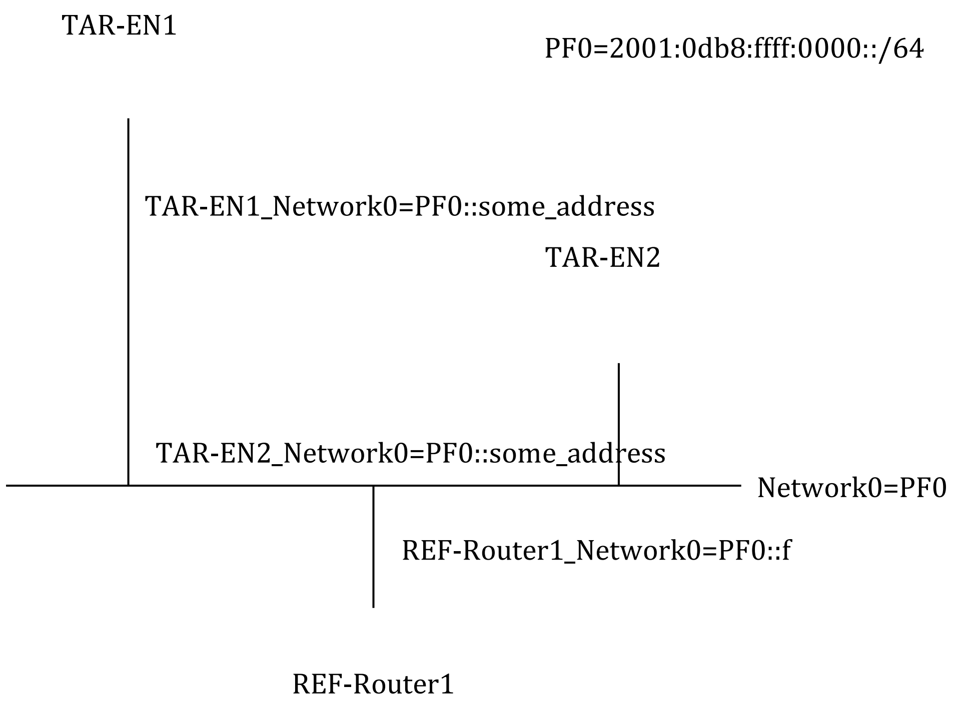
Invoking Neighbor Unreachability Detection

Requirements:

- TAR-EN1
  - Must support the application to send ICMPv6 Echo Request
- TAR-EN2 (passive node)
  - Must respond to ICMPv6 Echo Request with ICMPv6 Echo Reply

Initialization:

Use following topology



Reboot TAR-EN1 and TAR-EN2 making sure it has cleared its neighbor cache. Allow time for all devices on Network0 to perform Stateless Address Autoconfiguration and Duplicate Address Detection.

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

```
                                    (passive node)
  TAR-EN1 ------- transport ------- TAR-EN2

HOST1_SA1-O ----- spi=0x1000 -----> HOST2_SA1-I   ICMPv6 Echo Request
HOST1_SA2-I <---- spi=0x2000 ------ HOST2_SA2-O   ICMPv6 Echo Reply
HOST1_SA3-I <---- spi=0x3000 ------ HOST2_SA3-O   ICMPv6 Neighbor Solicitation
HOST1_SA4-O ----- spi=0x4000 -----> HOST2_SA4-I   ICMPv6 Neighbor Advertisement
```

HOST1_SA1-O and HOST2_SA1-I

Security Association Database (SAD)

source address	TAR-EN1_Network0
destination address	TAR-EN2_Network0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des1to2req
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha11to2req

Security Policy Database (SPD)

	HOST1_SA1-O	HOST2_SA1-I
source address	TAR-EN1_Network0	
destination address	TAR-EN2_Network0	
upper spec	ICMPv6 Echo Request	
direction	outbound	inbound
protocol	ESP	
mode	transport	

HOST1_SA2-I and HOST2_SA2-O

Security Association Database (SAD)

source address	TAR-EN2_Network0
destination address	TAR-EN1_Network0
SPI	0x2000
mode	Transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des2to1rep
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha12to1rep

Security Policy Database (SPD)

	HOST1_SA2-I	HOST2_SA2-O
source address	TAR-EN2_Network0	
destination address	TAR-EN1_Network0	
upper spec	ICMPv6 Echo Reply	
direction	inbound	outbound
protocol	ESP	
mode	transport	

HOST1_SA3-I and HOST2_SA3-O

Security Association Database (SAD)

source address	TAR-EN2_Network0
destination address	TAR-EN1_Network0
SPI	0x3000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des2to1sol
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha12to1sol

Security Policy Database (SPD)

	HOST1_SA3-I	HOST2_SA3-O
source address	TAR-EN2_Network0	
destination address	TAR-EN1_Network0	
upper spec	ICMPv6 Neighbor Solicitation	
direction	inbound	outbound
protocol	ESP	
mode	transport	

HOST1_SA4-O and HOST2_SA4-I

Security Association Database (SAD)

source address	TAR-EN1_Network0
destination address	TAR-EN2_Network0
SPI	0x4000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des1to2adv
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha11to2adv

Security Policy Database (SPD)

	HOST1_SA1-O	HOST2_SA1-I
source address	TAR-EN1_Network0	
destination address	TAR-EN2_Network0	
upper spec	ICMPv6 Neighbor Advertisement	
direction	outbound	inbound
protocol	ESP	
mode	transport	

Packets:

ICMPv6 Neighbor Solicitation (multicast)

IPv6	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network0 (solicited-node multicast address)
ICMPv6	Type	135 (Neighbor Solicitation)
	Target Address	TAR-EN2_Network0
	Source link-layer address Option	

ICMPv6 Neighbor Advertisement

IPv6	Source Address	TAR-EN2_Network0
	Destination Address	TAR-EN1_Network0
ICMPv6	Type	136 (Neighbor Advertisement)
	S	true
	O	true
	Target Address	TAR-EN2_Network0
	Target link-layer address Option	

ICMPv6 Echo Request with ESP1

IPv6	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des1to2req
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha11to2req
ICMPv6	Type	128 (Echo Request)

ICMPv6 Echo Reply with ESP2

IPv6	Source Address	TAR-EN2_Network0
	Destination Address	TAR-EN1_Network0
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des2to1rep
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha12to1rep
ICMPv6	Type	129 (Echo Reply)

ICMPv6 Neighbor Solicitation with ESP3

IPv6	Source Address	TAR-EN2_Network0
	Destination Address	TAR-EN1_Network0
ESP	SPI	0x3000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des2to1sol
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha12to1sol
ICMPv6	Type	135 (Neighbor Solicitation)
	Target Address	TAR-EN1_Network0
	Source link-layer address Option	

ICMPv6 Neighbor Advertisement with ESP4

IPv6	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network0
ESP	SPI	0x4000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des1to2adv
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha11to2adv
ICMPv6	Type	136 (Neighbor Advertisement)
	S	true
	O	true
	Target Address	TAR-EN1_Network0
	Target link-layer address Option	

Procedure:

```
                              (passive node)
  TAR-EN1                         TAR-EN2
       |                             |
    [NONE]                        [NONE]
       |                             |
[INCOMPLETE]                         |
       |                             |
       |------ plaintext ----->| ICMPv6 Neighbor Solicitation (multicast)
       |                             |
       |                          [STALE]
       |                             |
       |<----- plaintext ------| ICMPv6 Neighbor Advertisement
       |                             |
[REACHABLE]                       [DELAY]
       |                             |
       |------ ciphertext ---->| ICMPv6 Echo Request with ESP1
       |<----- ciphertext -----| ICMPv6 Echo Reply with ESP2
       |                             |       (Observable Result #1)
       |                             |
       |                             * wait DELAY_FIRST_PROBE_TIME (5) seconds
       |                             |
       |                          [PROBE]
       |                             |
       |<----- ciphertext -----| ICMPv6 Neighbor Solicitation with ESP3
       |                             |       (Observable Result #2)
       |------ ciphertext ---->| ICMPv6 Neighbor Advertisement with ESP4
       |                             |
       |                        [REACHABLE]
       |                             |
       V                             V
```

1. TAR-EN1 sends "ICMPv6 Echo Request with ESP1" to TAR-EN2

* Address Resolution ("ICMPv6 Neighbor Solicitation (multicast)" and "ICMPv6 Neighbor Advertisement") is invoked

2. Observe the packet transmitted by TAR-EN2
3. Save the command log on TAR-EN1
4. Observe the packet transmitted by TAR-EN2 for DELAY_FIRST_PROBE_TIME (5) seconds
5. Save the command log on TAR-EN1

Observable Result:

Observable Result #1
Step-2: TAR-EN2 transmits " ICMPv6 Echo Reply with ESP2 "
Observable Result #2
Step-4: TAR-EN2 transmits "ICMPv6 Neighbor Solicitation with ESP3"
TAR-EN1 responds to "ICMPv6 Neighbor Solicitation with ESP3" with "ICMPv6 Neighbor Advertisement with ESP4"

Possible Problems:

None.

Appendix-C annex-4.1.3 for the passive node

This appendix describes alternative methods to perform Test 4.1.3 on the passive node which doesn't have the application to send ICMPv6 Echo Request.
In these method, only TAR-EN2 role can be the passive node. If TAR-EN1 is the passive node, switch the role such that TAR-EN2 is the passive node.

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

Requirements:

- TAR-EN1
  - Must support the application to send ICMPv6 Echo Request
  - Must support the application to send UDP packet (e.g., DNS lookup client)
- TAR-EN2 (passive node)
  - 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)
  TAR-EN1 ======== tunnel ========= TAR-EN2

HOST1_SA1-O ----- spi=0x1000 -----> HOST2_SA1-I   ICMPv6 Echo Request
HOST1_SA2-I <---- spi=0x2000 ------ HOST2_SA2-O   ICMPv6 Echo Reply
HOST1_SA3-I <---- spi=0x3000 ------ HOST2_SA3-O   ICMPv6 Destination Unreachable
                                                      (Port unreachable)
```

HOST1_SA1-O and HOST2_SA1-I

Security Association Database (SAD)

source address	TAR-EN1_Network0
destination address	TAR-EN2_Network1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des1to2req
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha11to2req

Security Policy Database (SPD)

	HOST1_SA1-O	HOST2_SA1-I
tunnel source address	TAR-EN1_Network0	
tunnel destination address	TAR-EN2_Network1	
source address	TAR-EN1_Network0	
destination address	TAR-EN2_Network1	
upper spec	ICMPv6 Echo Request	
direction	outbound	inbound
protocol	ESP	
mode	transport	

HOST1_SA2-I and HOST2_SA2-O

Security Association Database (SAD)

source address	TAR-EN2_Network1
destination address	TAR-EN1_Network0
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des2to1rep
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha12to1rep

Security Policy Database (SPD)

	HOST1_SA2-I	HOST2_SA2-O
tunnel source address	TAR-EN2_Network1	
tunnel destination address	TAR-EN1_Network0	
source address	TAR-EN2_Network1	
destination address	TAR-EN1_Network0	
upper spec	ICMPv6 Echo Reply	
direction	inbound	outbound
protocol	ESP	
mode	transport	

HOST1_SA3-I and HOST2_SA3-O

Security Association Database (SAD)

source address	TAR-EN2_Network1
destination address	TAR-EN1_Network0
SPI	0x3000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des2to1dst
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha12to1dst

Security Policy Database (SPD)

	HOST1_SA3-I	HOST2_SA3-O
tunnel source address	TAR-EN2_Network1	
tunnel destination address	TAR-EN1_Network0	
source address	TAR-EN2_Network1	
destination address	TAR-EN1_Network0	
upper spec	ICMPv6 Destination Unreachable	
direction	inbound	outbound
protocol	ESP	
mode	transport	

Packets:

ICMPv6 Echo Request with ESP1

IPv6	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des1to2req
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha11to2req
IPv6	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
ICMPv6	Type	128 (Echo Request)

ICMPv6 Echo Reply with ESP2

IPv6	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des2to1rep
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha12to1rep
IPv6	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ICMPv6	Type	129 (Echo Reply)

UDP packet toward closed port

IPv6	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network1
UDP	Source Port	Any unused port on TAR-EN1
	Destination Port	Any closed port on TAR-EN2

ICMPv6 Destination Unreachable with ESP3

IPv6	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ESP	SPI	0x3000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des2to1dst
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha12to1dst
IPv6	Source Address	TAR-EN2_Network1
	Destination Address	TAR-EN1_Network0
ICMPv6	Type	1 (Destination Unreachable)
	Code	4 (Port unreachable)

Procedure:

```
                              (passive node)
TAR-EN1                        TAR-EN2
     |                              |
     |====== 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
```

1. TAR-EN1 sends "ICMPv6 Echo Request with ESP1" to TAR-EN2
2. Observe the packet transmitted by TAR-EN2
3. Save the command log on TAR-EN1
4. TAR-EN1 sends "UDP packet toward closed port" to TAR-EN2
5. Observe the packet transmitted by TAR-EN2
6. Save the command log on TAR-EN1

Observable Result:

Observable Result #1
Step-2: TAR-EN2 transmits "ICMPv6 Echo Reply with ESP2"
Observable Result #2
Step-5: TAR-EN2 transmits "ICMPv6 Destination Unreachable with ESP3"

Possible Problems:

None.

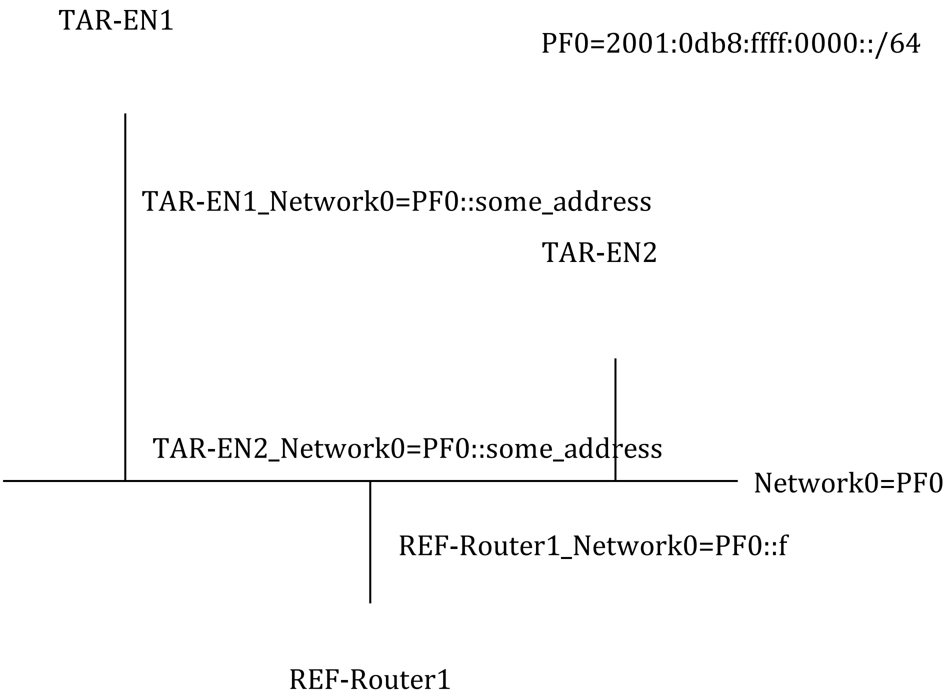
Invoking Neighbor Unreachability Detection

Requirements:

- TAR-EN1
  - Must support the application to send ICMPv6 Echo Request
- TAR-EN2 (passive node)
  - Must respond to ICMPv6 Echo Request with ICMPv6 Echo Reply

Initialization:

Use following topology



Reboot TAR-EN1 and TAR-EN2 making sure it has cleared its neighbor cache. Allow time for all devices on Network0 to perform Stateless Address Autoconfiguration and Duplicate Address Detection.

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

```
                                        (passive node)
  TAR-EN1 ======== tunnel ========= TAR-EN2

HOST1_SA1-O ----- spi=0x1000 -----> HOST2_SA1-I    ICMPv6 Echo Request
HOST1_SA2-I <---- spi=0x2000 ------ HOST2_SA2-O    ICMPv6 Echo Reply
HOST1_SA3-I <---- spi=0x3000 ------ HOST2_SA3-O    ICMPv6 Neighbor Solicitation
HOST1_SA4-O ----- spi=0x4000 -----> HOST2_SA4-I    ICMPv6 Neighbor Advertisement
```

HOST1_SA1-O and HOST2_SA1-I

Security Association Database (SAD)

source address	TAR-EN1_Network0
destination address	TAR-EN2_Network0
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des1to2req
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha11to2req

Security Policy Database (SPD)

	HOST1_SA1-O	HOST2_SA1-I
tunnel source address	TAR-EN1_Network0	
tunnel destination address	TAR-EN2_Network0	
source address	TAR-EN1_Network0	
destination address	TAR-EN2_Network0	
upper spec	ICMPv6 Echo Request	
direction	outbound	inbound
protocol	ESP	
mode	transport	

HOST1_SA2-I and HOST2_SA2-O

Security Association Database (SAD)

source address	TAR-EN2_Network0
destination address	TAR-EN1_Network0
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des2to1rep
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha12to1rep

Security Policy Database (SPD)

	HOST1_SA2-I	HOST2_SA2-O
tunnel source address	TAR-EN2_Network0	
tunnel destination address	TAR-EN1_Network0	
source address	TAR-EN2_Network0	
destination address	TAR-EN1_Network0	
upper spec	ICMPv6 Echo Reply	
direction	inbound	outbound
protocol	ESP	
mode	transport	

HOST1_SA3-I and HOST2_SA3-O

Security Association Database (SAD)

source address	TAR-EN2_Network0
destination address	TAR-EN1_Network0
SPI	0x3000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des2to1sol
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha12to1sol

Security Policy Database (SPD)

	HOST1_SA3-I	HOST2_SA3-O
tunnel source address	TAR-EN2_Network0	
tunnel destination address	TAR-EN1_Network0	
source address	TAR-EN2_Network0	
destination address	TAR-EN1_Network0	
upper spec	ICMPv6 Neighbor Solicitation	
direction	inbound	outbound
protocol	ESP	
mode	transport	

HOST1_SA4-O and HOST2_SA4-I

Security Association Database (SAD)

source address	TAR-EN1_Network0
destination address	TAR-EN2_Network0
SPI	0x4000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3des1to2adv
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readysha11to2adv

Security Policy Database (SPD)

	HOST1_SA1-O	HOST2_SA1-I
tunnel source address	TAR-EN1_Network0	
tunnel destination address	TAR-EN2_Network0	
source address	TAR-EN1_Network0	
destination address	TAR-EN2_Network0	
upper spec	ICMPv6 Neighbor Advertisement	
direction	outbound	inbound
protocol	ESP	
mode	transport	

Packets:

ICMPv6 Neighbor Solicitation (multicast)

IPv6	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network0 (solicited-node multicast address)
ICMPv6	Type	135 (Neighbor Solicitation)
	Target Address	TAR-EN2_Network0
	Source link-layer address Option	

ICMPv6 Neighbor Advertisement

IPv6	Source Address	TAR-EN2_Network0
	Destination Address	TAR-EN1_Network0
ICMPv6	Type	136 (Neighbor Advertisement)
	S	true
	O	true
	Target Address	TAR-EN2_Network0
	Target link-layer address Option	

ICMPv6 Echo Request with ESP1

IPv6	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des1to2req
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha11to2req
IPv6	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network0
ICMPv6	Type	128 (Echo Request)

ICMPv6 Echo Reply with ESP2

IPv6	Source Address	TAR-EN2_Network0
	Destination Address	TAR-EN1_Network0
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des2to1rep
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha12to1rep
IPv6	Source Address	TAR-EN2_Network0
	Destination Address	TAR-EN1_Network0
ICMPv6	Type	129 (Echo Reply)

ICMPv6 Neighbor Solicitation with ESP3

IPv6	Source Address	TAR-EN2_Network0
	Destination Address	TAR-EN1_Network0
ESP	SPI	0x3000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des2to1sol
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha12to1sol
IPv6	Source Address	TAR-EN2_Network0
	Destination Address	TAR-EN1_Network0
ICMPv6	Type	135 (Neighbor Solicitation)
	Target Address	TAR-EN1_Network0
	Source link-layer address Option	

ICMPv6 Neighbor Advertisement with ESP4

IPv6	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network0
ESP	SPI	0x4000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3des1to2adv
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readysha11to2adv
IPv6	Source Address	TAR-EN1_Network0
	Destination Address	TAR-EN2_Network0
ICMPv6	Type	136 (Neighbor Advertisement)
	S	true
	O	true
	Target Address	TAR-EN1_Network0
	Target link-layer address Option	

Procedure:

```
                              (passive node)
 TAR-EN1                        TAR-EN2
      |                            |
    [NONE]                       [NONE]
      |                            |
[INCOMPLETE]                       |
      |                            |
      |------ plaintext ----->| ICMPv6 Neighbor Solicitation (multicast)
      |                            |
      |                         [STALE]
      |                            |
      |<----- plaintext ------| ICMPv6 Neighbor Advertisement
      |                            |
[REACHABLE]                     [DELAY]
      |                            |
      |====== ciphertext ====>| ICMPv6 Echo Request with ESP1
      |<===== ciphertext =====| ICMPv6 Echo Reply with ESP2
      |                            |      (Observable Result #1)
      |                            |
      |                            * wait DELAY_FIRST_PROBE_TIME (5) seconds
      |                            |
      |                         [PROBE]
      |                            |
      |<===== ciphertext =====| ICMPv6 Neighbor Solicitation with ESP3
      |                            |      (Observable Result #2)
      |====== ciphertext ====>| ICMPv6 Neighbor Advertisement with ESP4
      |                            |
      |                       [REACHABLE]
      |                            |
      V                            V
```

1. TAR-EN1 sends "ICMPv6 Echo Request with ESP1" to TAR-EN2

* Address Resolution ("ICMPv6 Neighbor Solicitation (multicast)" and "ICMPv6 Neighbor Advertisement") is invoked

2. Observe the packet transmitted by TAR-EN2
3. Save the command log on TAR-EN1
4. Observe the packet transmitted by TAR-EN2 for DELAY_FIRST_PROBE_TIME (5) seconds
5. Save the command log on TAR-EN1

Observable Result:

Observable Result #1
Step-2: TAR-EN2 transmits " ICMPv6 Echo Reply with ESP2 "
Observable Result #2
Step-4: TAR-EN2 transmits "ICMPv6 Neighbor Solicitation with ESP3"
TAR-EN1 responds to "ICMPv6 Neighbor Solicitation with ESP3" with "ICMPv6 Neighbor Advertisement with ESP4"

Possible Problems:

None.

Appendix D: 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.