IPv6 Ready Logo

Phase-2 Conformance
Test Specification
IPsec and IKEv2

Technical Document
Revision 2.0.1

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

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.

Definitions

EN	End-Node
SGW	Security Gateway
NUT	Node Under Test
AH	Authentication Header
ESP	Encapsulating Security Protocol
IKE	Internet Key Exchange
SA	Security Association
CA	Certificate Authority
PSK	Pre-Shared Key
MTU	Maximum Transmission Unit
RA	Router Advertisement

Requirements

To obtain the IPv6 Ready Logo Phase-2 for IPsec (IPsec Logo), the Node Under Test (NUT) must satisfy following requirements.

Equipment Type

- End-Node (EN)

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

- Security Gateway (SGW)

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

Security Protocol

NUTs must utilize ESP regardless of the type of the NUT. The IPv6 Ready Logo Program does not test AH.

Mode

The mode requirement depends on the type of NUT.

- End-Node:

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

- SGW:

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

Keying

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

According to RFC 7321bis, Section 3:

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

If manual keying is used anyway, ENCR_AES_CBC MUST be used, and
ENCR_AES_CCM, ENCR_AES_GCM and ENCR_CHACHA20_POLY1305 MUST NOT be
used as these algorithms require IKE.
```

Following this recommendation, a configuration using Dynamic Keying, facilitated by IKE is used by default, and specifically IKEv2. IKEv1 is obsolete and not supported.

When IKEv2 is used, the encryption keys and Integrity keys are negotiated dynamically. Manual Keys MUST NOT be used.

Test Traffic

Most tests use ICMP/UDP/TCP for data traffic.

Category

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

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

Required Tests

The table below indicates the IPv6 Ready Logo requirement level for each test case.

Each test case is made up of individual test parts which use a specific topology. Test parts which are applicable to the DUT device type must be run.

For example, End-node devices must run all End-node vs. End-node test parts, as well as all End-node vs. SGW test parts. SGW devices must run all End-node vs. SGW test parts, as well as all SGW vs. SGW test parts.

Required Test Cases Section 1 (Applies to EN and SGW)
IPsec.Conf.1.1.1.1: IKE_SA_INIT Request Format
IPsec.Conf.1.1.1.2: IKE_SA_INIT Retransmission
IPsec.Conf.1.1.1.3: IKE_SA_INIT Cryptographic Algorithm Negotiation (A) AES128/SHA256/DH14 (B) AES256
IPsec.Conf.1.1.1.4: IKE_SA_INIT Exchange with N(COOKIE)
IPsec.Conf.1.1.1.7: IKE_SA_INIT inconsistent response proposal
IPsec.Conf.1.1.1.8: IKE_SA_INIT Forward Compatibility (A)(B)
IPsec.Conf.1.1.2.1: IKE_AUTH Request Format
IPsec.Conf.1.1.2.2: IKE_AUTH Exchange Succeeds
IPsec.Conf.1.1.2.3: IKE_AUTH Retransmission (A)(B)(C)
IPsec.Conf.1.1.2.4: State Synchronization (A)(B)
IPsec.Conf.1.1.2.5: IKE_AUTH Cryptographic Algorithm Negotiation (A) AES128/SHA256 (B) AES256/SHA256 (D) AESGCM (F) NULL
IPsec.Conf.1.1.2.6: IKE_AUTH N(NO_PROPOSAL_CHOSEN)
IPsec.Conf.1.1.2.7: IKE_AUTH Inconsistent response proposal
IPsec.Conf.1.1.2.8: Traffic Selector Negotiation
IPsec.Conf.1.1.2.9: Peer Identification (A)(B)(C)
IPsec.Conf.1.1.2.10: Authentication via RSA Digital Signature
IPsec.Conf.1.1.2.11: Authentication via PSK (A)(B)(C)
IPsec.Conf.1.1.2.12: IKE_AUTH Forward Compatibility
IPsec.Conf.1.1.2.13: IKE_AUTH Unrecognized Error (A)(B)
IPsec.Conf.1.1.3.1: IKE_AUTH Request Format in Tunnel Mode
IPsec.Conf.1.1.3.2: IKE_AUTH Exchange Succeeds in Tunnel Mode
IPsec.Conf.1.1.5.1: IKE_SA Deletion
IPsec.Conf.1.1.5.2: CHILD_SA Deletion
IPsec.Conf.1.2.1.1: IKE_SA_INIT Response Format
IPsec.Conf.1.2.1.2: IKE_SA_INIT Retransmission
IPsec.Conf.1.2.1.3: IKE_SA_INIT Cryptographic Algorithm Negotiation (A) AES128/SHA256/DH14 (B) AES256

IPsec.Conf.1.2.1.4: IKE_SA_INIT Version Number (A)(B)
IPsec.Conf.1.2.1.5: IKE_SA_INIT Multiple Transforms (A)(B)(C)
IPsec.Conf.1.2.1.6: IKE_SA_INIT Multiple Proposals
IPsec.Conf.1.2.1.7: IKE_SA_INIT Exchange with INVALID_KE_PAYLOAD
IPsec.Conf.1.2.1.8: IKE_SA_INIT Forward Compatibility (A)(B)
IPsec.Conf.1.2.1.9: IKE_SA_INIT Invalid
IPsec.Conf.1.2.2.1: IKE_AUTH Response Format
IPsec.Conf.1.2.2.2: IKE_AUTH Exchange Succeeds
IPsec.Conf.1.2.2.3: IKE_AUTH Retransmission
IPsec.Conf.1.2.2.4: State Synchronization (A)(B)
IPsec.Conf.1.2.2.5: IKE_AUTH Cryptographic Algorithm Negotiation (A) AES128/SHA256 (B) AES256/SHA256 (D) AESGCM (F) NULL
IPsec.Conf.1.2.2.6: IKE_AUTH Multiple Transforms (A)(B)(C)
IPsec.Conf.1.2.2.7: IKE_AUTH Multiple Proposals
IPsec.Conf.1.2.2.8: IKE_AUTH N(NO_PROPOSAL_CHOSEN)
IPsec.Conf.1.2.2.9: Traffic Selector Negotiation (A)(B)(C)
IPsec.Conf.1.2.2.10: Peer Identification (A)(B)(C)
IPsec.Conf.1.2.2.11: Authentication via RSA Digital Signature
IPsec.Conf.1.2.2.12: Authentication via PSK (A)(B)(C)
IPsec.Conf.1.2.2.13: IKE_AUTH Forward Compatibility
IPsec.Conf.1.2.2.14: Unrecognized Notify Type (A)(B)
IPsec.Conf.1.2.3.1: IKE_AUTH Response Format in Tunnel Mode
IPsec.Conf.1.2.3.2: IKE_AUTH Exchange Succeeds in Tunnel Mode
IPsec.Conf.1.2.5.1: INFORMATIONAL Exchange (A)(B)(C)
IPsec.Conf.1.2.5.2: IKE_SA Deletion
IPsec.Conf.1.2.5.3: CHILD_SA Deletion

Required Test Cases Section 2 (Applies to EN)
IPsec.Conf.2.1.1. Select SPD
IPsec.Conf.2.1.2. Select SPD (Next Layer Protocol Selectors) (A)(B)
IPsec.Conf.2.1.3. Sequence Number Increment
IPsec.Conf.2.1.4. Packet Too Big Reception
IPsec.Conf.2.1.5. Receipt of No Next Header (A)
IPsec.Conf.2.1.6. Bypass Policy
IPsec.Conf.2.1.7. Discard Policy
IPsec.Conf.2.1.8. Transport Mode Padding (A)(B)
IPsec.Conf.2.1.9. Invalid SPI
IPsec.Conf.2.1.10. Invalid ICV
IPsec.Conf.2.2.1. Tunnel Mode with SGW
IPsec.Conf.2.2.2. Tunnel Mode Select SPD
IPsec.Conf.2.2.3. Tunnel Mode Sequence Number Increment
IPsec.Conf.2.2.4. Tunnel Mode Packet Too Big Reception
IPsec.Conf.2.2.5. Tunnel Mode Receipt of No Next Header (A)

Required Test Cases Section 3 (Applies to SGW)
IPsec.Conf.3.1.1. Select SPD (2 SGW Peers)
IPsec.Conf.3.1.2. Select SPD (2 Hosts behind same Peer)
IPsec.Conf.3.1.3. Sequence Number Increment
IPsec.Conf.3.1.4. Packet Too Big Transmission
IPsec.Conf.3.1.5. Packet Too Big Forwarding
IPsec.Conf.3.1.6. Receipt of No Next Header (A)
IPsec.Conf.3.1.7. Bypass Policy
IPsec.Conf.3.1.8. Discard Policy
IPsec.Conf.3.1.9. Tunnel Mode Padding (A) (B)
IPsec.Conf.3.1.10. Invalid SPI
IPsec.Conf.3.1.11. Invalid ICV
IPsec.Conf.3.1.12. Tunnel Mode with End-Node

Required Test Cases Section 4
4.1.1 Applies to EN – See Table in Section 4.1
4.1.2 Applies to EN – See Table in Section 4.1
4.1.3 Applies to SGW – See Table in Section 4.1

Possible Problem Summary

The following test cases have documented possible problems that allow for altered or omitted steps in their procedures. Please see each specific test case listed for more information:

- IPsec.Conf.1.1.1.1: IKE_SA_INIT Request Format
- IPsec.Conf.1.1.1.2: IKE_SA_INIT Retransmission
- IPsec.Conf.1.1.1.5: IKE_SA_INIT Exchange with N(INVALID_KE_PAYLOAD)
- IPsec.Conf.1.1.1.6: IKE_SA_INIT Exchange; COOKIE and INVALID KE
- IPsec.Conf.1.1.2.1: IKE_AUTH Request Format
- IPsec.Conf.1.1.2.3: IKE_AUTH Retransmission
- IPsec.Conf.1.1.2.8: Traffic Selector Negotiation
- IPsec.Conf.1.1.2.11: Authentication via PSK
- IPsec.Conf.1.1.3.1: IKE_AUTH Request Format in Tunnel Mode
- IPsec.Conf.1.1.5.1: IKE_SA Deletion
- IPsec.Conf.1.1.5.2: CHILD_SA Deletion
- IPsec.Conf.1.2.1.1: IKE_SA_INIT Response Format
- IPsec.Conf.1.2.1.4: IKE_SA_INIT Version Number
- IPsec.Conf.1.2.2.1: IKE_AUTH Response Format
- IPsec.Conf.1.2.2.9: Traffic Selector Negotiation
- IPsec.Conf.1.2.2.12: Authentication via PSK
- IPsec.Conf.1.2.3.1: IKE_AUTH Response Format in Tunnel Mode
- IPsec.Conf.1.2.5.2: IKE_SA Deletion
- IPsec.Conf.2.1.2. Select SPD (Next Layer Protocol Selectors)
- IPsec.Conf.2.1.6. Bypass Policy
- IPsec.Conf.2.1.7. Discard Policy
- IPsec.Conf.2.2.4. Tunnel Mode Packet Too Big Reception
- IPsec.Conf.2.2.6. Tunnel Mode Bypass Policy
- IPsec.Conf.2.2.7. Tunnel Mode Discard Policy
- IPsec.Conf.3.1.5. Packet Too Big Forwarding
- IPsec.Conf.3.1.7. Bypass Policy
- IPsec.Conf.3.1.8. Discard Policy

References

This test specification focuses on the following IPsec related RFCs.

Algorithms		
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)
RFC8221	ESP Alg Req	Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH). P. Wouters, D. Migault, J. Mattsson, Y. Nir, T. Kivinen. October 2017. (Format: TXT=33610 bytes) (Obsoletes RFC7321) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC8221)
RFC8247	IKEv2 Alg Reqs	Algorithm Implementation Requirements and Usage Guidance for the Internet Key Exchange Protocol Version 2 (IKEv2). Y. Nir, T. Kivinen, P. Wouters, D. Migault. September 2017. (Format: TXT=44739 bytes) (Obsoletes RFC4307) (Updates RFC7296) (Status: PROPOSED STANDARD) (DOI: 10.17487/RFC8247)
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

IKEv2

Section 1.1.1: IKEv2 IKE_SA_INIT Initiator
Section 1.2.1: IKEv2 IKE_SA_INIT Responder
Section 1.1.4: IKEv2 INFORMATIONAL Initiator
Section 1.2.5: IKEv2 INFORMATIONAL Responder

Tests in this section are applicable to IKEv2 End-nodes and Security Gateways equally, without modification. An appropriate topology may be used depending on the device type. For example, NUT (End-Node) may use Topology in Figure 1. A NUT (SGW) may use Topology in Figure 3.

Section 1.1.2: IKEv2 IKE_AUTH Initiator
Section 1.2.2: IKEv2 IKE_AUTH Responder
Section 1.2.3: IKEv2 IKE_AUTH Exchange Tunnel Mode Responder

Tests in this section are applicable to IKEv2 End-nodes and Security Gateway devices, with minor accommodations. Security Gateway devices operate only in Tunnel Mode, and therefore may omit the Notify(USE_TRANSPORT_MODE) payload. End-node devices may also omit this payload, with no loss of generality. An appropriate topology may be used depending on the device type. For example, NUT (End-Node) may use Topology in Figure 1. A NUT (SGW) may use Topology in Figure 3.

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

1. Set global address of NUT via SLAAC(NUT_Link0)
2. Set MTU of NUT via RA (MTU value is 1500 for Link0)
3. IPsec Transport Mode between NUT and EN1 and EN2

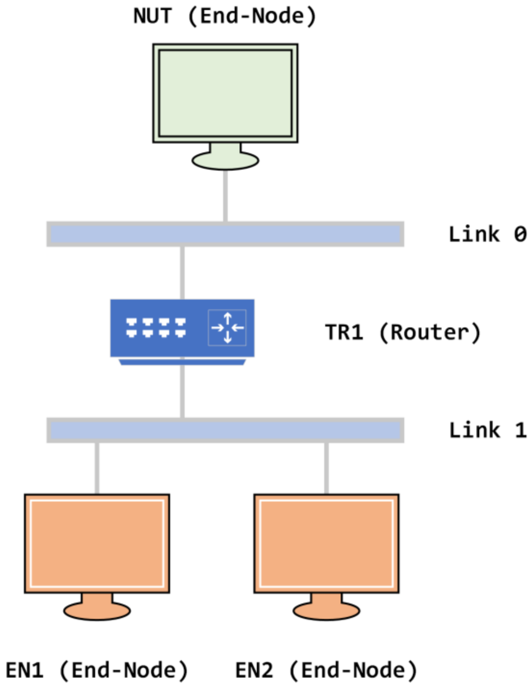


Figure 1 Topology for End-Node: Transport and Tunnel mode with End-Node

End-Node vs. SGW Tunnel Mode

1. Set global address to NUT by RA
2. Set MTU to NUT by RA (MTU value is 1500 for Link0)
3. IPsec Tunnel Mode between NUT and EN1.

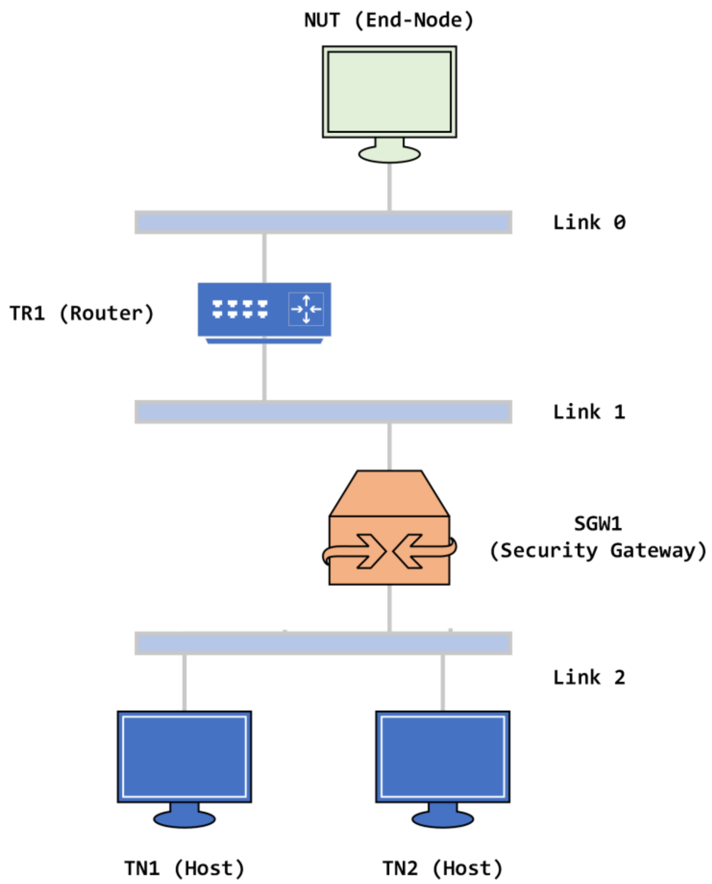


Figure 2 Topology for End-Node: Tunnel mode with SGW

SGW: Tunnel Mode with End-Node

1. Set global address of NUT manually (NUT_Link0, NUT_Link1)
2. Set routing table of NUT manually (TR1_Link1 for Link2)
3. Set MTU of NUT manually for Link0 and Link1 (MTU value is 1500 for Link0 and Link1)
4. IPsec Tunnel Mode between NUT and EN1.

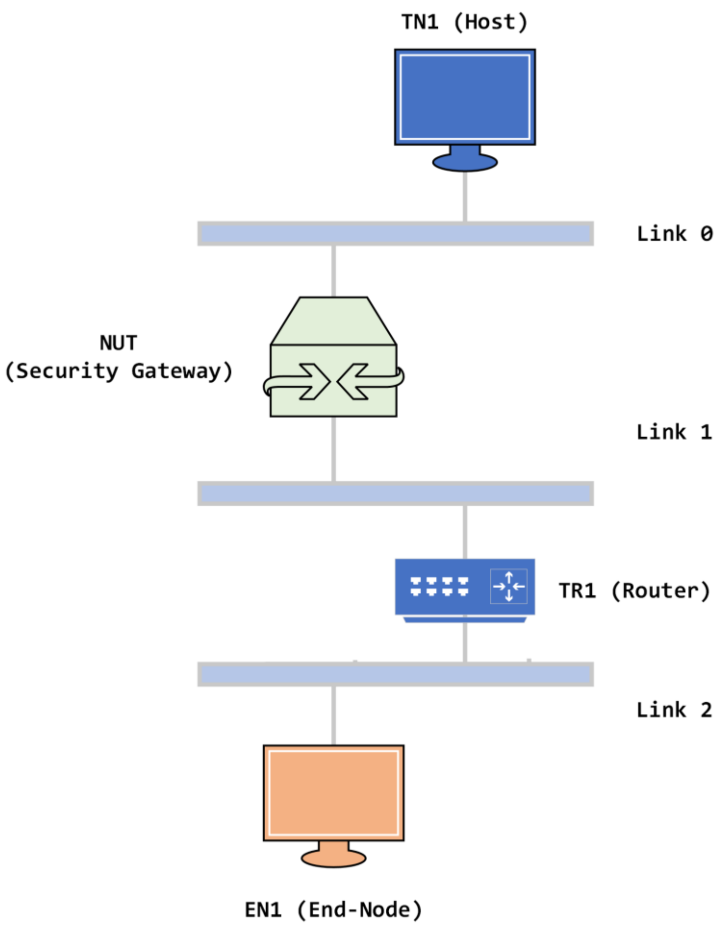


Figure 3 Topology for SGW: Tunnel mode with End-Node

SGW: Tunnel Mode

1. Set global address of NUT manually (NUT_Link0, NUT_Link1)
2. Set routing table of NUT manually (TR1_Link1 for Link2, Link3 and Link4)
3. Set MTU of NUT manually for Link0 and Link1 (MTU value is 1500 for Link0 and Link1)

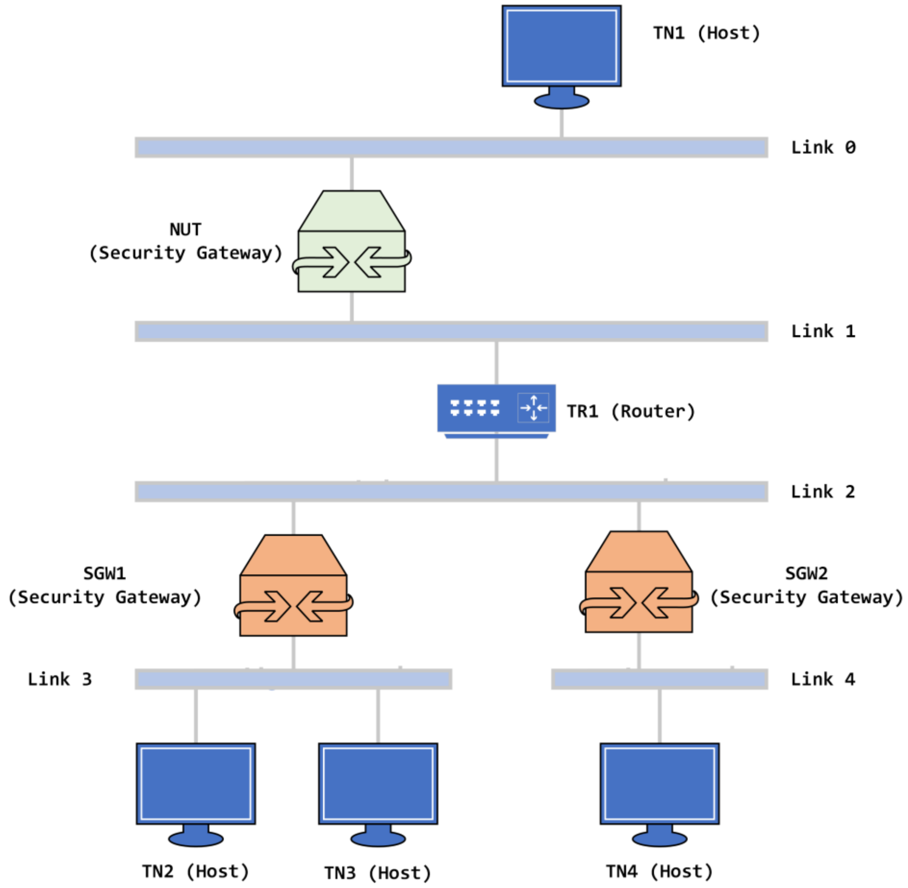


Figure 4 Topology for SGW: Tunnel mode with SGW

Description

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

Common Configurations

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

Common Configuration: Sections 1, 2 and 3

The Common Configurations described below should be utilized for test cases in Sections 1, 2, and 3, unless otherwise modified or specified by the test case. Both End-Node and SGW devices should utilize the configurations described below.

Global Security Associations

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

IKEv2 is mandatory to claim IPsec support.

IKEv2 Settings	
Authentication Method	PSK: IKETEST12345678!
ID Type (Local and Remote)	ID_IPV6_ADDR

IKE SA Configuration	
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)

CHILD SA (ESP) Configuration	
ESP Encryption Algorithm	ENCR_AES_CBC (128-bit)
ESP Integrity Algorithm	AUTH_HMAC_SHA2_256_128

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

Common Configuration: Section 4

Reference the list of algorithms specified in the Section 4.1: ESP Common Configurations.

Section 1: IKEv2

This Chapter describes the tests for IKEv2 Initiator

1.1. IKEv2 Initiator

1.1.1. IKE_SA_INIT Exchange

IPsec.Conf.1.1.1.1: IKE_SA_INIT Request Format

Purpose:

To verify a properly formatted IKE_SA_INIT Request

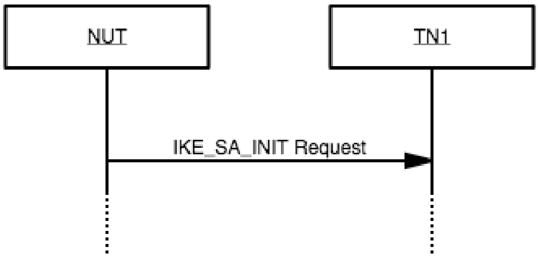
References:

- [RFC 7296] 1.2, 2.10, 3.1, 3.2, 3.3, 3.4, 3.9

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request. Verify fields according to **Table A** below.

Table A:

Payload			Field	Value
IKE Header			iSPI	Non-Zero
			rSPI	0
			Major Version	2
			Minor Version	0
			Exchange Type	IKE_SA_INIT (34)
			Flags	(00001000)2 = (08)16
			Message ID	0
SA Payload	Proposal		Last	0 or 2
			Proposal #	1
			Protocol ID	IKE (1)
			SPI Size	0
			# Transforms	4
		Transform	Last	0 or 3
			Transform Type	ENCR (1)
			Transform ID	(According to Common Configuration)
		Transform	Last	0 or 3
			Transform Type	PRF (2)
			Transform ID	(According to Common Configuration)
		Transform	Last	0 or 3
			Transform Type	INTEG (3)
			Transform ID	(According to Common Configuration)
		Transform	Last	0 or 3
			Transform Type	DH (4)
			Transform ID	(According to Common Configuration)
KE Payload			DH Group	(According to Common Configuration)
			Key Exchange Data	(According to DH Group)
Nonce Payload			Nonce Data	Unique value of length 16-256 octets

Possible Problems:

- The IKE_SA_INIT Request may have additional payloads not described above and can be ignored. The payloads may be in any order.
- There may be more than one Proposal in the SA Payload. One proposal must match the above.
- SA Payload Proposal Transforms may be in any order

IPsec.Conf.1.1.1.2: IKE_SA_INIT Retransmission

Purpose:

To verify correct retransmission of IKE_SA_INIT Requests

References:

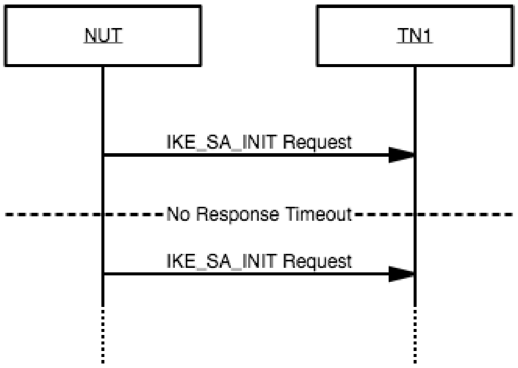
- [RFC 7296] 2.1, 2.2, 2.4

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

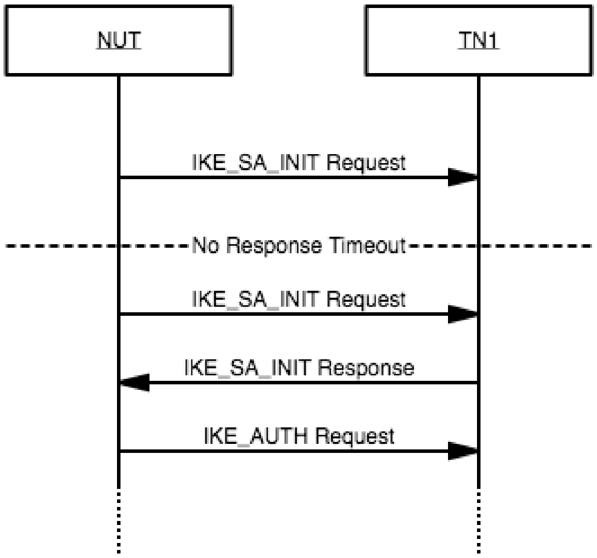
Procedure:

Part A: Retransmission



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	Wait for timeout.	The NUT retransmits a valid IKE_SA_INIT Request which is bitwise identical to the previously transmitted IKE_SA_INIT Request.

Part B: Retransmission Succeeds



Step	Action	Expected Result
3.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
4.	Wait for timeout.	The NUT retransmits a valid IKE_SA_INIT Request which is bitwise identical to the previously transmitted IKE_SA_INIT Request.
5.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.

Possible Problems:

- The length of the timeout for retransmission is unspecified, and usually not configurable by the user.

IPsec.Conf.1.1.1.3: IKE_SA_INIT Cryptographic Algorithm Negotiation

Purpose:

To verify algorithm negotiation during IKE_SA_INIT for IKE_SA.

References:

- [RFC 7296] 2.7, 3.3, 3.3.2, 3.3.5
- [RFC 8247]

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - Use the following Transforms corresponding to each part:

Common Configuration:

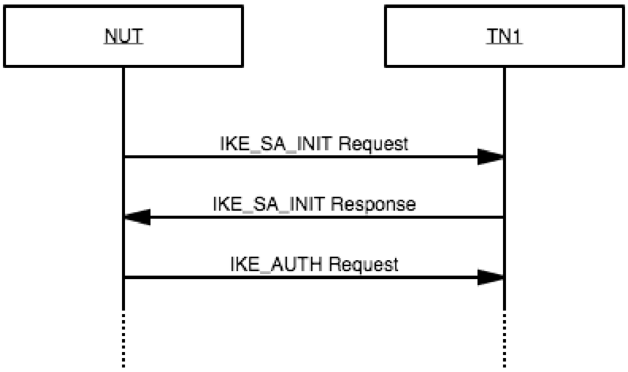
Type	Transform
ENCR (1)	ENCR_AES_CBC 128-bit (12)
PRF (2)	PRF_HMAC_SHA2_256 (5)
INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
DH (4)	2048-bit MODP Group (14)

The device should send the transforms specified in the Common Configuration. The table below substitutes the given Transform Type and Transform ID according to the part specified.

Part	Transform Type	Transform ID
A	ENCR (1)	ENCR_AES_CBC 128-bit (12)
	PRF (2)	PRF_HMAC_SHA2_256 (5)
	INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
	DH (4)	2048-bit MODP Group (14)
B - AES256	ENCR (1)	ENCR_AES_CBC 256-bit (12)
C -CHACHA	ENCR (1)	ENCR_CHACHA20_POLY1305 (28)
	INTEG (3)	Omitted or NONE (0)
D - AESGCM	ENCR (1)	ENCR_AES_GCM_16 128-bit (20)
	INTEG (3)	Omitted or NONE (0)
E - AESCCM	ENCR (1)	ENCR_AES_CCM_8 128-bit (14)
	INTEG (3)	Omitted or NONE (0)
F - SHA512	PRF (2)	PRF_HMAC_SHA2_512 (7)

	INTEG (3)	AUTH_HMAC_SHA2_512_256 (14)
G - AESXCBC	PRF (2)	PRF_AES128_XCBC (4)
	INTEG (3)	AUTH_AES_XCBC_96 (5)
H - DH19	DH (4)	256-bit random ECP group (19)

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request with transforms according to the table above.
2.	TN1 transmits a valid IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.

Possible Problems:

- None.

IPsec.Conf.1.1.1.4: IKE_SA_INIT Exchange with N(COOKIE)

Purpose:

To verify correct processing and transmission of COOKIE notifications.

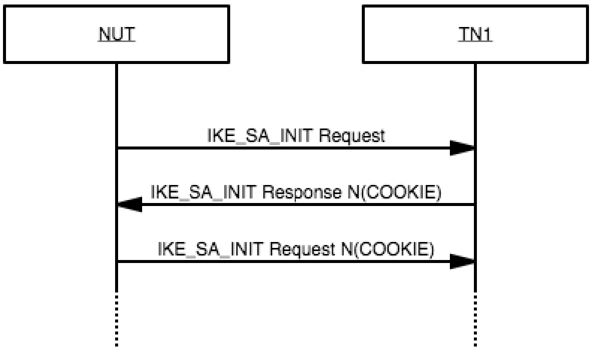
References:

- [RFC 7296] 2.6, 3.10.1

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits a valid IKE_SA_INIT Response containing only a Notifiy Payload of type COOKIE (16390) and valid Notification Data.	The NUT transmits a valid IKE_SA_INIT Request unchanged, except for the addition of a COOKIE Notification Payload as the first payload.

Possible Problems:

- None.

IPsec.Conf.1.1.1.5: IKE_SA_INIT Exchange with N(INVALID_KE_PAYLOAD)

Purpose:

To verify correct processing of N(INVALID_KE_PAYLOAD).

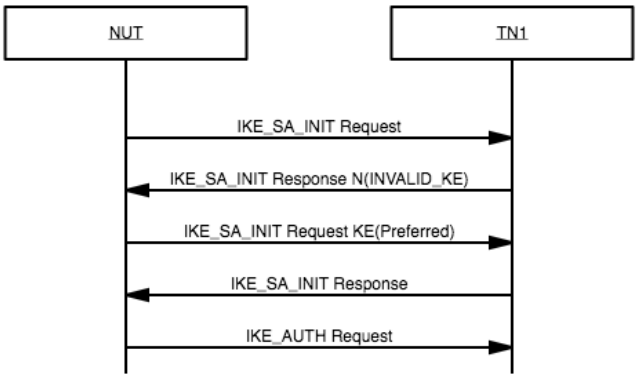
References:

- [RFC 7296] 1.2, 2.21.1, 3.10.1

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - Configure the DUT to prefer a different DH Group from the one specified in the Common Configuration.

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits a valid IKE_SA_INIT Response containing a Notify Paylaod of type INVALID_KE_PAYLOAD (17), specifying the preferred DH Group from the Common Configuration in the Data.	The NUT transmits a valid IKE_SA_INIT Request. It has a KE Payload using the preferred group.

3.	TN1 transmits a valid IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.

Possible Problems:

- The NUT may only support a single DH Group which makes this test impossible.

IPsec.Conf.1.1.1.6: IKE_SA_INIT Exchange; COOKIE and INVALID KE

Purpose:

To verify correct processing and transmission of COOKIE notifications when combined with INVALID_KE_PAYLOAD notifications.

References:

- [RFC 7296] 2.6, 2.6.1

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - Configure the DUT to prefer a different DH Group from the one specified in the Common Configuration.

Procedure:

Part A: Optimized Responder

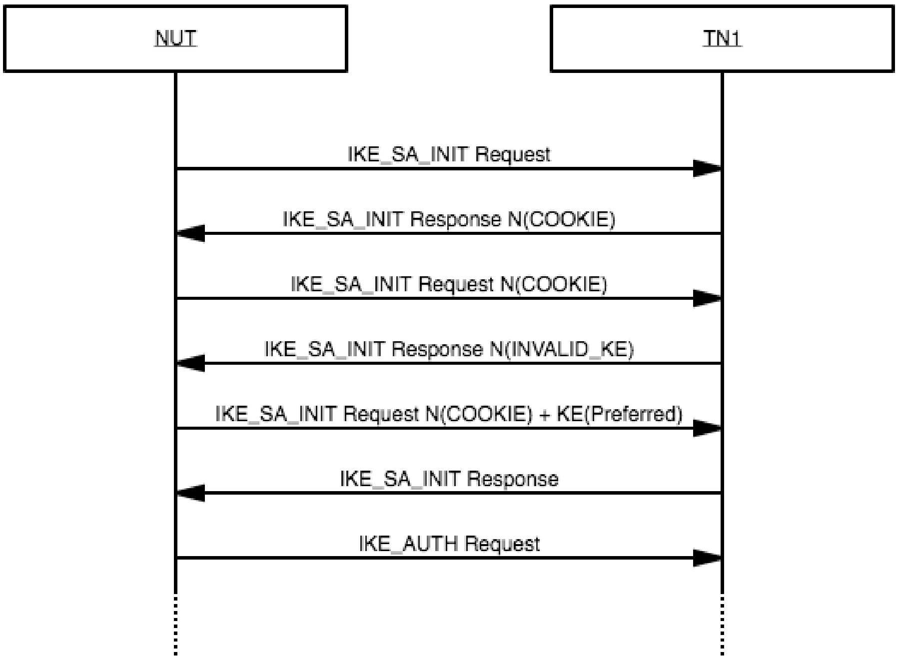


FIGURE 1 - INITIATOR INCLUDES COOKIE IN NEXT REPLY

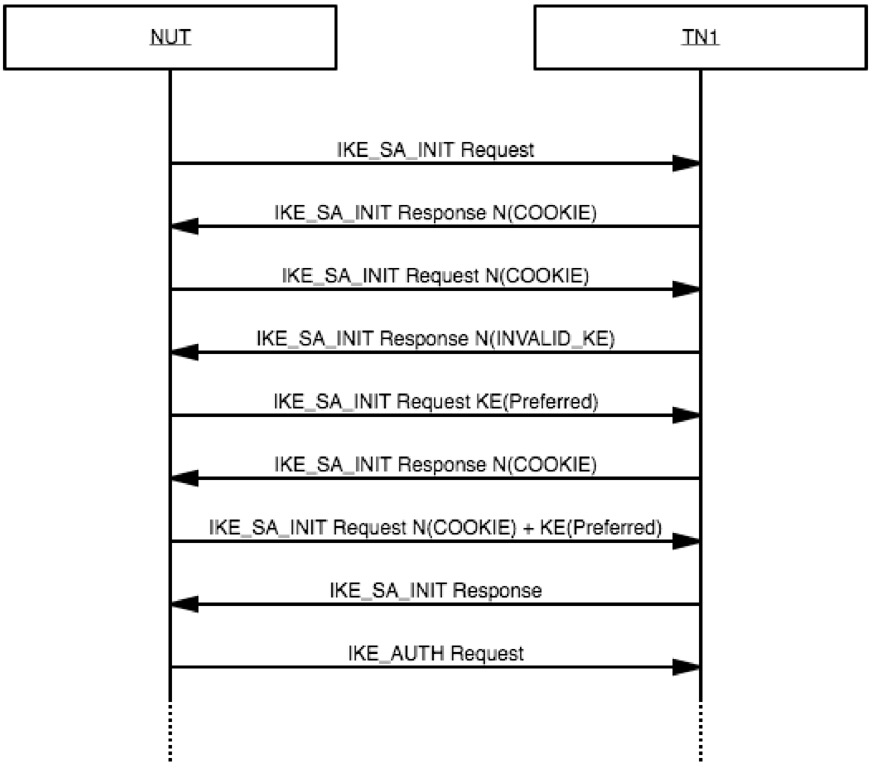
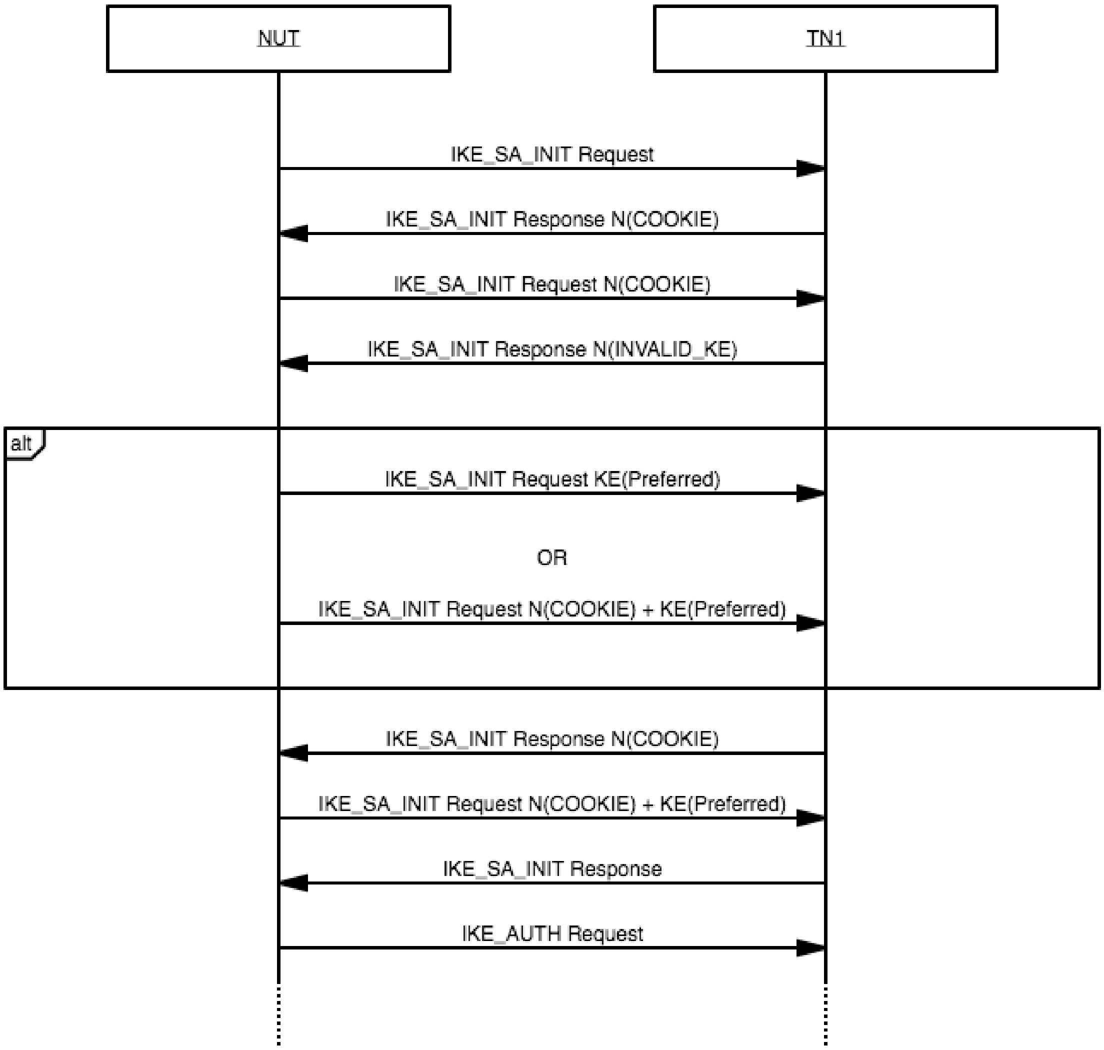


FIGURE 2 - INITIATOR DOES NOT INCLUDE COOKIE IN NEXT REPLY

Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits a valid IKE_SA_INIT Response containing only a Notifiy Payload of type COOKIE (16390) and valid Notification Data.	The NUT transmits a valid IKE_SA_INIT Request unchanged, except for the addition of a COOKIE Notification Payload as the first payload.
3.	TN1 transmits a valid IKE_SA_INIT Response containing a Notify Paylaod of type INVALID_KE_PAYLOAD (17), specifying the preferred DH Group from the Common Configuration in the Data.	The NUT transmits a valid IKE_SA_INIT Request, with a KE Payload using the preferred group. If it also contains a Notify Payload of type COOKIE (Figure 1) proceed to step 5, otherwise, proceed to step 4 (Figure 2),.
4.	TN1 transmits a valid IKE_SA_INIT Response containing a Notify Payload of	The NUT transmits a valid IKE_SA_INIT Request, with a KE Payload using the preferred group. It also contains a Notify Payload of type COOKIE.

	type COOKIE and valid Notifcation Data.	
5.	TN1 transmits a valid IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.

Part B: Unoptimized Responder



Step	Action	Expected Result
6.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
7.	TN1 transmits a valid IKE_SA_INIT Response containing only a Notifiy Payload of type COOKIE	The NUT transmits a valid IKE_SA_INIT Request unchanged, except for the addition of a COOKIE Notification Payload as the first payload.

	(16390) and valid Notification Data.	
8.	TN1 transmits a valid IKE_SA_INIT Response containing a Notify Paylaod of type INVALID_KE_PAYLOAD (17), specifying the preferred DH Group from the Common Configuration in the Data.	The NUT transmits a valid IKE_SA_INIT Request. It has a KE Payload using the preferred group. It may also contain a Notify Payload of type COOKIE.
9.	TN1 transmits a valid IKE_SA_INIT Response containing a Notify Payload of type COOKIE and valid Notifcation Data.	The NUT transmits a valid IKE_SA_INIT Request, with a KE Payload using the preferred group. It also contains a Notify Payload of type COOKIE.
10.	TN1 transmits a valid IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.

Possible Problems:

- The NUT may only support a single DH Group which makes this test impossible.

IPsec.Conf.1.1.1.7: IKE_SA_INIT inconsistent response proposal

Purpose:

To verify correct handling of an IKE_SA_INIT Response with an inconsistent SA Proposal.

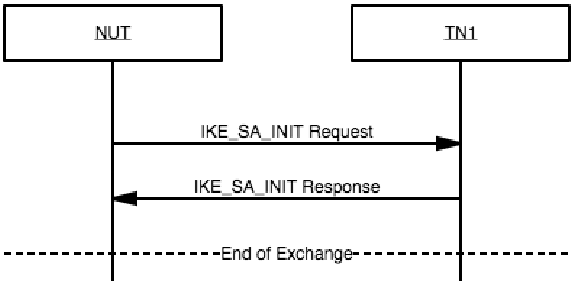
References:

- [RFC 7296] 3.3.6

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits a valid IKE_SA_INIT Response containing an SA Proposal that does not match any of the Requested Proposals.	The NUT does not transmit a valid IKE_AUTH Request.

Possible Problems:

- None.

IPsec.Conf.1.1.1.8: IKE_SA_INIT Forward Compatibility

Purpose:

To verify forward compatibility using the reserved and version fields.

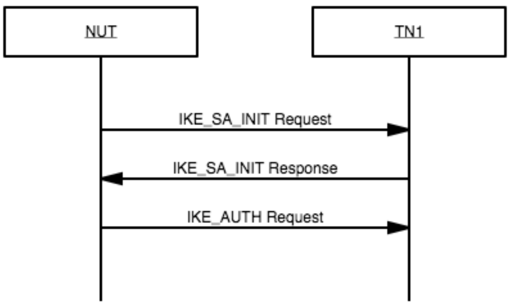
References:

- [RFC 7296] 2.5, 3.1

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Part A: Non-zero Reserved Bits

Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits a valid IKE_SA_INIT Response. The reserved bits in the IKE Header are set to 1.	The NUT transmits a valid IKE_AUTH Request.

Part B: Version Bit Set

Step	Action	Expected Result
3.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.

4.	TN1 transmits a valid IKE_SA_INIT Response. The version bit in the Flags field is set.	The NUT transmits a valid IKE_AUTH Request.

Possible Problems:

- None.

1.1.2. IKE_AUTH Exchange

IPsec.Conf.1.1.2.1: IKE_AUTH Request Format

Purpose:

To verify a properly formatted IKE_AUTH Request

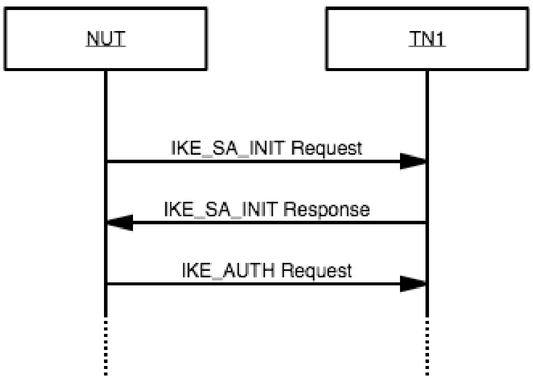
References:

- [RFC 7296] 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13, 3.14

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request. Verify fields according to **Table A** (Encrypted) and **Table B** (Decrypted Payloads) below.

Table A:

Payload	Field	Value
IKE Header	iSPI	Non-Zero
	rSPI	Non-Zero (From IKE_SA_INIT Response)
	Next Payload	Encrypted and Authenticated (46)
	Major Version	2
	Minor Version	0
	Exchange Type	IKE_AUTH (35)
	Flags	(00001000)2 = (08)16
	Message ID	1
Encrypted Payload	Initialization Vector	Valid
	Encrypted IKE Payloads	Valid
	Padding/Pad Length	Valid
	Integrity Checksum Data	Valid

Table B (Payloads within Encrypted IKE Payload):

Payload			Field	Value
ID Payload			ID Type	ID_IPV6_ADDR (5)
			ID Data	Valid
Authentication Payload			Authentication Method	Shared Key Message Integrity Code (2)
			Authentication Data	Valid
Notify Payload			Payload Length	8
			Protocol ID	0
			SPI Size	0
			Notify Message Type	USE_TRANSPORT_MODE (16391)
SA Payload	Proposal		Last	0 or 2
			Proposal #	1
			Protocol ID	ESP (3)
			SPI Size	4
			SPI	Valid
			# Transforms	3
			Last	0 or 3
			Transform Type	ENCR (1)
			Transform ID	(According to Common Configuration)
			Last	0 or 3
			Transform Type	INTEG (3)

			Transform ID	(According to Common Configuration)
			Last	0 or 3
			Transform Type	ESN (5)
			Transform ID	(According to Common Configuration)
TSi			# Traffic Selectors	1 or 2
	Traffic Selector		TS Type	TS_IPV6_ADDR_RANGE (8)
			IP Protocol ID	0
			Selector Length	40
			Start Port	0
			End Port	65535
			Starting Address	NUT IPv6 Address
			Ending Address	NUT IPv6 Address
TSr			# Traffic Selectors	1 or 2
	Traffic Selector		TS Type	TS_IPV6_ADDR_RANGE (8)
			IP Protocol ID	0
			Selector Length	40
			Start Port	0
			End Port	65535
			Starting Address	TN1 IPv6 Address
			Ending Address	TN1 IPv6 Address

Possible Problems:

- The IKE_AUTH Request may have additional payloads not described above and can be ignored. The payloads may be in any order.
- There may be more than one Proposal in the SA Payload. One proposal must match the above.
- SA Payload Proposal Transforms may be in any order
- There may be more than one traffic selector in the TSi and TSr payloads. The last traffic selector must match the above.

IPsec.Conf.1.1.2.2: IKE_AUTH Exchange Succeeds

Purpose:

To verify a IKE_AUTH Exchange completed successfully under normal conditions.

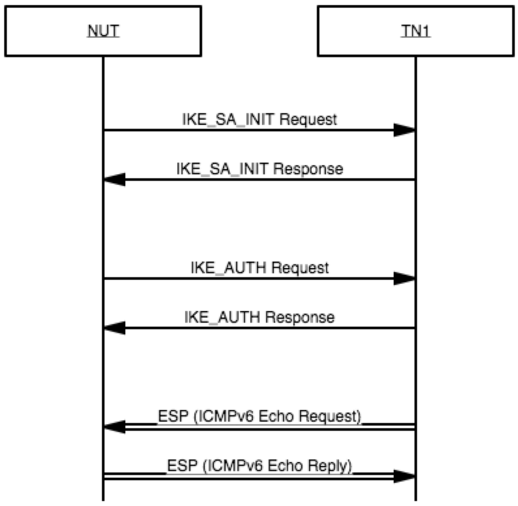
References:

- [RFC 7296] 1.2

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
3.	TN1 transmits a valid IKE_AUTH Response.	
4.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.1.2.3: IKE_AUTH Retransmission

Purpose:

To verify correct retransmission of IKE_AUTH Requests.

References:

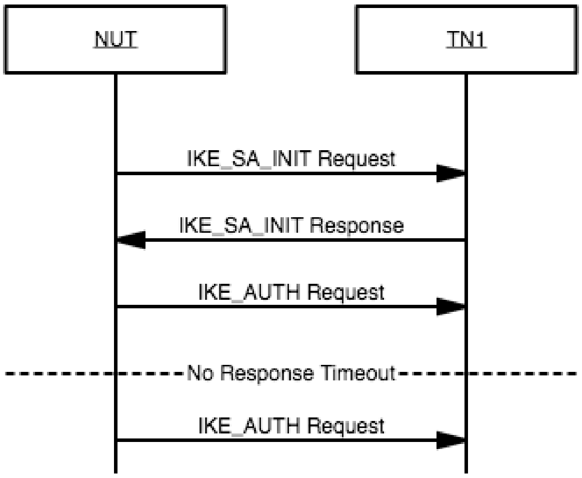
- [RFC 7296] 2.1, 2.2, 2.4

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

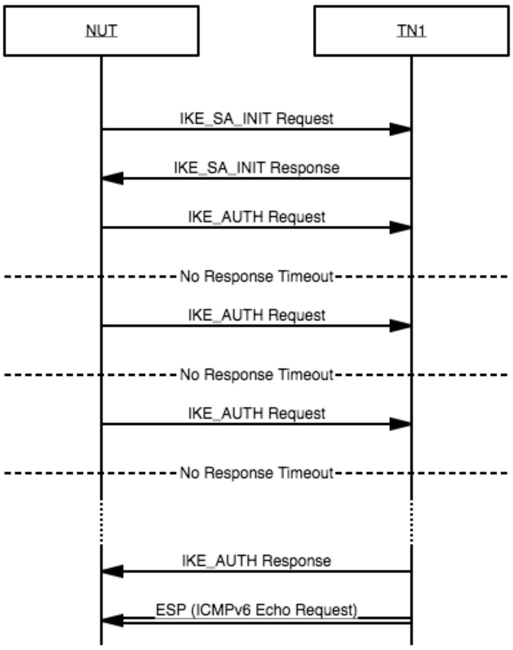
Part A: Retransmission

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
3.	Wait for timeout.	The NUT retransmits a valid IKE_AUTH Request which is bitwise identical to the previously transmitted IKE_AUTH Request.

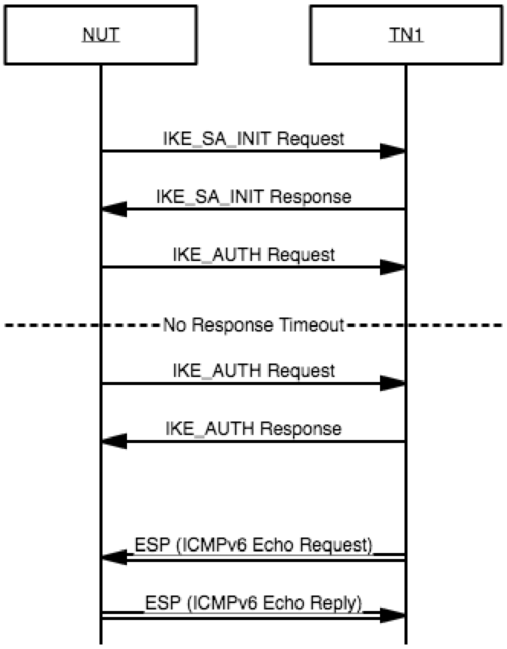
Part B: Retransmission Fails



Step	Action	Expected Result
4.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
5.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
6.	Wait for timeout.	The NUT retransmits a valid IKE_AUTH Request which is bitwise identical to the previously transmitted IKE_AUTH Request.
7.	Wait for final timeout.	The NUT ceases to retransmit IKE_AUTH Request messages.
8.	TN1 transmits an IKE_AUTH Response	
9.	TN1 Transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT does not transmits an ESP ICMPv6 Echo Reply as negotiated.

Part C: Retransmission Succeeds

Procedure:



Step	Action	Expected Result
10.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
11.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
12.	Wait for timeout.	The NUT retransmits a valid IKE_AUTH Request which is bitwise identical to the previously transmitted IKE_AUTH Request.
13.	TN1 transmits an IKE_AUTH Response	
14.	TN1 Transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- The length of the timeout for retransmission is unspecified, and usually not configurable by the user.

IPsec.Conf.1.1.2.4: State Synchronization

Purpose:

To verify IKEv2 state is not lost due to cryptographically unprotected messages.

References:

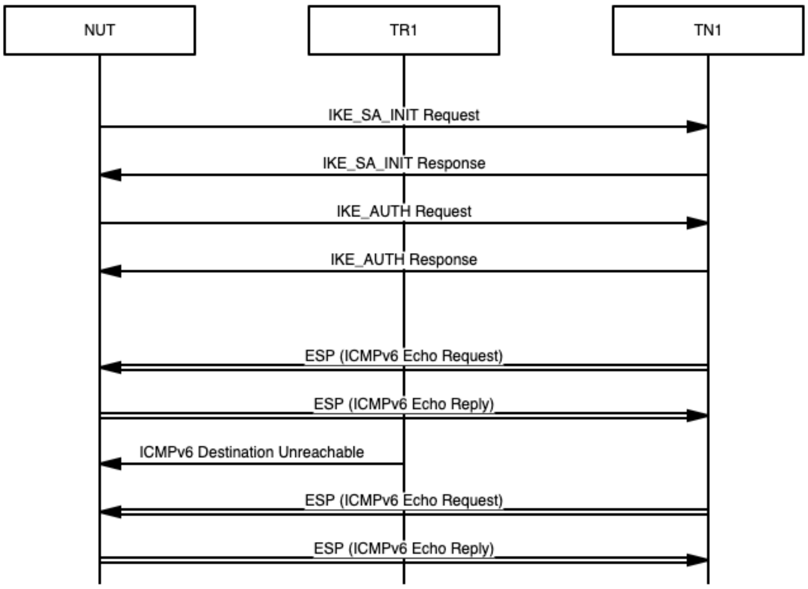
- [RFC 7296] 2.4

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration

Procedure:

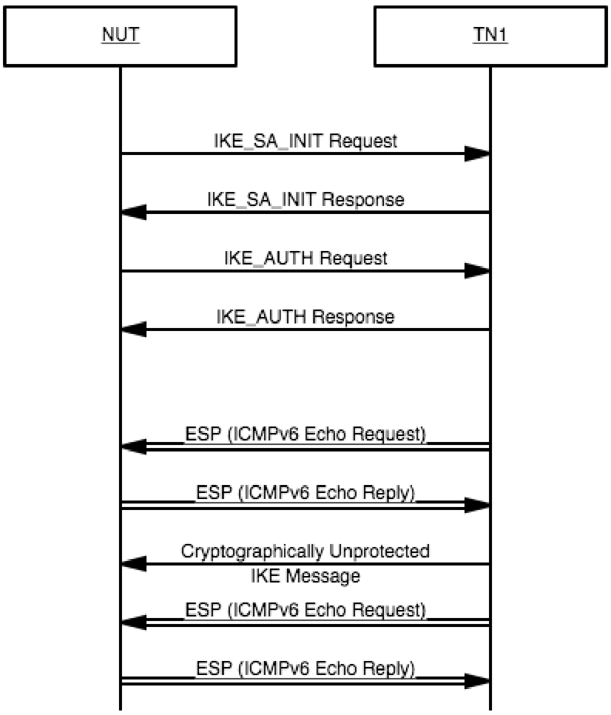
Part A: ICMPv6



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
3.	TN1 transmits a valid IKE_AUTH Response.	
4.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.
5.	TR1 transmits an ICMPv6 Destination Unreachable Message to the NUT.	
6.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Payload	Field	Value
IKE Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	41 (N)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	any
	Length	any
Notify Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	3 (ESP)
	SPI Size	0
	Notify Message Type	11 (INVALID_SPI)

PACKET 1 - CRYPTOGRAPHICALLY UNPROTECTED IKE MESSAGE



Step	Action	Expected Result
7.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
8.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
9.	TN1 transmits a valid IKE_AUTH Response.	
10.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.
11.	TN1 transmits a cryptographically unprotected IKE Message	
12.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.1.2.5: IKE_AUTH Cryptographic Algorithm Negotiation

Purpose:

To verify algorithm negotiation during IKE_AUTH for ESP CHILD_SA

References:

- [RFC 7296] 2.7, 3.3.2, 3.3.5
- [RFC 8221]

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - Use the following Transforms corresponding to each part:

Common Configuration:

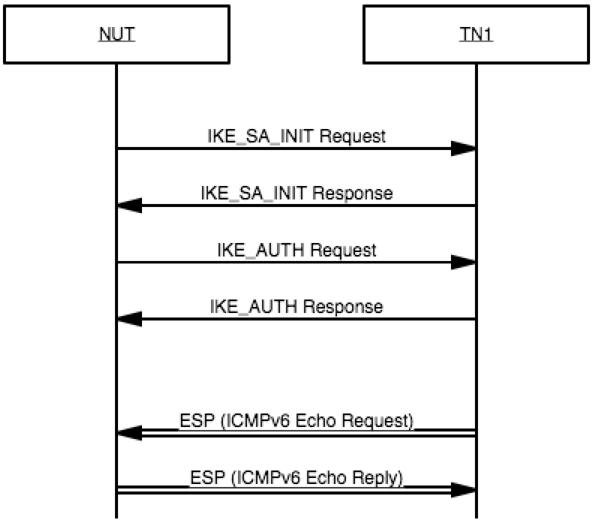
Type	Transform
ENCR (1)	ENCR_AES_CBC 128-bit (12)
INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
ESN (5)	No ESN (0)

The device should send the transforms specified in the Common Configuration. The table below substitutes the given Transform Type and Transform ID according to the part specified.

Part	Transform Type	Transform ID
A	ENCR (1)	ENCR_AES_CBC 128-bit (12)
	INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
B - AES256	ENCR (1)	ENCR_AES_CBC 256-bit (12)
	INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
C - CHACHA	ENCR (1)	ENCR_CHACHA20_POLY1305 (28)
	INTEG (3)	Omitted or NONE (0)
D - AESGCM	ENCR (1)	ENCR_AES_GCM_16 128-bit (20)
	INTEG (3)	Omitted or NONE (0)
E - AESCCM	ENCR (1)	ENCR_AES_CCM_8 128-bit (14)
	INTEG (3)	Omitted or NONE (0)
F -NULL	ENCR (1)	ENCR_NULL (11)
	INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
G - SHA512	ENCR (1)	ENCR_NULL (11)

	INTEG (3)	AUTH_HMAC_SHA2_512_256 (14)
H - AESXCBC	ENCR (1)	ENCR_NULL (11)
	INTEG (3)	AUTH_AES_XCBC_96 (5)

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits a valid IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request with transforms according to the table above.
3.	TN1 transmits a valid IKE_AUTH Response.	
4.	TN1 Transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.1.2.6: IKE_AUTH N(NO_PROPOSAL_CHOSEN)

Purpose:

To verify an IKE_SA remains setup after reception of N(NO_PROPOSAL_CHOSEN) in IKE_AUTH.

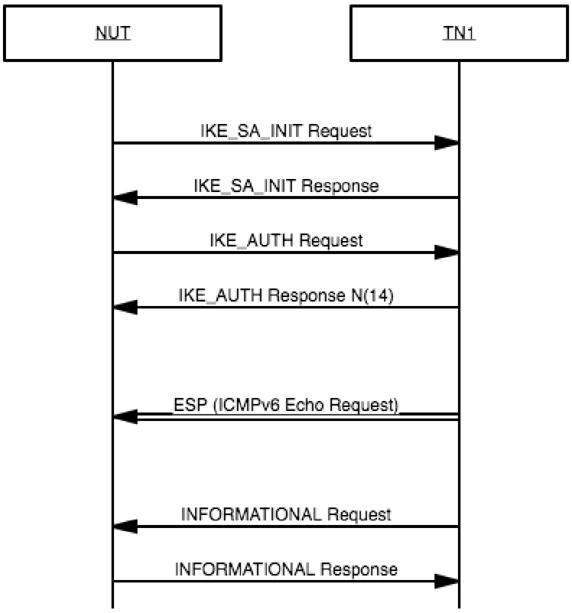
References:

- [RFC 7296] 1.2, 2.7, 2.21.2, 3.10.1

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
3.	TN1 transmits an IKE_AUTH Response. It does not contain an SA Payload, or any Traffic Selector Payload. It does contain a Notify Payload of type NO_PROPOSAL_CHOSEN (14). It is otherwise valid.	
4.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT does not transmit an ESP ICMPv6 Echo Reply as negotiated.
5.	TN1 transmits an INFORMATIONAL Request with an Encrypted Payload with no data.	The NUT transmits an INFORMATIONAL Response with an Encrypted Payload with no data.

Possible Problems:

- None.

IPsec.Conf.1.1.2.7: IKE_AUTH Inconsistent response proposal

Purpose:

To verify an IKE_SA remains setup after reception of an KE_AUTH response with an inconsistent proposal.

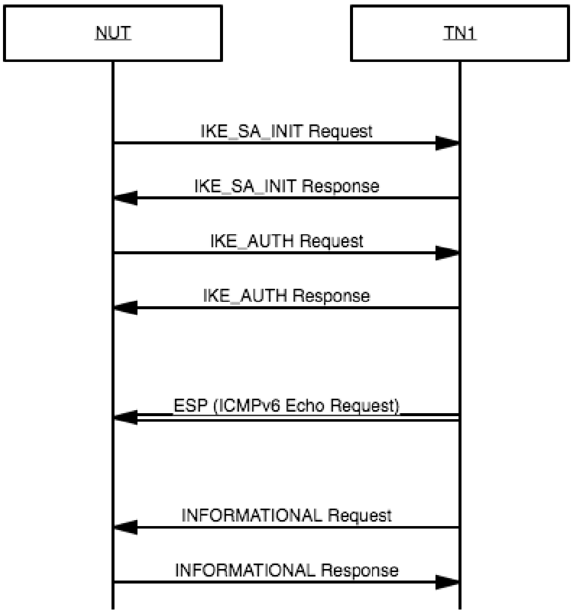
References:

- [RFC 7296] 2.21.2, 3.3.6

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
3.	TN1 transmits a valid IKE_AUTH Response containing an SA Proposal that does not match any of the Requested Proposals.	
4.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT does not transmit an ESP ICMPv6 Echo Reply as negotiated.
5.	TN1 transmits an INFORMATIONAL Request with an Encrypted Payload with no data.	The NUT transmits an INFORMATIONAL Response with an Encrypted Payload with no data.

Possible Problems:

- None.

IPsec.Conf.1.1.2.8: Traffic Selector Negotiation

Purpose:

To verify a device is able to process an IKE_AUTH Response with Traffic Selectors configured to be more narrow than was originally proposed.

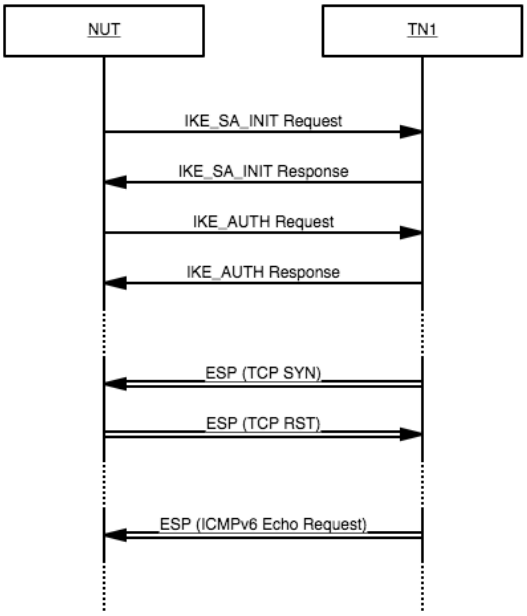
References:

- [RFC 7296] 2.9

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - No additional Traffic Selector Configuration is done

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request. It contains traffic selectors matching ANY protocol, all ports, and addresses specific to TN1 and the NUT. It is unchanged from the Common Configuration.
3.	TN1 transmits a valid IKE_AUTH Response. The traffic selectors in the response specify an IP Protocol ID of TCP (6), for TSi and TSr.	
4.	TN1 transmits a TCP-SYN packet with IPsec ESP to a closed port on the NUT.	The NUT transmits a valid ESP TCP RST to TN1.
5.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT does not transmit an ESP ICMPv6 Echo Reply, nor does it transmit a non-encrypted ICMPv6 Echo Reply.

Possible Problems:

- The NUT may transmit a CREATE_CHILD_SA Request with Traffic Selectors matching ICMPv6 Echo Reply. This does not indicate a failure.
- A Security Gateway device may have additional Traffic Selectors, or Traffic Selectors representing a range of addresses. This should not be considered a failure.

IPsec.Conf.1.1.2.9: Peer Identification

Purpose:

To verify authentication using different Identification Types.

References:

- [RFC 7296] Sections 2.15, 3.5

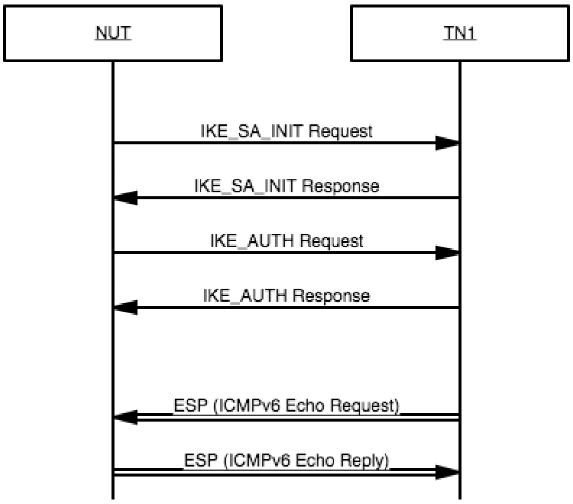
Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - Configure the devices to authenticate using the Identification Types according to each part specified in the table below.

TABLE 1 - IDENTIFICATION TYPES

Part	NUT ID Type	TN1 ID Type
A	ID_IPV6_ADDR	ID_IPV6_ADDR
B	ID_IPV6_ADDR	ID_FQDN
C	ID_IPV6_ADDR	ID_RFC822_ADDR
D	ID_FQDN	ID_IPV6_ADDR
E	ID_FQDN	ID_FQDN
F	ID_FQDN	ID_RFC822_ADDR
G	ID_RFC822_ADDR	ID_IPV6_ADDR
H	ID_RFC822_ADDR	ID_FQDN
I	ID_RFC822_ADDR	ID_RFC822_ADDR

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request. The IDi Payload contains an ID Type according to the part in the NUT ID Type column in the table above (Table 1).
3.	TN1 transmits a valid IKE_AUTH Response. The IDr Payload contains an ID Type according to the part in the TN1 ID Type column in the table above (Table 1).	
4.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.1.2.10: Authentication via RSA Digital Signature

Purpose:

To verify authentication of a peer via RSA Digital Signature

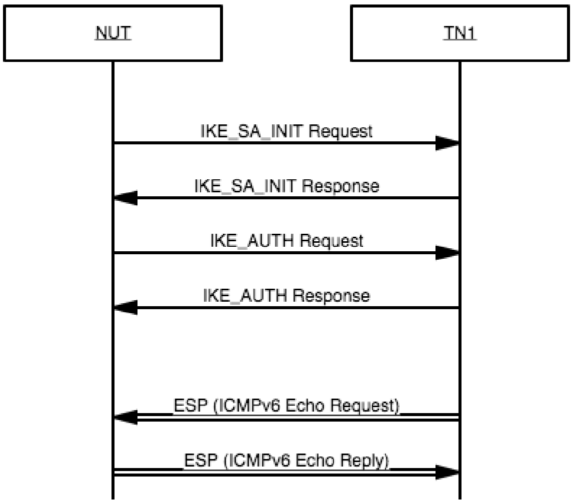
References:

- [RFC 7296] Sections 2.15, 3.5, 3.6, 3.7, 3.8

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - Configure the devices to use the RSA Digital Signature (1) Authentication Method for both DUT and TN1, in place of Shared Key (3).

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response. The Response contains a CERTREQ Payload specifying X.509 Certificate - Signature (4) and the data corresponding to the preferred CA.	The NUT transmits a valid IKE_AUTH Request. The Authentication Payload specifies an Auth Method of RSA Digital Signature (1), and contains valid authentication data. If the request contains a CERTREQ Payload, it is valid and formatted properly. If the request contains a CERT Payload, it is valid and formatted properly.
3.	TN1 transmits a valid IKE_AUTH Response. It contains a CERT Payload, it is valid and formatted properly. The certificate specified is the one used for authentication. The Authentication Payload specifies an Auth Method of	

	RSA Digital Signature (1), and contains valid authentication data.	
4.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.1.2.11: Authentication via PSK

Purpose:

To verify authentication of a peer via Shared Key Message Integrity Code

References:

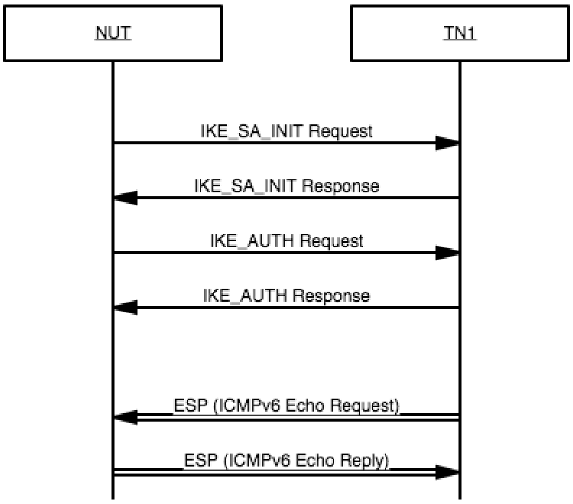
- [RFC 7296] Sections 2.15

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:

Part A: Authentication Succeeds



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request. The Authentication Payload specifies an Auth Method of Shared Key Message Integrity Code (2), and contains valid authentication data.
3.	TN1 transmits a valid IKE_AUTH Response. The Authentication Payload specifies an Auth Method of Shared Key Message Integrity Code (2), and contains valid authentication data.	
4.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Part B: Authentication with Hex Encoding of PSK

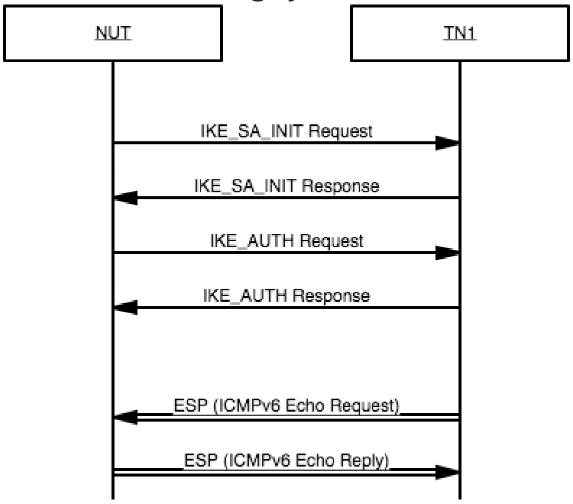


TABLE 2 - HEX PSK

PSK	NUT and TN1
Local & Remote	0xabadcafeabadcafeabadcafeabadcafe

Step	Action	Expected Result
5.	Initialize the NUT. Use the HEX PSK specified in the table above. (Table 2)	The NUT transmits a valid IKE_SA_INIT Request.
6.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request. The Authentication Payload specifies an Auth Method of Shared Key Message Integrity Code (2), and contains valid authentication data.
7.	TN1 transmits a valid IKE_AUTH Response. The Authentication Payload specifies an Auth Method of Shared Key Message Integrity Code (2), and contains valid authentication data.	
8.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Part C: Authentication with PSK fails

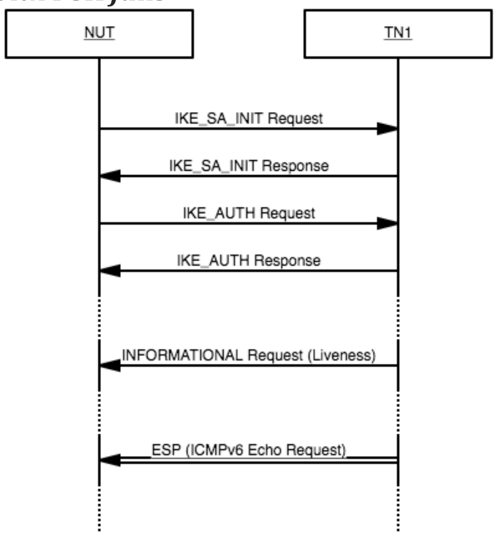


TABLE 3 - MISMATCHED PSK

PSK	NUT	TN1
Local & Remote	"IKETEST-1234"	"NOMATCH"

Step	Action	Expected Result
9.	Initialize the NUT. Use the HEX PSK specified in the table above. (Table 3)	The NUT transmits a valid IKE_SA_INIT Request.
10.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request. The Authentication Payload specifies an Auth Method of Shared Key Message Integrity Code (2), and contains valid authentication data.
11.	TN1 transmits a valid IKE_AUTH Response. The Authentication Payload specifies an Auth Method of Shared Key Message Integrity Code (2), and contains valid authentication data.	
12.	TN1 transmits an INFORMATIONAL Request	The NUT does not transmit a response to the liveness check.

	liveness check with no payloads except for an empty Encrypted Payload.	
13.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT does not transmit an ESP ICMPv6 Echo Reply.

Possible Problems:

- **Possible Problem Part B:** This is a true "byte" representation of a key. This key cannot be represented via ASCII input, and must be handled separately. For example: The ASCII byte representation of "abadcafeabadcafeabadcafeabadcafe" is 0x 6162616463616665616261646361666561626164636166656162616463616665, which is not equal to the HEX PSK given.
- **Possible Problem Part C:** The NUT MAY send an INFORMATIONAL Response without cryptographic protection with a notification of INVALID_IKE_SPI.

IPsec.Conf.1.1.2.12: IKE_AUTH Forward Compatibility

Purpose:

To verify that the contents of the IKE_AUTH Response Reserved field are ignored.

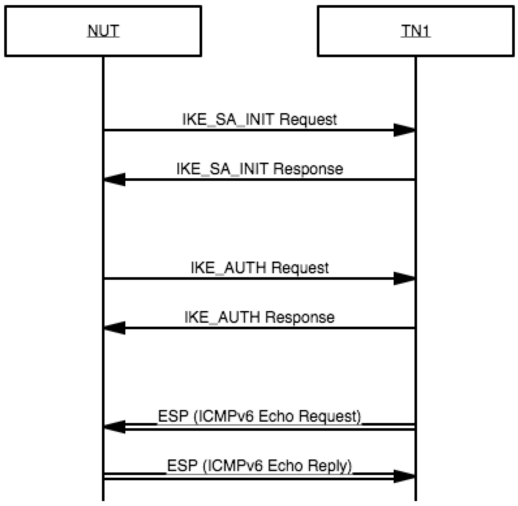
References:

- [RFC 7296] 2.5, 3.1

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
3.	TN1 transmits a valid IKE_AUTH Response. The reserved bits in the IKE Header are set to 1.	
4.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.1.2.13: IKE_AUTH Unrecognized Error

Purpose:

To verify correct handling of unrecognized error notifications.

References:

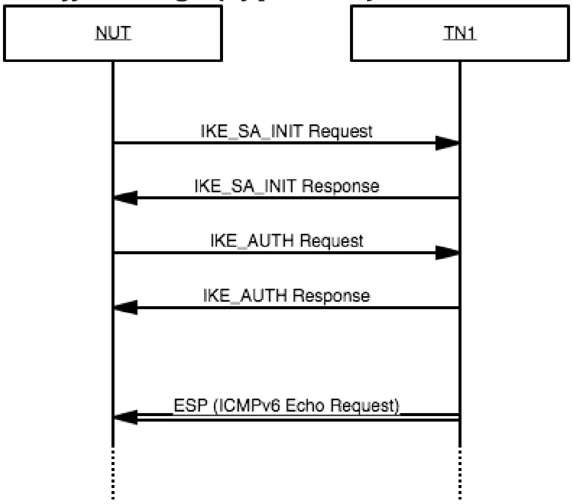
- [RFC 7296] 3.10.1

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

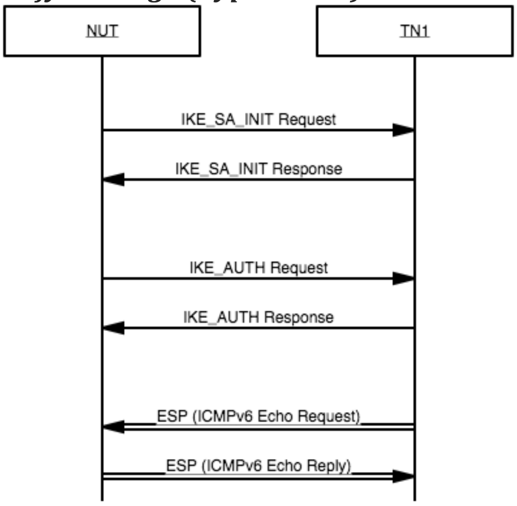
Procedure:

Part A: Unrecognized Notify Message (Type Error)



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
3.	TN1 transmits a valid IKE_AUTH Response. It contains a Notify payload of unrecognized Notify Message Type value (16383).	
4.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT does not transmit a valid ESP ICMPv6 Echo Reply as negotiated.

Part B: Unrecognized Notify Message (Type Status)



Step	Action	Expected Result
5.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
6.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
7.	TN1 transmits a valid IKE_AUTH Response. It contains a Notify payload of unrecognized Notify Message Type value (65535).	
8.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

1.1.3. IKE_AUTH Exchange - Tunnel Mode

IPsec.Conf.1.1.3.1: IKE_AUTH Request Format in Tunnel Mode

Purpose:

To verify a properly formatted IKE_AUTH Request in Tunnel Mode

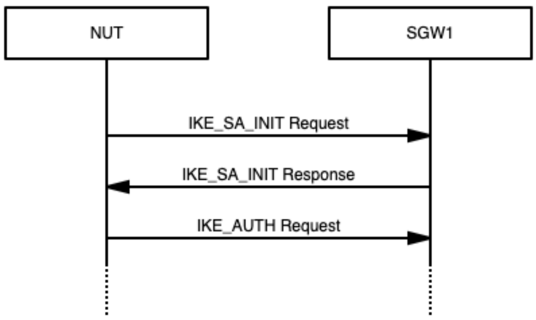
References:

- [RFC 7296] 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13, 3.14

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	SGW1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request. Verify fields according to **Table A** (Encrypted) and **Table B** (Decrypted Payloads) below. The NUT uses **TUNNEL Mode**, with Traffic Selectors matching **Network2.**

Table A:

Payload	Field	Value
IKE Header	iSPI	Non-Zero

	rSPI	Non-Zero (From IKE_SA_INIT Response)
	Next Payload	Encrypted and Authenticated (46)
	Major Version	2
	Minor Version	0
	Exchange Type	IKE_AUTH (35)
	Flags	(00001000)2 = (08)16
	Message ID	1
Encrypted Payload	Initialization Vector	Valid
	Encrypted IKE Payloads	Valid
	Padding/Pad Length	Valid
	Integrity Checksum Data	Valid

Table B (Payloads within Encrypted IKE Payload):

Payload			Field	Value
ID Payload			ID Type	ID_IPV6_ADDR (5)
			ID Data	Valid
Authentication Payload			Authentication Method	Shared Key Message Integrity Code (2)
			Authentication Data	Valid
SA Payload	Proposal		Last	0 or 2
			Proposal #	1
			Protocol ID	ESP (3)
			SPI Size	4
			SPI	Valid
			# Transforms	3
			Last	0 or 3
			Transform Type	ENCR (1)
			Transform ID	(According to Common Configuration)
			Last	0 or 3
			Transform Type	INTEG (3)
			Transform ID	(According to Common Configuration)
			Last	0 or 3
			Transform Type	ESN (5)
			Transform ID	(According to Common Configuration)
TSi			# Traffic Selectors	1 or 2
	Traffic Selector		TS Type	TS_IPV6_ADDR_RANGE (8)
			IP Protocol ID	0
			Selector Length	40

		Start Port	0
		End Port	65535
		Starting Address	NUT IPv6 Address
		Ending Address	NUT IPv6 Address
TSr		# Traffic Selectors	1 or 2
	Traffic Selector	TS Type	TS_IPV6_ADDR_RANGE (8)
		IP Protocol ID	0
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NETWORK2::0000
		Ending Address	NETWORK2::FFFF

Possible Problems:

- The IKE_AUTH Request may have additional payloads not described above and can be ignored. The payloads may be in any order.
- There may be more than one Proposal in the SA Payload. One proposal must match the above.
- SA Payload Proposal Transforms may be in any order
- There may be more than one traffic selector in the TSi and TSr payloads. The last traffic selector must match the above.

IPsec.Conf.1.1.3.2: IKE_AUTH Exchange Succeeds in Tunnel Mode

Purpose:

To verify a IKE_AUTH Exchange completed successfully under normal conditions utilizing Tunnel Mode.

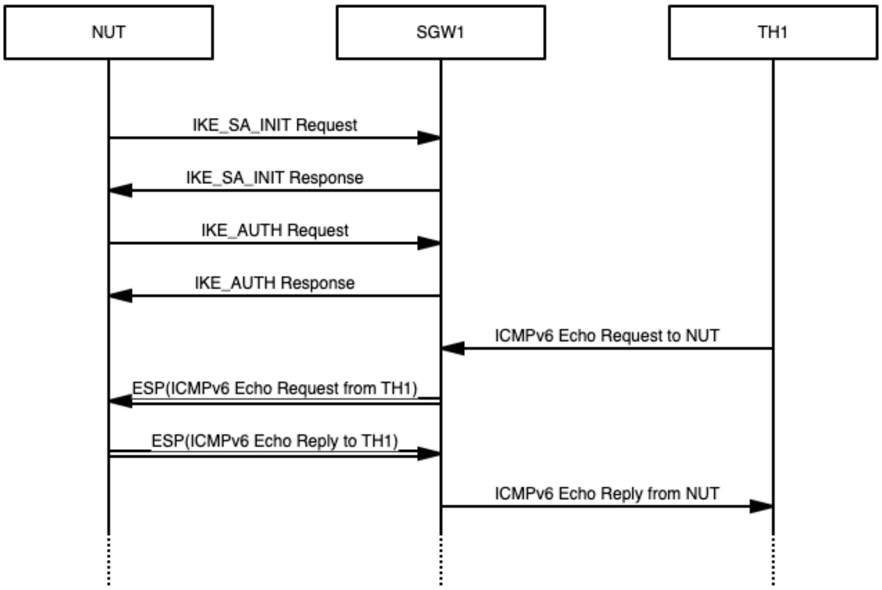
References:

- [RFC 7296] 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13, 3.14

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	SGW1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
3.	SGW1 transmits a valid IKE_AUTH Response.	
4.	SGW1 transmits an ESP Tunneled ICMPv6 Echo Request as negotiated on behalf of TH1.	The NUT transmits a valid ESP Tunneled ICMPv6 Echo Reply as negotiated in response to TH1.

Possible Problems:

- None.

1.1.4. CREATE_CHILD_SA Exchange

1.1.5. INFORMATIONAL Exchange

IPsec.Conf.1.1.5.1: IKE_SA Deletion

Purpose:

To verify transmission of IKE_SA Delete Payload.

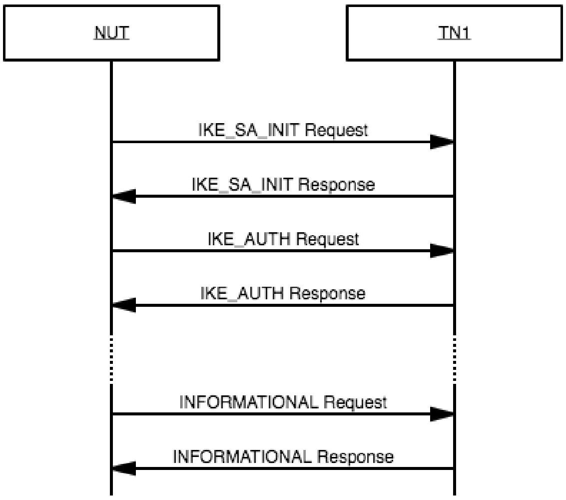
References:

- [RFC 7296] 1.4.1, 2.4

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
3.	Wait for IKE_SA Lifetime to expire, or cause the NUT to delete the IKE_SA.	The NUT transmits a valid INFORMATIONAL Request with a DELETE Payload according to Table A below.
4.	TN1 transmits an INFORMATIONAL Response to delete the IKE_SA.	The NUT does not transmit any further IKEv2 messages using this IKE_SA.

Table A:

Payload	Field	Value
DELETE Payload	Protocol ID	IKE (1)
	SPI Size	0

Possible Problems:

- It may be impossible to cause the device to delete an SA.
- The NUT may transmit an INFORMATIONAL Request with a Delete Payload including 2 (ESP) as Protocol ID, 4 as SPI Size and SPI value to delete CHILD_SA before transmitting an INFORMATIONAL Request to delete IKE_SA.

IPsec.Conf.1.1.5.2: CHILD_SA Deletion

Purpose:

To verify transmission of CHILD_SA Delete Payload.

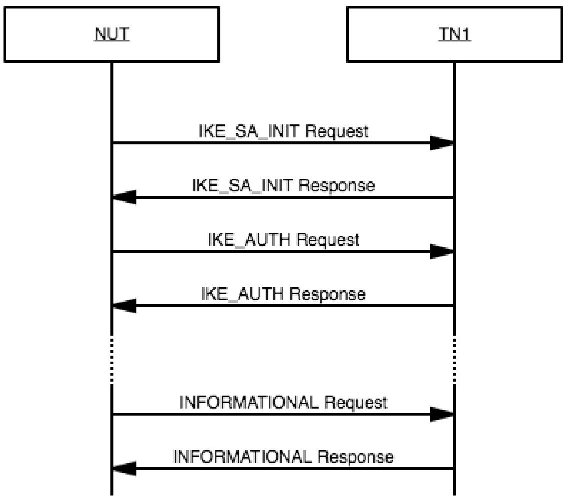
References:

- [RFC 7296] 1.4.1, 2.4

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT.	The NUT transmits a valid IKE_SA_INIT Request.
2.	TN1 transmits an IKE_SA_INIT Response.	The NUT transmits a valid IKE_AUTH Request.
3.	Wait for CHILD_SA Lifetime to expire or cause the NUT to delete the CHILD_SA.	The NUT transmits a valid INFORMATIONAL Request with a DELETE Payload according to Table A below.

Table A:

Payload	Field	Value
DELETE Payload	Protocol ID	ESP (1)
	SPI Size	4
	# SPIs	1
	SPI	CHILD_SA SPI

Possible Problems:

- It may be impossible to cause the device to delete an SA.
- The NUT may transmit an INFORMATIONAL Request with a Delete Payload to delete the IKE_SA, which deletes all CHILD_SA SPIs implicitly.

1.2. Responder

1.2.1. IKE_SA_INIT Exchange

IPsec.Conf.1.2.1.1: IKE_SA_INIT Response Format

Purpose:

To verify a properly formatted IKE_SA_INIT Response

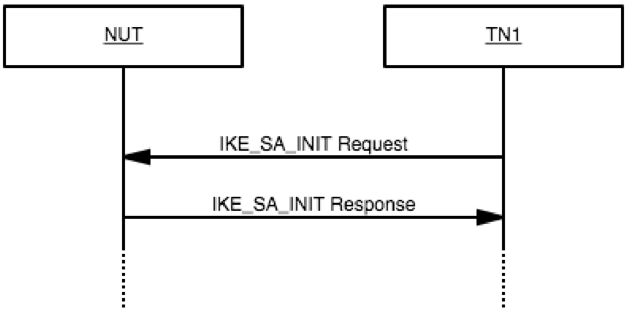
References:

- [RFC 7296] 1.2, 2.10, 3.1, 3.2, 3.3, 3.4, 3.9

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	TN1 transmits an IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response. Verify fields according to **Table A** below. The accepted SA must align with that proposed by TN1.

Table A:

Payload			Field	Value
IKE Header			iSPI	Non-Zero, equal to iSPI in IKE_SA_INIT Request
			rSPI	Non-Zero
			Major Version	2
			Minor Version	0
			Exchange Type	IKE_SA_INIT (34)
			Flags	(00100000)2 = (20)16
			Message ID	0
SA Payload	Proposal		Last	0 or 2
			Proposal #	1
			Protocol ID	IKE (1)
			SPI Size	0
			# Transforms	4
		Transform	Last	0 or 3
			Transform Type	ENCR (1)
			Transform ID	(According to Common Configuration)
		Transform	Last	0 or 3
			Transform Type	PRF (2)
			Transform ID	(According to Common Configuration)
		Transform	Last	0 or 3
			Transform Type	INTEG (3)
			Transform ID	(According to Common Configuration)
		Transform	Last	0 or 3
			Transform Type	DH (4)
			Transform ID	(According to Common Configuration)
KE Payload			DH Group	(According to Common Configuration)
			Key Exchange Data	(According to DH Group)
Nonce Payload			Nonce Data	Unique value of length 16-256 octets

Possible Problems:

- The IKE_SA_INIT Response may have additional payloads not described above and can be ignored. The payloads may be in any order.
- SA Payload Proposal Transforms may be in any order.
- SA Payload Proposal Transforms may be in any order.

IPsec.Conf.1.2.1.2: IKE_SA_INIT Retransmission

Purpose:

To verify correct retransmission of IKE_SA_INIT Response

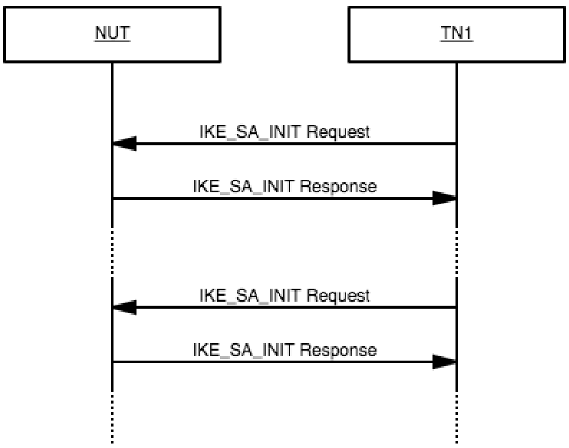
References:

- [RFC 7296] 2.1, 2.2, 2.4

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	TN1 transmits an IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	Wait 10 seconds.	The NUT does not transmit any IKEv2 packets for the newly initiated session.
3.	TN1 retransmits the IKE_SA_INIT Request from Step 1.	The NUT transmits a valid IKE_SA_INIT Response that is bitwise identical to the one transmitted in Step 1.

Possible Problems:

- None.

IPsec.Conf.1.2.1.3: IKE_SA_INIT Cryptographic Algorithm Negotiation

Purpose:

To verify algorithm negotiation during IKE_SA_INIT for IKE_SA

References:

- [RFC 7296] 2.7, 3.3, 3.3.2, 3.3.5
- [RFC 8247]

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - Use the following Transforms corresponding to each part:

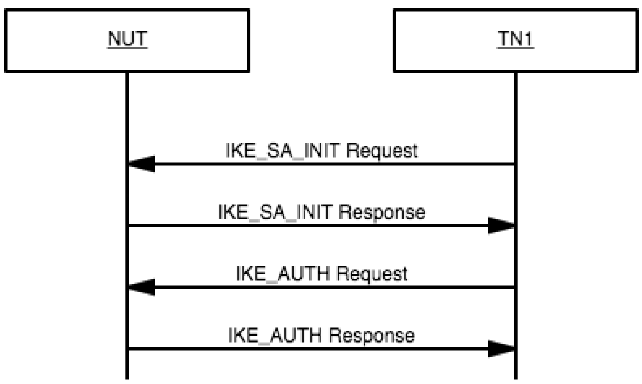
Common Configuration:

Type	Transform
ENCR (1)	ENCR_AES_CBC 128-bit (12)
PRF (2)	PRF_HMAC_SHA2_256 (5)
INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
DH (4)	2048-bit MODP Group (14)

The device should send the transforms specified in the Common Configuration. The table below substitutes the given Transform Type and Transform ID according to the part specified.

Part	Transform Type	Transform ID
A	ENCR (1)	ENCR_AES_CBC 128-bit (12)
	PRF (2)	PRF_HMAC_SHA2_256 (5)
	INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
	DH (4)	2048-bit MODP Group (14)
B - AES256	ENCR (1)	ENCR_AES_CBC 256-bit (12)
C -CHACHA	ENCR (1)	ENCR_CHACHA20_POLY1305 (28)
	INTEG (3)	Omitted or NONE (0)
D - AESGCM	ENCR (1)	ENCR_AES_GCM_16 128-bit (20)
	INTEG (3)	Omitted or NONE (0)
E - AESCCM	ENCR (1)	ENCR_AES_CCM_8 128-bit (14)
	INTEG (3)	Omitted or NONE (0)
F - SHA512	PRF (2)	PRF_HMAC_SHA2_512 (7)
	INTEG (3)	AUTH_HMAC_SHA2_512_256 (14)
G - AESXCBC	PRF (2)	PRF_AES128_XCBC (4)
	INTEG (3)	AUTH_AES_XCBC_96 (5)
H - DH19	DH (4)	256-bit random ECP group (19)

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request with transforms according to the table above.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request encrypted with the algorithms according to the table above.	The NUT transmits a valid IKE_AUTH Response encrypted with the algorithms according to the table above.

Possible Problems:

- None.

IPsec.Conf.1.2.1.4: IKE_SA_INIT Version Number

Purpose:

To verify correct processing of a higher version number.

References:

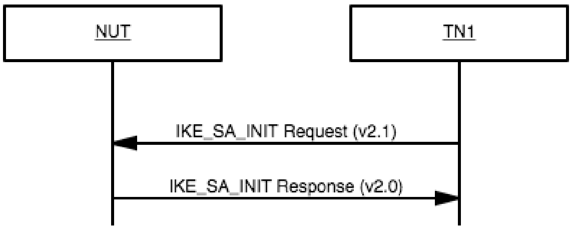
- [RFC 7296] 2.1, 2.2, 2.4, 2.5

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration

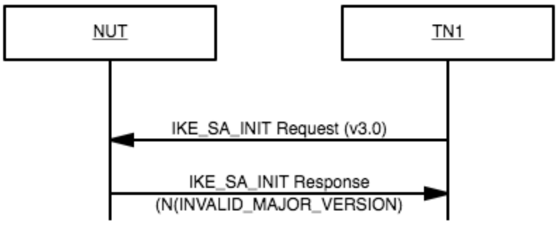
Procedure:

Part A: Higher Minor Version Number



Step	Action	Expected Result
1.	TN1 transmits an IKE_SA_INIT Request with a Major Version of 2 and a Minor Version of 1.	The NUT transmits a valid IKE_SA_INIT Response.

Part B: Higher Major Version Number



Step	Action	Expected Result
2.	TN1 transmits an IKE_SA_INIT Request with a Major Version of 3 and a Minor Version of 0.	The NUT transmits a valid IKE_SA_INIT Response containing a Notify Payload of Type INVALID_MAJOR_VERSION (5).

Possible Problems:

- In Part B, the device MUST drop the message and SHOULD send the INVALID_MAJOR_VERSION Notification. With a valid reason, an implementation may not support sending this notification.

IPsec.Conf.1.2.1.5: IKE_SA_INIT Multiple Transforms

Purpose:

To verify correct processing of an SA Proposal with Multiple Transforms of a single type.

References:

- [RFC 7296] 2.7, 3.3, 3.3.2, 3.3.5

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - Use the following Transforms corresponding to each part:

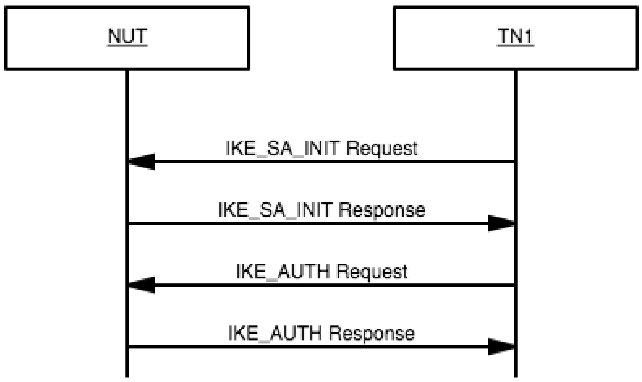
Common Configuration:

Type	Transform
ENCR (1)	ENCR_AES_CBC 128-bit (12)
PRF (2)	PRF_HMAC_SHA2_256 (5)
INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
DH (4)	2048-bit MODP Group (14)

The device should send the transforms specified in the Common Configuration. The table below adds the given Transform Type and Transform ID to the proposal according to the part specified, so that 5 transforms are proposed (6 in the case of Part B), with at two of the same type.

Part	Transform Type	Transform ID
A	ENCR (1)	ENCR_AES_CBC 256-bit (12)
B	PRF (2)	PRF_HMAC_SHA2_512 (7)
	INTEG (3)	AUTH_HMAC_SHA2_512_256 (14)
C	DH (4)	256-bit random ECP group (19)

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request with transforms according to the table above.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request encrypted with the algorithms according to the table above.	The NUT transmits a valid IKE_AUTH Response encrypted with the algorithms according to the table above.

Possible Problems:

- None.

IPsec.Conf.1.2.1.6: IKE_SA_INIT Multiple Proposals

Purpose:

To verify correct processing of an SA Proposal with Multiple Proposals type.

References:

- [RFC 7296] 2.7, 3.3, 3.3.2, 3.3.5

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - Use the following Transforms corresponding to each part:

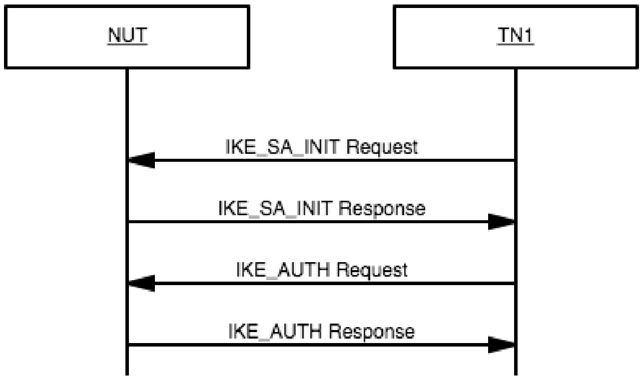
Common Configuration:

Type	Transform
ENCR (1)	ENCR_AES_CBC 128-bit (12)
PRF (2)	PRF_HMAC_SHA2_256 (5)
INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
DH (4)	2048-bit MODP Group (14)

The device should send the transforms specified in the Common Configuration. Additionally, it should send a second proposal according to the table below, for a total of 2 SA Proposals.

Type	Transform
ENCR (1)	ENCR_AES_CBC 256-bit (12)
PRF (2)	PRF_HMAC_SHA2_512 (7)
INTEG (3)	AUTH_HMAC_SHA2_512_256 (14)
DH (4)	256-bit random ECP group (19)

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request with 2 SA Proposals according to the table above.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request encrypted with the algorithms according to the table above.	The NUT transmits a valid IKE_AUTH Response encrypted with the algorithms according to the table above.

Possible Problems:

- None.

IPsec.Conf.1.2.1.7: IKE_SA_INIT Exchange with INVALID_KE_PAYLOAD

Purpose:

To verify correct processing of N(INVALID_KE_PAYLOAD) during IKE_SA_INIT

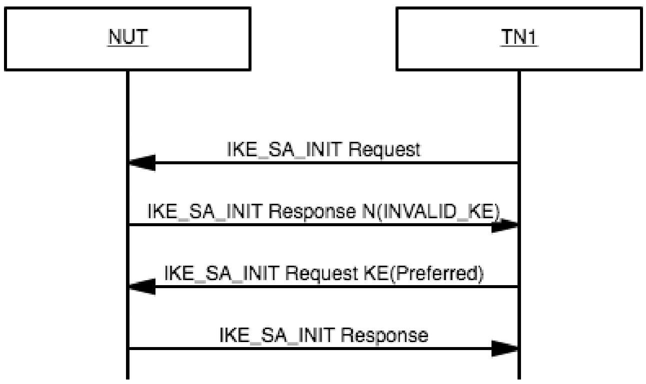
References:

- [RFC 7296] 1.2, 2.2, 2.6, 2.21.1, 3.4

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Payload			Field	Value
IKE Header			*According to Common Configuration*	
SA Payload	Proposal	Transform	ENCR_AES_CBC 128-bit (12)	
		Transform	PRF_HMAC_SHA2_256 (5)	
		Transform	AUTH_HMAC_SHA2_256_128 (12)	
		Transform	2048-bit MODP Group (14)	
		Transform	256-bit random ECP group (19)	
KE Payload			DH Group	256-bit random ECP group (19)
			Key Exchange Data	(According to DH Group 19)
Nonce Payload			*According to Common Configuration*	

PACKET 2 - IKE_SA_INIT REQUEST

Step	Action	Expected Result
1.	TN1 transmits an IKE_SA_INIT Request according to Packet 2 - IKE_SA_INIT Request above.	The NUT transmits a valid IKE_SA_INIT Response containing only a Notify Payload of Type INVALID_KE_PAYLOAD (17) with a data field equal to 14 (2048-bit MODP Group). The rSPI Field is 0.
2.	TN1 transmits an IKE_SA_INIT Request according to Packet 2 - IKE_SA_INIT Request above, however the KE Payload has been modified to use DH Group 14 according to the Common Configuration.	The NUT transmits a valid IKE_SA_INIT Response.

Possible Problems:

- None.

IPsec.Conf.1.2.1.8: IKE_SA_INIT Forward Compatibility

Purpose:

To verify forward compatibility using the reserved and version fields.

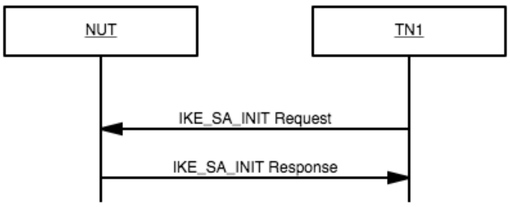
References:

- [RFC 7296] 2.5, 3.1

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Part A: Non-zero Reserved Bits

Step	Action	Expected Result
1.	TN1 transmits an IKE_SA_INIT Request. The reserved bits in the IKE Header are set to 1.	The NUT transmits a valid IKE_SA_INIT Response.

Part B: Version Bit Set

Step	Action	Expected Result
2.	TN1 transmits an IKE_SA_INIT Request. The version bit in the Flags field is set.	The NUT transmits a valid IKE_SA_INIT Response.

Possible Problems:

- None.

IPsec.Conf.1.2.1.9: IKE_SA_INIT Invalid

Purpose:

To verify an Invalid IKE_SA_INIT is ignored.

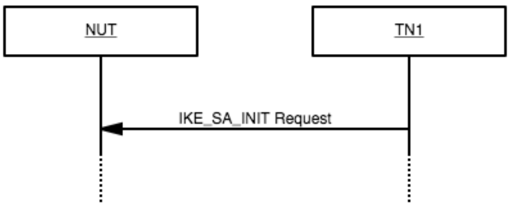
References:

- [RFC 7296] 2.21

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	TN1 transmits an IKE_SA_INIT Request. The Response bit is set to 1.	The NUT does not transmit an IKE_SA_INIT Response.

Possible Problems:

- None.

1.2.2. IKE_AUTH Exchange

IPsec.Conf.1.2.2.1: IKE_AUTH Response Format

Purpose:

To verify a properly formatted IKE_AUTH Response

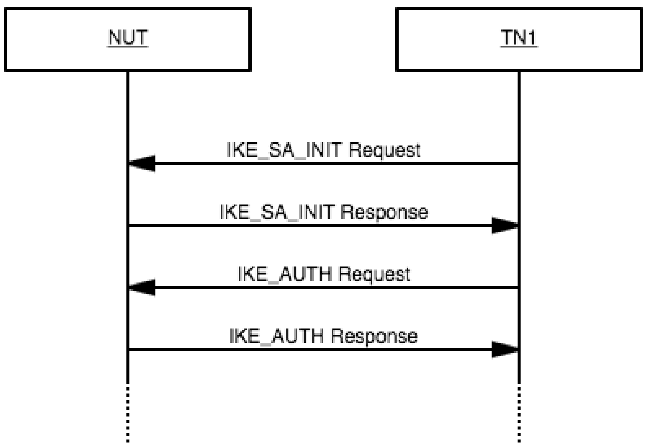
References:

- [RFC 7296] 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13, 3.14

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	TN1 transmits an IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits an IKE_AUTH Request.	The NUT transmits a valid IKE_AUTH Response. Verify fields according to **Table A** (Encrypted) and **Table B** (Decrypted Payloads) below. The accepted SA and Traffic Selectors must align with those proposed by TN1.

Table A:

Payload	Field	Value
IKE Header	iSPI	Non-Zero, equal to iSPI in IKE_SA_INIT Request and IKE_AUTH Request.
	rSPI	Non-Zero (From IKE_SA_INIT Response)
	Next Payload	Encrypted and Authenticated (46)
	Major Version	2
	Minor Version	0
	Exchange Type	IKE_AUTH (35)
	Flags	(00100000)2 = (20)16
	Message ID	1
Encrypted Payload	Initialization Vector	Valid
	Encrypted IKE Payloads	Valid
	Padding/Pad Length	Valid
	Integrity Checksum Data	Valid

Table B (Payloads within Encrypted IKE Payload):

Payload			Field	Value
ID Payload			ID Type	ID_IPV6_ADDR (5)
			ID Data	Valid
Authentication Payload			Authentication Method	Shared Key Message Integrity Code (2)
			Authentication Data	Valid
Notify Payload			Payload Length	8
			Protocol ID	0
			SPI Size	0
			Notify Message Type	USE_TRANSPORT_MODE (16391)
SA Payload	Proposal		Last	0 or 2
			Proposal #	1
			Protocol ID	ESP (3)
			SPI Size	4
			SPI	Valid
			# Transforms	3
			Last	0 or 3
			Transform Type	ENCR (1)
			Transform ID	(According to Common Configuration)
			Last	0 or 3
			Transform Type	INTEG (3)

			Transform ID	(According to Common Configuration)
			Last	0 or 3
			Transform Type	ESN (5)
			Transform ID	(According to Common Configuration)
TSi			# Traffic Selectors	1 or 2
	Traffic Selector		TS Type	TS_IPV6_ADDR_RANGE (8)
			IP Protocol ID	0
			Selector Length	40
			Start Port	0
			End Port	65535
			Starting Address	NUT IPv6 Address
			Ending Address	NUT IPv6 Address
TSr			# Traffic Selectors	1 or 2
	Traffic Selector		TS Type	TS_IPV6_ADDR_RANGE (8)
			IP Protocol ID	0
			Selector Length	40
			Start Port	0
			End Port	65535
			Starting Address	TN1 IPv6 Address
			Ending Address	TN1 IPv6 Address

Possible Problems:

- The IKE_AUTH Request may have additional payloads not described above and can be ignored. The payloads may be in any order.
- SA Payload Proposal Transforms may be in any order

IPsec.Conf.1.2.2.2: IKE_AUTH Exchange Succeeds

Purpose:

To verify a IKE_AUTH Exchange completed successfully under normal conditions.

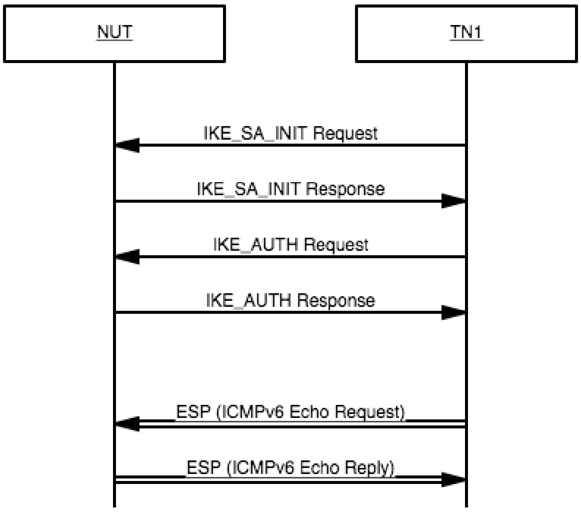
References:

- [RFC 7296] 1.2

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request.	The NUT transmits a valid IKE_AUTH Response.
3.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.2.2.3: IKE_AUTH Retransmission

Purpose:

To verify correct retransmission of IKE_AUTH Responses.

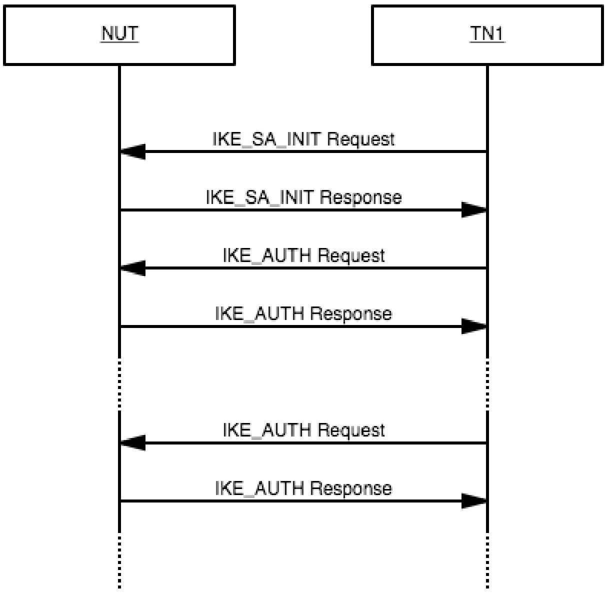
References:

- [RFC 7296] 2.1, 2.2, 2.4

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request.	The NUT transmits a valid IKE_AUTH Response.
3.	Wait 10 seconds.	The NUT does not transmit any IKEv2 packets for the newly initiated session.
4.	TN1 retransmits the IKE_AUTH Request from Step 2.	The NUT transmits a valid IKE_AUTH Response that is bitwise identical to the one transmitted in Step 2.

Possible Problems:

- None.

IPsec.Conf.1.2.2.4: State Synchronization

Purpose:

To verify IKEv2 state is not lost due to cryptographically unprotected messages.

References:

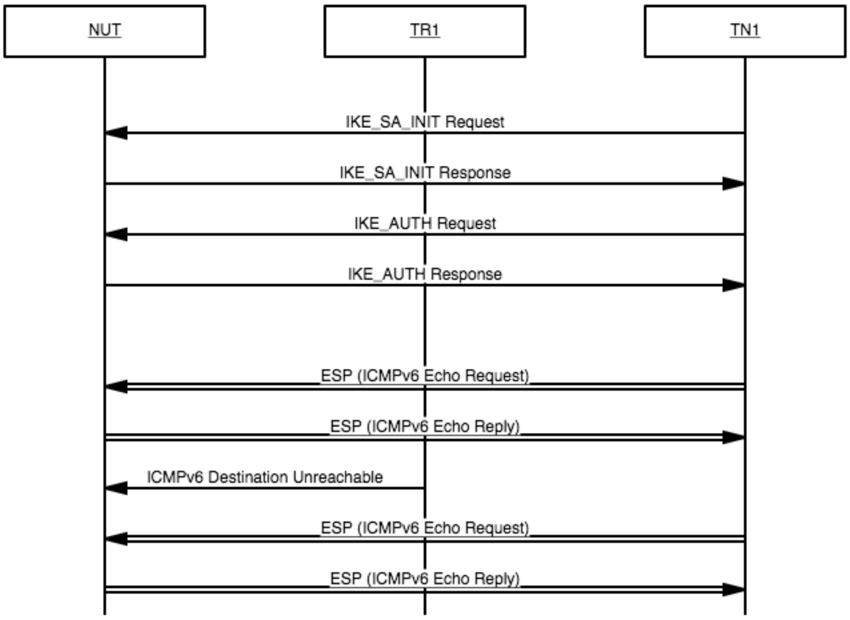
- [RFC 7296] 2.4

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration

Procedure:

Part A: ICMPv6

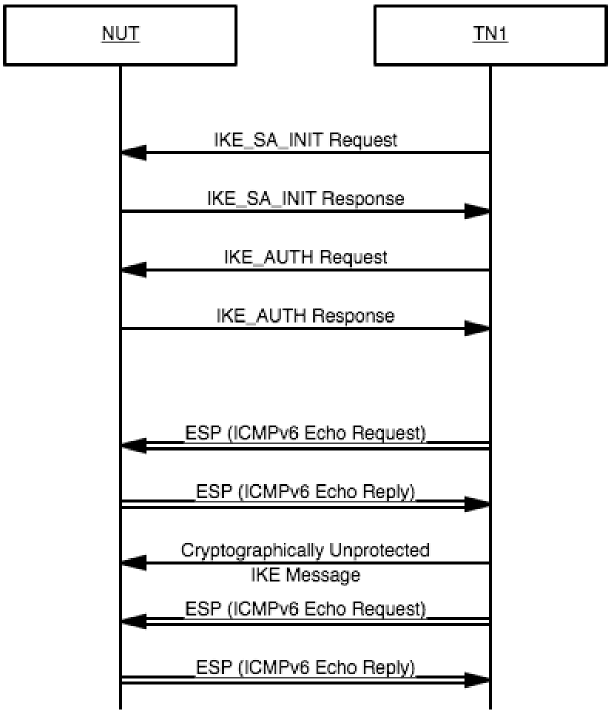


Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request.	The NUT transmits a valid IKE_AUTH Response.
3.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.
4.	TR1 transmits an ICMPv6 Destination Unreachable Message to the NUT.	
5.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Part B: IKE

Payload	Field	Value
IKE Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	41 (N)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	any
	Length	any
Notify Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	3 (ESP)
	SPI Size	0
	Notify Message Type	11 (INVALID_SPI)

PACKET 3 - CRYPTOGRAPHICALLY UNPROTECTED IKE MESSAGE



Step	Action	Expected Result
6.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
7.	TN1 transmits a valid IKE_AUTH Request.	The NUT transmits a valid IKE_AUTH Response.
8.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.
9.	TN1 transmits a cryptographically unprotected IKE Message	
10.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.2.2.5: IKE_AUTH Cryptographic Algorithm Negotiation

Purpose:

To verify algorithm negotiation during IKE_AUTH for ESP CHILD_SA

References:

- [RFC 7296] 2.7, 3.3.2, 3.3.5
- [RFC 8221]

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration
    - Use the following Transforms corresponding to each part:

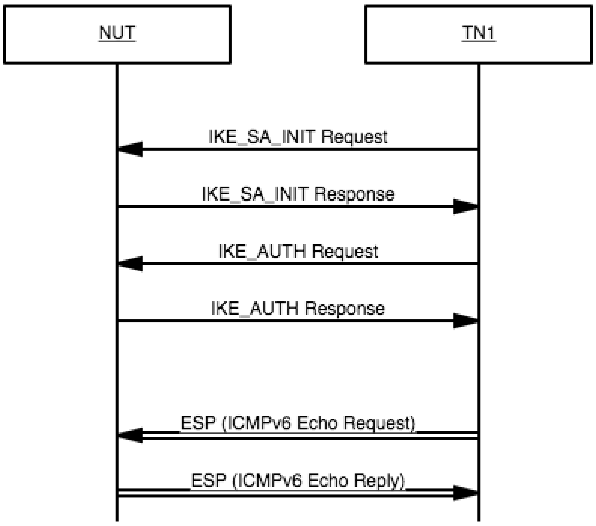
Common Configuration:

Type	Transform
ENCR (1)	ENCR_AES_CBC 128-bit (12)
INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
ESN (5)	No ESN (0)

The device should send the transforms specified in the Common Configuration. The table below substitutes the given Transform Type and Transform ID according to the part specified.

Part	Transform Type	Transform ID
A	ENCR (1)	ENCR_AES_CBC 128-bit (12)
	INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
B - AES256	ENCR (1)	ENCR_AES_CBC 256-bit (12)
	INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
C - CHACHA	ENCR (1)	ENCR_CHACHA20_POLY1305 (28)
	INTEG (3)	Omitted or NONE (0)
D - AESGCM	ENCR (1)	ENCR_AES_GCM_16 128-bit (20)
	INTEG (3)	Omitted or NONE (0)
E - AESCCM	ENCR (1)	ENCR_AES_CCM_8 128-bit (14)
	INTEG (3)	Omitted or NONE (0)
F -NULL	ENCR (1)	ENCR_NULL (11)
	INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
G - SHA512	ENCR (1)	ENCR_NULL (11)
	INTEG (3)	AUTH_HMAC_SHA2_512_256 (14)
H - AESXCBC	ENCR (1)	ENCR_NULL (11)
	INTEG (3)	AUTH_AES_XCBC_96 (5)

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request with algorithms according to the table above.	The NUT transmits a valid IKE_AUTH Response with algorithms according to the table above.
3.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.2.2.6: IKE_AUTH Multiple Transforms

Purpose:

To verify correct processing of an SA Proposal with Multiple Transforms of a single type.

References:

- [RFC 7296] 2.7, 3.3, 3.3.2, 3.3.5

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - Use the following Transforms corresponding to each part:

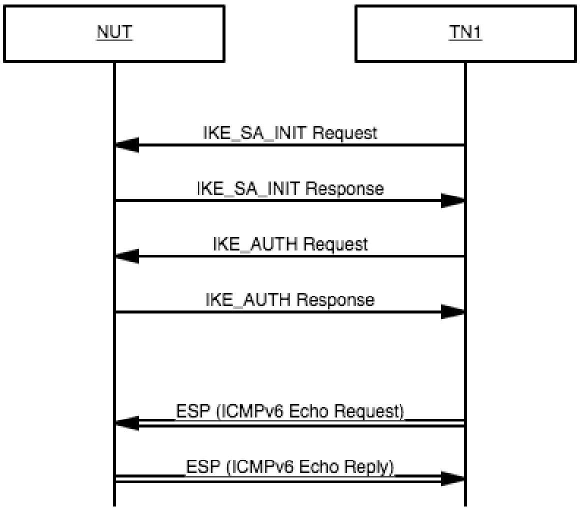
Common Configuration:

Type	Transform
ENCR (1)	ENCR_AES_CBC 128-bit (12)
INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
ESN (5)	No ESN (0)

The device should send the transforms specified in the Common Configuration. The table below adds the given Transform Type and Transform ID to the proposal according to the part specified, so that 4 transforms are proposed, with two of the same type.

Part	Transform Type	Transform ID
A	ENCR (1)	ENCR_AES_CBC 256-bit (12)
B	INTEG (3)	AUTH_HMAC_SHA2_512_256 (14)
C	ESN (5)	Yes ESN (1)

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request encrypted with the algorithms according to the table above.	The NUT transmits a valid IKE_AUTH Response encrypted with the algorithms according to the table above.
3.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.2.2.7: IKE_AUTH Multiple Proposals

Purpose:

To verify correct processing of an SA Proposal with Multiple Proposals type.

References:

- [RFC 7296] 2.7, 3.3, 3.3.2, 3.3.5

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - Use the following Transforms corresponding to each part:

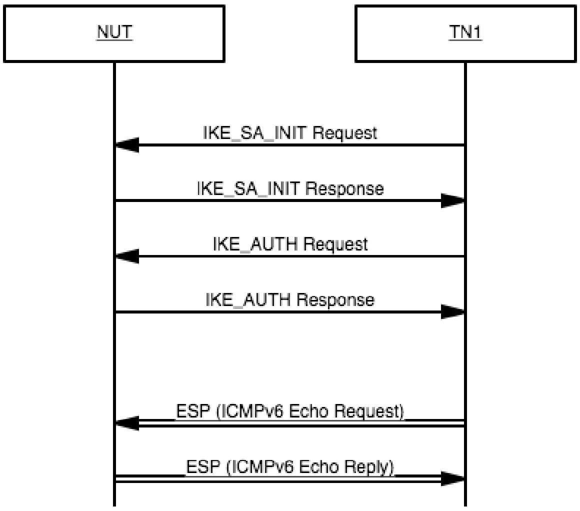
Common Configuration:

Type	Transform
ENCR (1)	ENCR_AES_CBC 128-bit (12)
INTEG (3)	AUTH_HMAC_SHA2_256_128 (12)
ESN (5)	No ESN (0)

The device should send the transforms specified in the Common Configuration. Additionally, it should send a second proposal according to the table below, for a total of 2 SA Proposals.

Type	Transform
ENCR (1)	ENCR_AES_CBC 256-bit (12)
INTEG (3)	AUTH_HMAC_SHA2_512_256 (14)
ESN (5)	Yes ESN (1)

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request with 2 SA Proposal and algorithms according to the table above.	The NUT transmits a valid IKE_AUTH Response with algorithms according to the table above.
3.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.2.2.8: IKE_AUTH N(NO_PROPOSAL_CHOSEN)

Purpose:

To verify an IKE_SA remains setup after transmission of N(NO_PROPOSAL_CHOSEN) in IKE_AUTH.

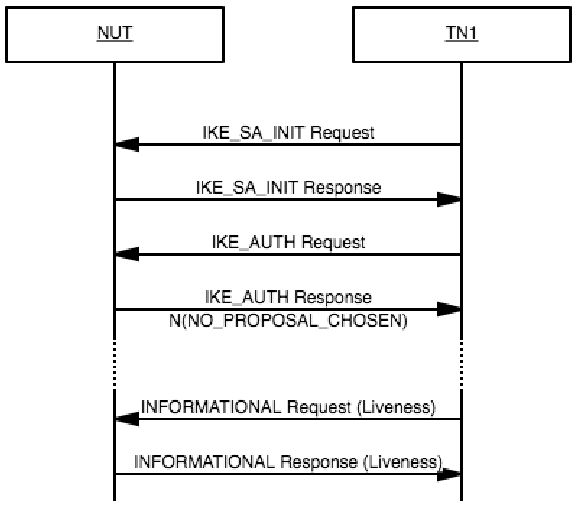
References:

- [RFC 7296] 1.2, 2.7, 2.21.2, 3.10.1

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request. The SA Proposal does not match the Common Configuration.	The NUT transmits a valid IKE_AUTH Response, with a Notify Payload of type NO_PROPOSAL_CHOSEN (14).
3.	Wait 10 seconds.	
4.	TN1 transmits an INFORMATIONAL Request with no payloads except for an empty Encrypted Payload.	The NUT transmits an INFORMATIONAL Response with no payloads except for an empty Encrypted Payload.

Possible Problems:

- None.

IPsec.Conf.1.2.2.9: Traffic Selector Negotiation

Purpose:

To verify a device is able to transmit an IKE_AUTH Response with Traffic Selectors configured to be narrower than was originally proposed.

References:

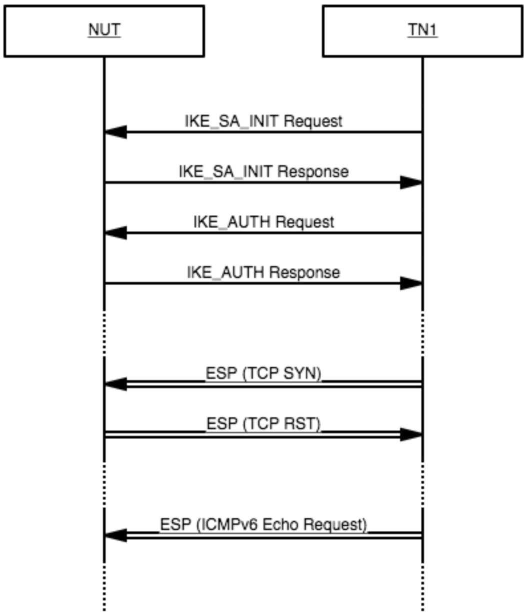
- [RFC 7296] 2.9

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - Additionally, the NUT is configured with traffic selectors for **TCP** according to the table below.

NUT Traffic Selectors	
Remote Traffic Selector	TN1_Network1
Local Traffic Selector	NUT_Network0
Protocol/Port	**TCP**/ANY

Procedure:



Part A: Narrowing from a single Traffic Selector Proposal

Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request according to the Common Configuration.	The NUT transmits a valid IKE_AUTH Response indicating TCP Traffic Selectors according to the table above. The Traffic Selector Payloads specify an IP Protocol ID of TCP (6), for TSi and TSr.
3.	TN1 transmits a TCP-SYN packet with IPsec ESP to a closed port on the NUT.	The NUT transmits a valid ESP TCP RST to TN1.
4.	TN1 transmits an ESP ICMPv6 Echo Request using the negotiated Security Association, ignoring the negotiated Traffic Selector Policy.	The NUT does not transmit an ESP ICMPv6 Echo Reply, nor does it transmit a non-encrypted ICMPv6 Echo Reply.

Part B: Narrowing from multiple Traffic Selector Proposals

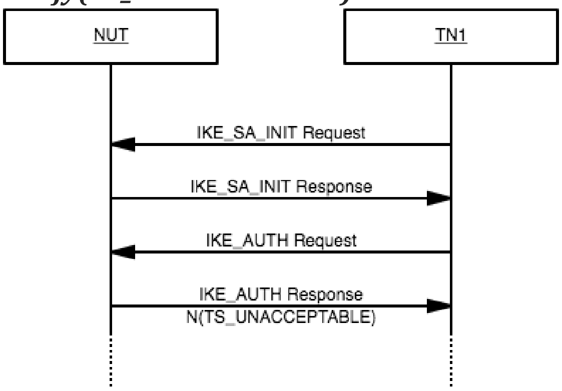
		# Traffic Selectors	**2**
TSi	Traffic Selector	TS Type	TS_IPV6_ADDR_RANGE (8)
		IP Protocol ID	**6 (TCP)**
		Selector Length	40
		Start Port	*Unassigned Local Port*
		End Port	*Unassigned Local Port*
		Starting Address	TN1 IPv6 Address
		Ending Address	TN1 IPv6 Address
	Traffic Selector	TS Type	TS_IPV6_ADDR_RANGE (8)
		IP Protocol ID	0
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1 IPv6 Address
		Ending Address	TN1 IPv6 Address
TSr		**# Traffic Selectors**	**2**
	Traffic Selector	TS Type	TS_IPV6_ADDR_RANGE (8)
		IP Protocol ID	**6 (TCP)**
		Selector Length	40
		Start Port	*Unassigned Remote Port*
		End Port	*Unassigned Remote Port*
		Starting Address	NUT IPv6 Address
		Ending Address	NUT IPv6 Address
	Traffic Selector	TS Type	TS_IPV6_ADDR_RANGE (8)
		IP Protocol ID	0
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT IPv6 Address
		Ending Address	NUT IPv6 Address

PACKET 4 - IKE_AUTH WITH MULTIPLE TRAFFIC SELECTORS

Step	Action	Expected Result
5.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
6.	TN1 transmits a valid IKE_AUTH Request according to Packet 4 - IKE_AUTH with Multiple Traffic Selectors above.	The NUT transmits a valid IKE_AUTH Response indicating TCP Traffic Selectors according to the table above. The Traffic Selector Payloads specify an IP Protocol ID of TCP (6), for TSi and TSr.

7.	TN1 transmits a TCP-SYN packet with IPsec ESP to a closed port on the NUT.	The NUT transmits a valid ESP TCP RST to TN1.
8.	TN1 transmits an ESP ICMPv6 Echo Request using the negotiated Security Association, ignoring the negotiated Traffic Selector Policy.	The NUT does not transmit an ESP ICMPv6 Echo Reply, nor does it transmit a non-encrypted ICMPv6 Echo Reply.

Part C: Transmitting Notify(TS_UNACCEPTABLE)



TSi		**# Traffic Selectors**	**1**
	Traffic Selector	TS Type	TS_IPV6_ADDR_RANGE (8)
		IP Protocol ID	**17 (UDP)**
		Selector Length	40
		Start Port	*Unassigned Local Port*
		End Port	*Unassigned Local Port*
		Starting Address	TN1 IPv6 Address
		Ending Address	TN1 IPv6 Address
TSr		**# Traffic Selectors**	**1**
	Traffic Selector	TS Type	TS_IPV6_ADDR_RANGE (8)
		IP Protocol ID	**17 (UDP)**
		Selector Length	40
		Start Port	*Unassigned Remote Port*
		End Port	*Unassigned Remote Port*
		Starting Address	NUT IPv6 Address
		Ending Address	NUT IPv6 Address

PACKET 5 - IKE_AUTH WITH UDP TRAFFIC SELECTOR

Step	Action	Expected Result
9.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.

10.	TN1 transmits a valid IKE_AUTH Request according to Packet 5 - IKE_AUTH with UDP Traffic Selector above.	The NUT transmits a valid IKE_AUTH Response with a Notify Payload of type TS_UNACCEPTABLE (38).

Possible Problems:

- The NUT may transmit a CREATE_CHILD_SA Request with Traffic Selectors matching ICMPv6 Echo Reply. This does not indicate a failure.
- A Security Gateway device may have additional Traffic Selectors, or Traffic Selectors representing a range of addresses. This should not be considered a failure.

IPsec.Conf.1.2.2.10: Peer Identification

Purpose:

To verify authentication using different Identification Types.

References:

- [RFC 7296] Sections 2.15, 3.5

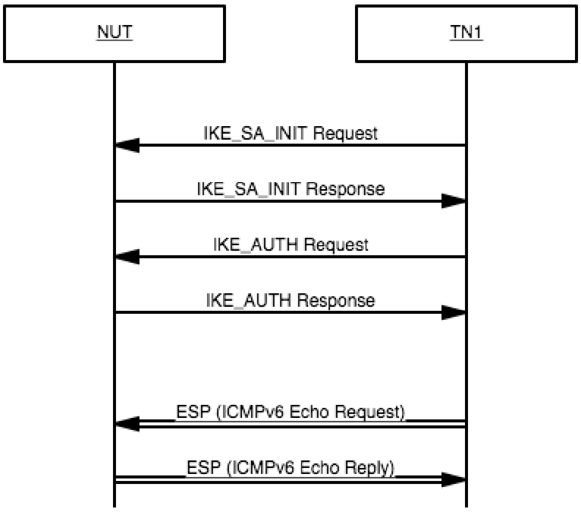
Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration
  - Configure the devices to authenticate using the Identification Types according to each part specified in the table below.

TABLE 4 - IDENTIFICATION TYPES

Part	NUT ID Type	TN1 ID Type
A	ID_IPV6_ADDR	ID_IPV6_ADDR
B	ID_IPV6_ADDR	ID_FQDN
C	ID_IPV6_ADDR	ID_RFC822_ADDR
D	ID_FQDN	ID_IPV6_ADDR
E	ID_FQDN	ID_FQDN
F	ID_FQDN	ID_RFC822_ADDR
G	ID_RFC822_ADDR	ID_IPV6_ADDR
H	ID_RFC822_ADDR	ID_FQDN
I	ID_RFC822_ADDR	ID_RFC822_ADDR

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request. The IDi Payload contains an ID Type according to the part in the TN1 ID Type column in the table above (Table 4 - Identification Types).	The NUT transmits a valid IKE_AUTH Response. The IDr Payload contains an ID Type according to the part in the NUT ID Type column in the table above (Table 4 - Identification Types)
3.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.2.2.11: Authentication via RSA Digital Signature

Purpose:

To verify authentication of a peer via RSA Digital Signature

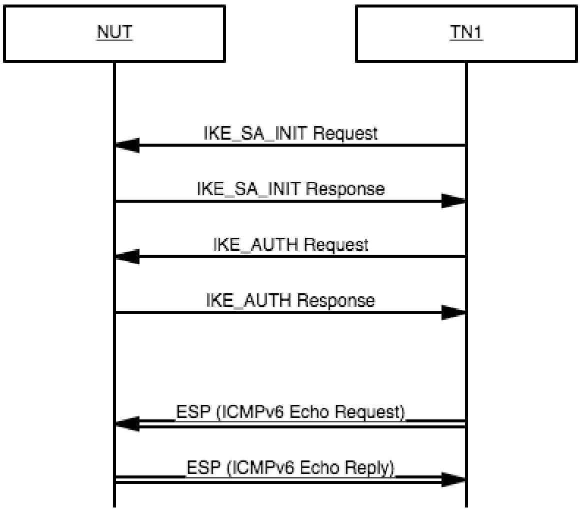
References:

- [RFC 7296] Sections 2.15, 3.5, 3.6, 3.7, 3.8

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration
    - Configure the devices to use the RSA Digital Signature (1) Authentication Method for both DUT and TN1, in place of Shared Key (3).

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response. If the Response contains a CERTREQ Payload, it is valid and formatted properly.
2.	TN1 transmits a valid IKE_AUTH Request. The Request contains a CERTREQ Payload specifying X.509 Certificate - Signature (4) and the data corresponding to the preferred CA.	The NUT transmits a valid IKE_AUTH Response. The Authentication Payload specifies an Auth Method of RSA Digital Signature (1), and contains valid authentication data. If the Response contains a CERT Payload, it is valid and formatted properly.
3.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.2.2.12: Authentication via PSK

Purpose:

To verify authentication of a peer via Shared Key Message Integrity Code

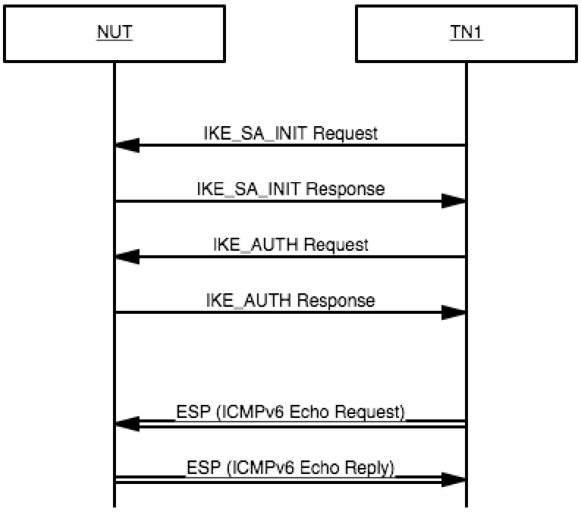
References:

- [RFC 7296] Sections 2.15

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration

Procedure:



Part A: Authentication Succeeds

Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request. The Authentication Payload specifies an Auth Method of Shared Key Message Integrity Code (2), and contains valid authentication data.	The NUT transmits a valid IKE_AUTH Response. The Authentication Payload specifies an Auth Method of Shared Key Message Integrity Code (2), and contains valid authentication data.
3.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Part B: Authentication with Hex Encoding of PSK

TABLE 5 - HEX PSK

PSK	NUT and TN1
Local & Remote	0xabadcafeabadcafeabadcafeabadcafe

Step	Action	Expected Result
4.	Initialize the NUT. Use the HEX PSK specified in the table above (Table 5 - Hex PSK).	
5.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
6.	TN1 transmits a valid IKE_AUTH Request. The Authentication Payload specifies an Auth Method of Shared Key Message Integrity Code (2), and contains valid authentication data.	The NUT transmits a valid IKE_AUTH Response. The Authentication Payload specifies an Auth Method of Shared Key Message Integrity Code (2), and contains valid authentication data.
7.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Part C: Authentication with PSK fails

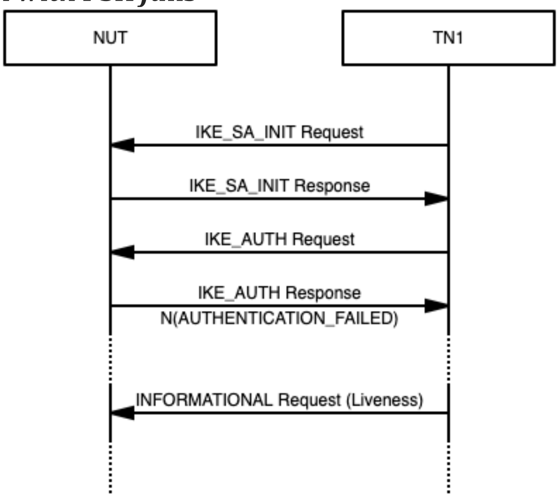


TABLE 6 - MISMATCHED PSK

PSK	NUT	TN1
Local & Remote	"IKETEST-1234"	"NOMATCH"

Step	Action	Expected Result
8.	Initialize the NUT. Use the HEX PSK specified in the table above. (Table 6 - Mismatched PSK)	
9.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
10.	TN1 transmits a valid IKE_AUTH Request. The Authentication Payload specifies an Auth Method of Shared Key Message Integrity Code (2), and contains valid authentication data.	The NUT transmits a valid IKE_AUTH Response. The Response contains a Notify Payload of Type AUTHENTICATION_FAILED (24).
11.	TN1 transmits an INFORMATIONAL Request liveness check with no payloads except for an empty Encrypted Payload.	The NUT does not transmit a response to the liveness check.

Possible Problems:

- **Part B:** This is a true "byte" representation of a key. This key cannot be represented via ASCII input and must be handled separately. For example: The ASCII byte representation of "abadcafeabadcafeabadcafeabadcafe" is 0x 61626164636166656162616463616665616261646361666561626164636166665, which is not equal to the HEX PSK given.
- **Part C:** The NUT MAY send an INFORMATIONAL Response without cryptographic protection with a notification of INVALID_IKE_SPI.

IPsec.Conf.1.2.2.13: IKE_AUTH Forward Compatibility

Purpose:

To verify that the contents of the IKE_AUTH Response Reserved field are ignored.

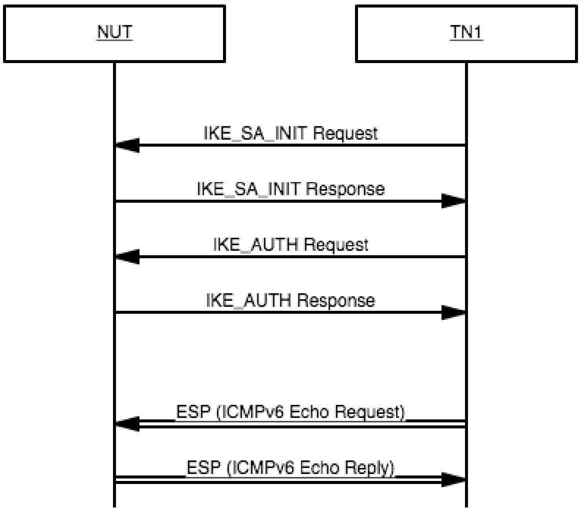
References:

- [RFC 7296] 2.5, 3.1

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request. The reserved bits in the IKE Header are set to 1.	The NUT transmits a valid IKE_AUTH Response.
3.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

IPsec.Conf.1.2.2.14: Unrecognized Notify Type

Purpose:

To verify unrecognized Notify Types are correctly processed.

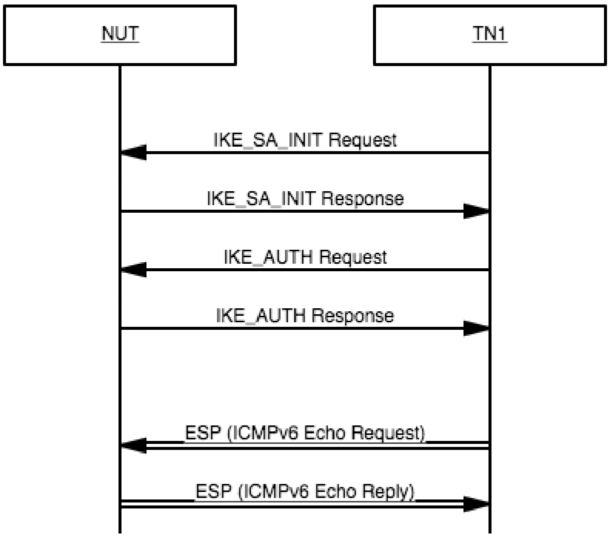
References:

- [RFC 7296] 2.5, 3.1

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Part A: Unrecognized Notify Type of Error (16383)

Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request. The Request contains a Notify Payload of Type Private Use (16383)	The NUT transmits a valid IKE_AUTH Response.
3.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Part B: Unrecognized Notify Type of Status (65535)

Step	Action	Expected Result
4.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
5.	TN1 transmits a valid IKE_AUTH Request. The Request contains a Notify Payload of Type Private Use (65535)	The NUT transmits a valid IKE_AUTH Response.
6.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.

Possible Problems:

- None.

1.2.3. IKE_AUTH Exchange - Tunnel Mode

IPsec.Conf.1.2.3.1: IKE_AUTH Response Format in Tunnel Mode

Purpose:

To verify a properly formatted IKE_AUTH Response in Tunnel Mode

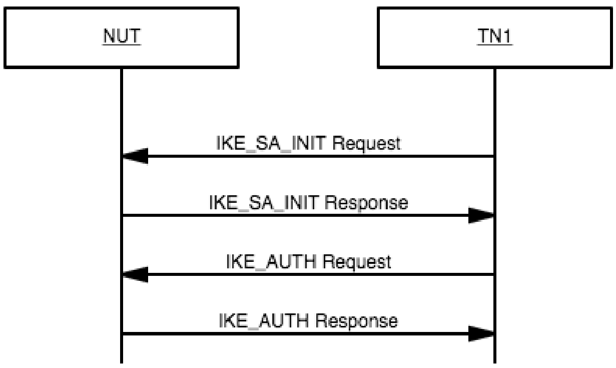
References:

- [RFC 7296] 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13, 3.14

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	TN1 transmits an IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits an IKE_AUTH Request.	The NUT transmits a valid IKE_AUTH Response. Verify fields according to Table A (Encrypted) and Table B (Decrypted Payloads) below. The NUT uses **TUNNEL Mode**, with Traffic Selectors matching **Network2.**

Table A:

Payload	Field	Value
IKE Header	iSPI	Non-Zero (From IKE_SA_INIT Request)
	rSPI	Non-Zero
	Next Payload	Encrypted and Authenticated (46)
	Major Version	2
	Minor Version	0
	Exchange Type	IKE_AUTH (35)
	Flags	(00100000)2 = (20)16
	Message ID	1
Encrypted Payload	Initialization Vector	Valid
	Encrypted IKE Payloads	Valid
	Padding/Pad Length	Valid
	Integrity Checksum Data	Valid

Table B (Payloads within Encrypted IKE Payload):

Payload			Field	Value
ID Payload			ID Type	ID_IPV6_ADDR (5)
			ID Data	Valid
Authentication Payload			Authentication Method	Shared Key Message Integrity Code (2)
			Authentication Data	Valid
SA Payload	Proposal		Last	0 or 2
			Proposal #	1
			Protocol ID	ESP (3)
			SPI Size	4
			SPI	Valid
			# Transforms	3
			Last	0 or 3
			Transform Type	ENCR (1)
			Transform ID	(According to Common Configuration)
			Last	0 or 3
			Transform Type	INTEG (3)
			Transform ID	(According to Common Configuration)
			Last	0 or 3
			Transform Type	ESN (5)
			Transform ID	(According to Common Configuration)
TSi			# Traffic Selectors	1 or 2
	Traffic Selector		TS Type	TS_IPV6_ADDR_RANGE (8)

		IP Protocol ID	0
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NETWORK2::0000
		Ending Address	NETWORK2::FFFF
TSr		# Traffic Selectors	1 or 2
	Traffic Selector	TS Type	TS_IPV6_ADDR_RANGE (8)
		IP Protocol ID	0
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT IPv6 Address
		Ending Address	NUT IPv6 Address

Possible Problems:

- The IKE_AUTH Request may have additional payloads not described above and can be ignored. The payloads may be in any order.
- There may be more than one Proposal in the SA Payload. One proposal must match the above.
- SA Payload Proposal Transforms may be in any order.

There may be more than one traffic selector in the TSi and TSr payloads. The last traffic selector must match the above.

IPsec.Conf.1.2.3.2: IKE_AUTH Exchange Succeeds in Tunnel Mode

Purpose:

To verify a IKE_AUTH Exchange completed successfully under normal conditions utilizing Tunnel Mode.

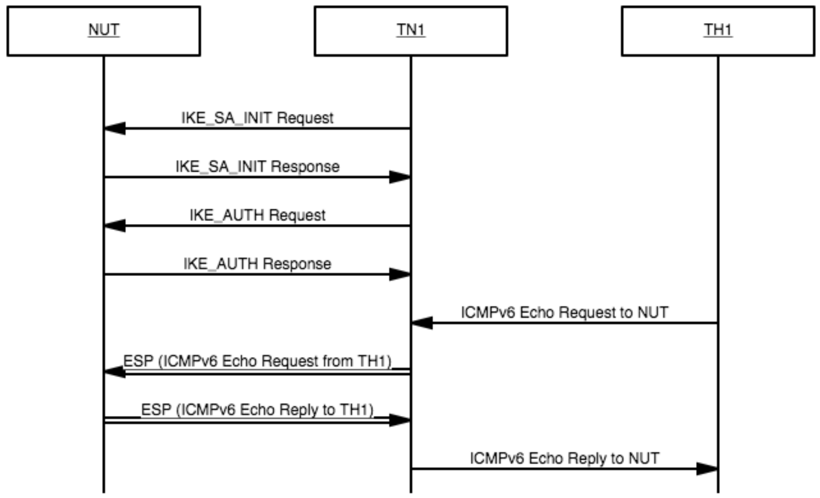
References:

- [RFC 7296] 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13, 3.14

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request.	The NUT transmits a valid IKE_AUTH Response.
3.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT transmits a valid ESP ICMPv6 Echo Reply as negotiated.
4.	TN1 transmits an ESP Tunneled ICMPv6 Echo Request as negotiated on behalf of TH1.	The NUT transmits a valid ESP Tunneled ICMPv6 Echo Reply as negotiated in response to TH1.

Possible Problems:

- None.

1.2.4. CREATE_CHILD_SA Exchange

1.2.5. INFORMATIONAL Exchange

IPsec.Conf.1.2.5.1: INFORMATIONAL Exchange

Purpose:

To verify capability to respond to Liveness Checks via empty INFORMATIONAL Request.

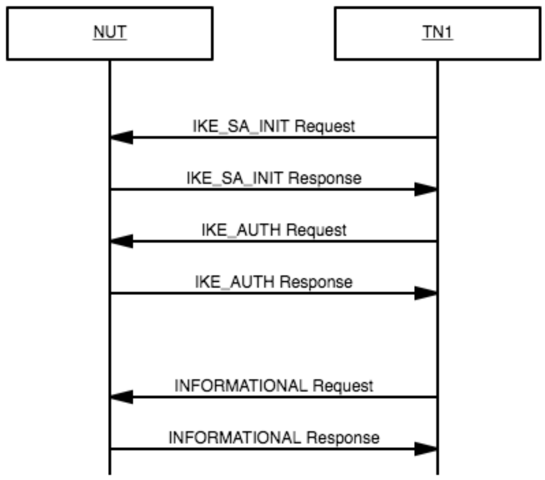
References:

- [RFC 7296] 1.4.1, 2.4

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Part A: Liveness Check

Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request.	The NUT transmits a valid IKE_AUTH Response.
3.	Wait 10 seconds.	
4.	TN1 transmits an INFORMATIONAL Request with no payloads except for an empty Encrypted Payload.	The NUT transmits an INFORMATIONAL Response with no payloads except for an empty Encrypted Payload.

Part B: Retransmission

Step	Action	Expected Result
5.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
6.	TN1 transmits a valid IKE_AUTH Request.	The NUT transmits a valid IKE_AUTH Response.
7.	Wait 10 seconds.	
8.	TN1 transmits an INFORMATIONAL Request with no payloads except for an empty Encrypted Payload.	The NUT transmits an INFORMATIONAL Response with no payloads except for an empty Encrypted Payload.
9.	TN1 retransmits the INFORMATIONAL Request from Step 8.	The NUT transmits a valid INFORMATIONAL Response that is bitwise identical to the one transmitted in Step 8.

Part C: Non-Zero Reserved Fields

Step	Action	Expected Result
10.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
11.	TN1 transmits a valid IKE_AUTH Request.	The NUT transmits a valid IKE_AUTH Response.
12.	Wait 10 seconds.	
13.	TN1 transmits an INFORMATIONAL Request with no payloads except for an empty Encrypted Payload. All Reserved fields in the message are set to 1.	The NUT transmits an INFORMATIONAL Response with no payloads except for an empty Encrypted Payload.

Possible Problems:

- None.

IPsec.Conf.1.2.5.2: IKE_SA Deletion

Purpose:

To verify a device correctly processes and responds to an INFORMATIONAL Request to delete an IKE_SA.

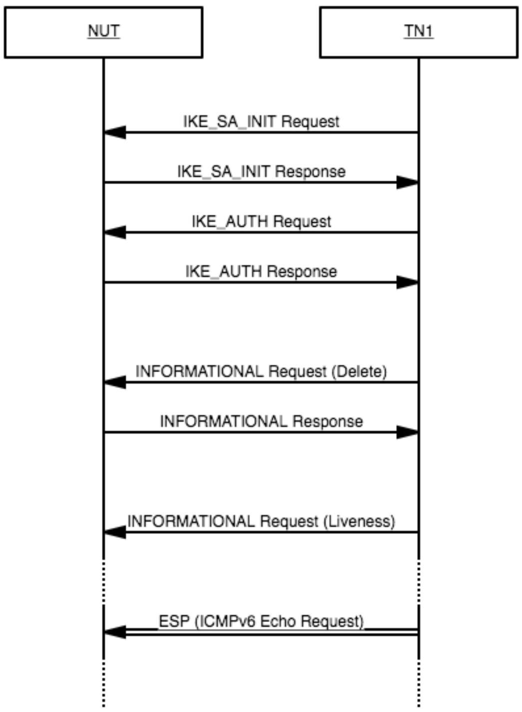
References:

- [RFC 7296] 1.4.1, 2.4

Initialization:

- Network Topology
  - Connect the devices according to Common Topology
- Configuration
  - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request.	The NUT transmits a valid IKE_AUTH Response.
3.	Wait 10 seconds.	
4.	TN1 transmits an INFORMATIONAL request with a Delete payload with a Protocol ID of 1 (IKE_SA), a SPI Size of 0, and no SPI value.	The NUT transmits an INFORMATIONAL response with no payloads except for an empty Encrypted Payload.
5.	TN1 transmits an INFORMATIONAL Request liveness check with no payloads except for an empty Encrypted Payload.	The NUT does not transmit a response to the liveness check.
6.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT does not transmit an ESP ICMPv6 Echo Reply.

Possible Problems:

- In step 5, the NUT MAY send an INFORMATIONAL Response without cryptographic protection with a notification of INVALID_IKE_SPI.

IPsec.Conf.1.2.5.3: CHILD_SA Deletion

Purpose:

To verify a device correctly processes and responds to and INFORMATIONAL Request to delete a CHILD_SA.

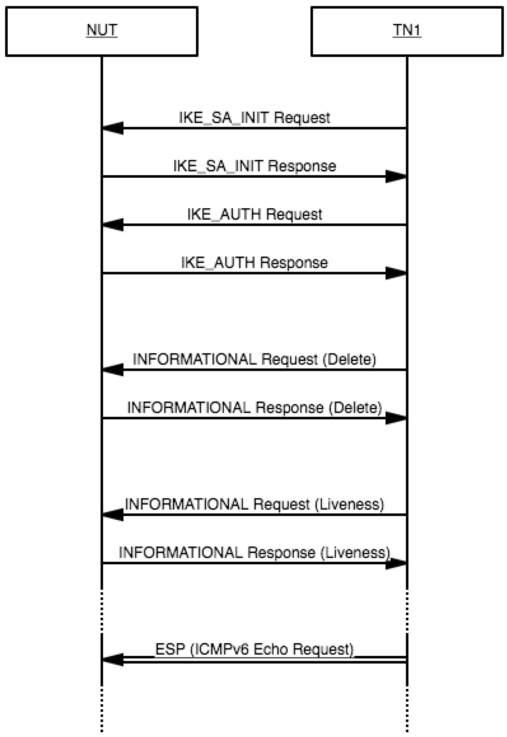
References:

- [RFC 7296] 1.4.1, 2.4

Initialization:

- Network Topology
    - Connect the devices according to Common Topology
- Configuration
    - Configure the devices according to the Common Configuration

Procedure:



Step	Action	Expected Result
1.	TN1 transmits a valid IKE_SA_INIT Request.	The NUT transmits a valid IKE_SA_INIT Response.
2.	TN1 transmits a valid IKE_AUTH Request.	The NUT transmits a valid IKE_AUTH Response.
3.	Wait 10 seconds.	
4.	TN1 transmits an INFORMATIONAL request with a Delete payload with a Protocol ID of 3 (ESP), a SPI Size of 4, and a SPI value equal to TN1's inbound ESP SPI.	The NUT transmits an INFORMATIONAL Response Delete payload with a Protocol ID of 3 (ESP), a SPI Size of 4, and a SPI value equal to the NUT's inbound ESP SPI. See Table A below.
5.	TN1 transmits an INFORMATIONAL Request liveness check with no payloads except for an empty Encrypted Payload.	The NUT transmits an INFORMATIONAL Response with no payloads except for an empty Encrypted Payload.
6.	TN1 transmits an ESP ICMPv6 Echo Request as negotiated.	The NUT does not transmit an ESP ICMPv6 Echo Reply.

Table A:

Payload	Field	Value
DELETE Payload	Protocol ID	ESP (1)
	SPI Size	4
	# SPIs	1
	SPI	CHILD_SA SPI

Possible Problems:

- None.

Section 2: IPsec End-Node

This Chapter describes the test specification for End-Node.

The test specification consists of 2 sections pertaining to IPsec Architecture, one each for Transport and Tunnel Mode.

IKEv2 Tests which are specific to End-Node IPsec Architecture may also be included.

2.1. IPsec/ESP Architecture (Transport Mode)

IPsec.Conf.2.1.1. Select SPD

Purpose:

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

References:

- [RFC 4301] 4.4.1.1
- [RFC 4303]
- [RFC 7296] 2.9

Initialization:

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

Databases:

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

Policy 1	
Peer	EN1_Link1
Mode	Transport
Remote Traffic Selector	EN1_Link1
Local Traffic Selector	NUT_Link0
Protocol/Port	ANY/ANY

Policy 2	
Peer	EN2_Link1
Mode	Transport
Remote Address	EN2_Link1
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA1-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA1-I's ESP

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA1-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA1-O's ESP

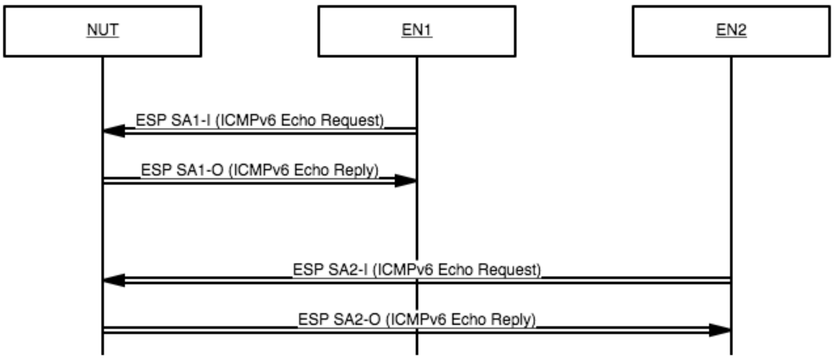
IP Header	Source Address	EN2_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic3
	Sequence	1
	Encrypted Data/ICV	SA2-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA2-I's ESP

IP Header	Source Address	NUT_Link0
	Destination Address	EN2_Link1
ESP	SPI	Dynamic4
	Sequence	1
	Encrypted Data/ICV	SA2-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA2-O's ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits ICMP Echo Request with SA1-I's ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with SA1-O's ESP
4.	EN2 transmits ICMP Echo Request with SA2-I's ESP	
5.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with SA2-O's ESP

Possible Problems:

- None

IPsec.Conf.2.1.2. Select SPD (Next Layer Protocol Selectors)

Purpose:

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

References:

- [RFC 4301] 4.4.1.1
- [RFC 4303]
- [RFC 7296] 2.9

Initialization:

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

Databases:

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

Part A: Select ICMPv6 Type

Policy 1	
Peer	EN1_Link1
Mode	Transport
Remote Address	EN1_Link1
Local Address	NUT_Link0
Protocol/Port	ICMPv6/128 (Echo Request)

Policy 2	
Peer	EN1_Link1
Mode	Transport
Remote Address	EN2_Link1
Local Address	NUT_Link0
Protocol/Port	ICMPv6/129 (Echo Reply)

Packets:

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA1-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA1-I's ESP

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA2-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA2-O's ESP

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic3
	Sequence	1
	Encrypted Data/ICV	SA1-O
ICMP	Type	128 (Echo Request)

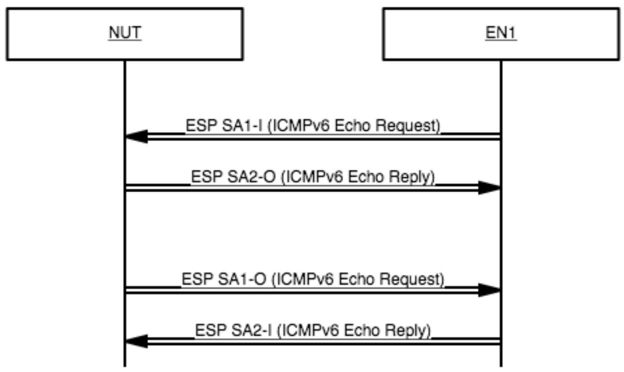
ICMP Echo Request with SA1-O's ESP

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic4
	Sequence	1

	Encrypted Data/ICV	SA2-I
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA2-I's ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits ICMP Echo Request with SA1-I's ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with SA2-O's ESP
4.	Transmit ICMP Echo Request with SA1-O's ESP from the NUT to the Global unicast address of EN1	
5.	Observe the packets transmitted on Link0	EN1 transmits ICMP Echo Reply with SA2-I's ESP

Part B: Select TCP Port

Policy 1	
Peer	EN1_Link1
Mode	Transport
Remote Address/Port	EN1_Link1/50001
Local Address/Port	NUT_Link0/55005
Protocol	TCP

Policy 2	
Peer	EN1_Link1
Mode	Transport
Remote Address/Port	EN1_Link1/60001
Local Address/Port	NUT_Link0/65005
Protocol	TCP

Packets:

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA1-I
TCP	Type	SYN
	Source Port	50001
	Destination Port	55005

TCP SYN with SA1-I's ESP

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA2-O
TCP	Type	RST
	Source Port	55005
	Destination Port	50001

TCP RST Reply with SA1-O's ESP

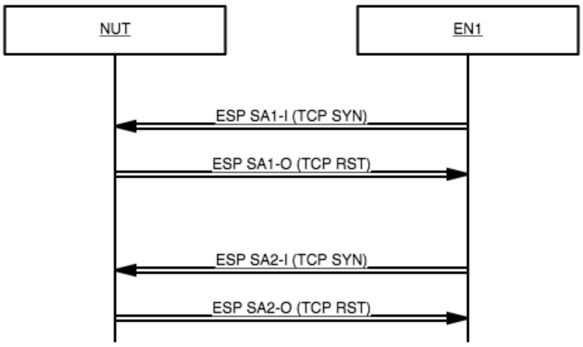
IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic3
	Sequence	1
	Encrypted Data/ICV	SA1-I
TCP	Type	SYN
	Source Port	60001
	Destination Port	65005

TCP SYN with SA1-I's ESP

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic4
	Sequence	1
	Encrypted Data/ICV	SA2-O
TCP	Type	RST
	Source Port	65005
	Destination Port	60001

TCP RST Reply with SA1-O's ESP

Procedure:



Step	Action	Expected Result
6.	Initialize the NUT	
7.	EN1 transmits TCP SYN with SA1-I's ESP	
8.	Observe the packets transmitted on Link0	The NUT transmits TCP RST with SA1-O's ESP
9.	Transmit TCP SYN with SA2-I's ESP from the NUT to the Global unicast address of EN1	
10.	Observe the packets transmitted on Link0	EN1 transmits TCP RST with SA2-O's ESP

Possible Problems:

- Possible Problem Part A: NUT may be a passive node that does not implement an application for sending Echo Requests. In this case, steps 4 and 5 may be omitted.
- Possible Problem Part B:

- Ensure the NUT has no service listening on the prescribed ports, or select alternative ports.

IPsec.Conf.2.1.3. Sequence Number Increment

Purpose:

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

References:

- [RFC 4301] 4.4.2.1
- [RFC 4303] 2.2
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	EN1_Link1
Mode	Transport
Remote Address	EN1_Link1
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1st = 1, 2nd = 2
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

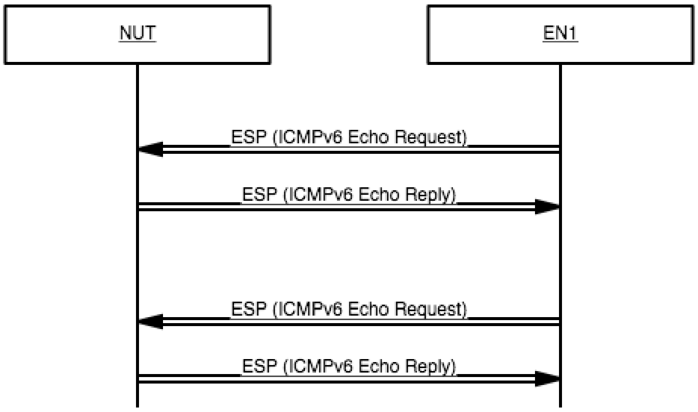
ICMP Echo Request with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic2
	Sequence	1st = 1, 2nd = 2
	Encrypted Data/ICV	SA-O

ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits an ICMP Echo Reply with ESP with an ESP Sequence Number of 1
4.	EN1 transmits ICMP Echo Request with ESP	
5.	Observe the packets transmitted on Link0	The NUT transmits an ICMP Echo Reply with ESP with an ESP Sequence Number of 2

Possible Problems:

- None

IPsec.Conf.2.1.4. Packet Too Big Reception

Purpose:

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

References:

- [RFC 4301] 7
- [RFC 4303] 3.3.4
- [RFC 7296]

Initialization:

- Topology
  - Connect the devices according to Common Topology
- Configuration
  - Use Global Security Associations
  - In addition, configure TR1_Link1 to have an MTU of 1280 bytes.

Databases:

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

Policy 1	
Peer	EN1_Link1
Mode	Transport
Remote Address	EN1_Link1
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
	Payload Length	1240
Fragment Header	Offset	0
	More	1
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP 1

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
	Payload Length	116
Fragment Header	Offset	154
	More	0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	Rest of ICMP Echo Request with ESP

Fragmented ICMP Echo Request with ESP 2

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
	Payload Length	1340
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	TR1_Link1
	Destination Address	NUT_Link0
ICMP	Type	2 (Packet Too Big)
	MTU	1280
	Data	1232 Byte of ICMP Echo Reply with ESP

ICMP Error Message (Packet Too Big)

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
	Payload Length	1240
Fragment	Offset	0
	More Flag	1

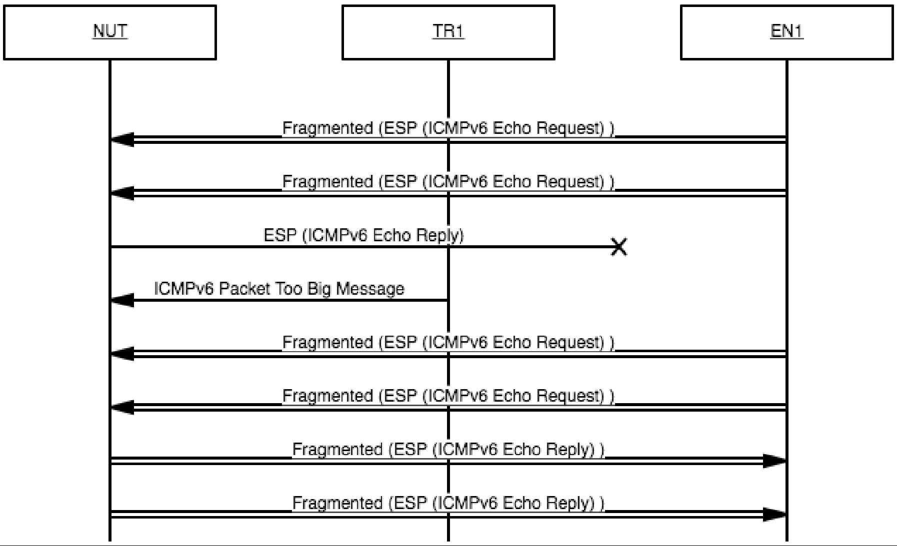
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

Fragmented ICMP Echo Reply with ESP 1

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
	Payload Length	116
Fragment	Offset	154
	More Flag	0
Data	Data	Rest of ICMP Echo Reply with ESP

Fragmented ICMP Echo Reply with ESP 2

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits Fragmented ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP
4.	TR1 transmits ICMP Error Message (Packet Too Big) to the NUT	
5.	EN1 sends Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2	
6.	Observe the packets transmitted on Link0	The NUT transmits Fragmented ICMP Echo Reply with ESP 1 and Fragmented ICMP Echo Reply with ESP 2

Possible Problems:

- None

IPsec.Conf.2.1.5. Receipt of No Next Header

Purpose:

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

References:

- [RFC 4301] 3.2
- [RFC 4303] 2.6
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	EN1_Link1
Mode	Transport
Remote Address	EN1_Link1
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA-I's ESP

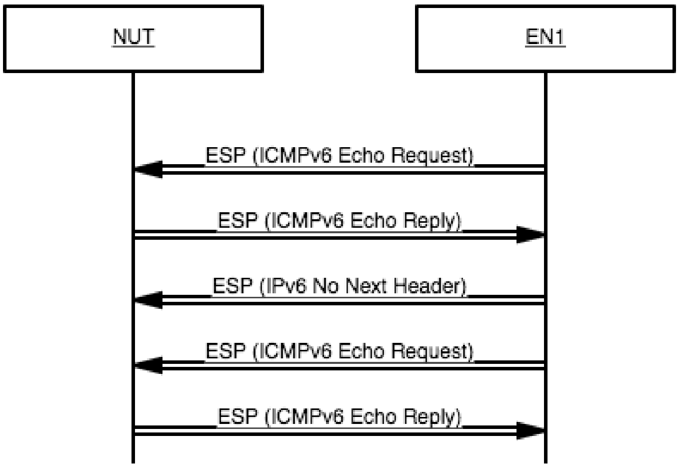
IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA-O's ESP

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
	Next Header	no next header (59)
Upper Layer	Data	empty

No Next Header with SA-I's ESP

Procedure:



Part A: No Next Header

Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits ICMP Echo Request with SA-I's ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with SA-O's ESP
4.	EN1 transmits No Next Header with SA-I's ESP (The ESP sequence number must be incremented according to the packet transmitted at step 2)	
5.	EN1 transmits ICMP Echo Request with SA-O's ESP (The ESP sequence number must be incremented according to the packet transmitted at step 4)	
6.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with SA-O's ESP

Part B: TFC Padding with No Next Header

Step	Action	Expected Result
7.	Initialize the NUT	
8.	EN1 transmits ICMP Echo Request with SA-I's ESP	
9.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with SA-O's ESP
10.	EN1 transmits No Next Header with SA-O's ESP (The ESP sequence number must be incremented according to the packet transmitted at step 2 and the data in the upper layer consists of random bytes as the plaintext portion)	
11.	EN1 transmits ICMP Echo Request with SA-O's ESP (The ESP sequence number must be incremented according to the packet transmitted at step 4)	
12.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with SA-O's ESP

Possible Problems:

- None

IPsec.Conf.2.1.6. Bypass Policy

Purpose:

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

References:

- [RFC 4301] 4.4.1
- [RFC 4303]
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	EN1_Link1
Mode	Transport
Remote Address	EN1_Link1
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Policy 2	
Peer	EN2_Link1
Mode	BYPASS
Remote Address	EN2_Link1
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
	Payload Length	1460
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
	Payload Length	1460
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

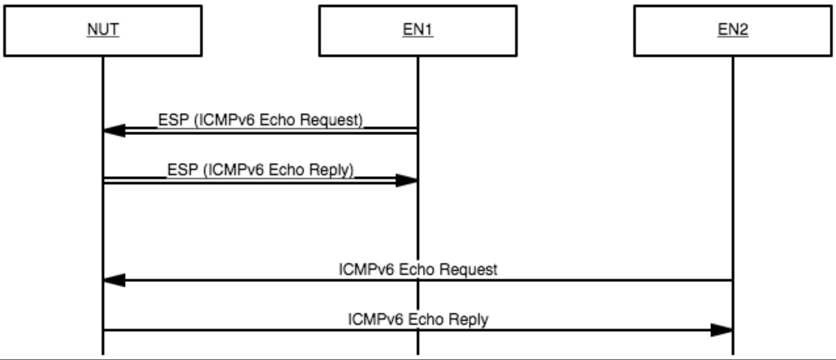
IP Header	Source Address	EN2_Link1
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	NUT_Link0
	Destination Address	EN2_Link1
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with SA-O's ESP
4.	EN2 transmits ICMP Echo Request	
5.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply

Possible Problems:

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

IPsec.Conf.2.1.7. Discard Policy

Purpose:

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

References:

- [RFC 4301] 4.4.1
- [RFC 4303]
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	EN1_Link1
Mode	Transport
Remote Address	EN1_Link1
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Policy 2	
Peer	EN2_Link1
Mode	DISCARD
Remote Address	EN1_Link1
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
	Payload Length	1460
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
	Payload Length	1460
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

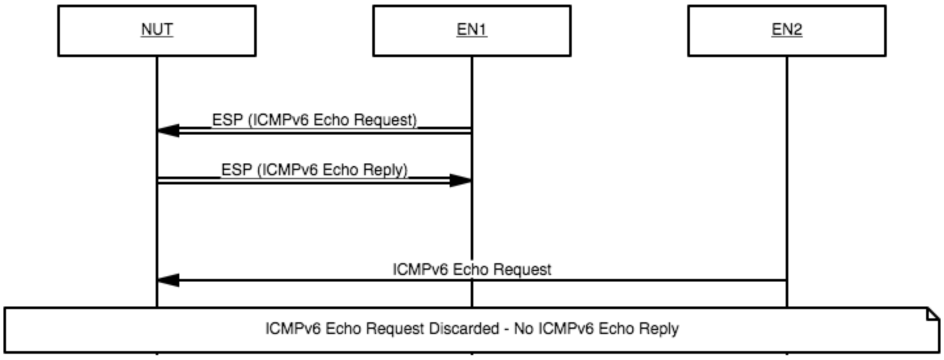
IP Header	Source Address	EN2_Link1
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	NUT_Link0
	Destination Address	EN2_Link1
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP
4.	EN2 transmits ICMP Echo Request	
5.	Observe the packets transmitted on Link0	The NUT never transmits ICMP Echo Reply

Possible Problems:

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

IPsec.Conf.2.1.8. Transport Mode Padding

Purpose:

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

References:

- [RFC 4301] 4.2
- [RFC 4303] 2.4
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	EN1_Link1
Mode	Transport
Remote Address	EN1_Link1
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Part A: Transport Mode Padding (PadLen 7)

Packets:

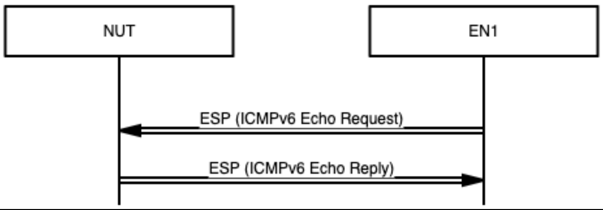
IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
	Padding	Sequential
	Padding Length	**7**
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP 1

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
	Padding Length	7+8n (0 <= n <= 31)
ICMP	Type	129 (Echo Reply)
	Data Length	7

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits ICMP Echo Request with ESP 1	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP

Part B: Transport Mode Padding (PadLen 255)
Packets:

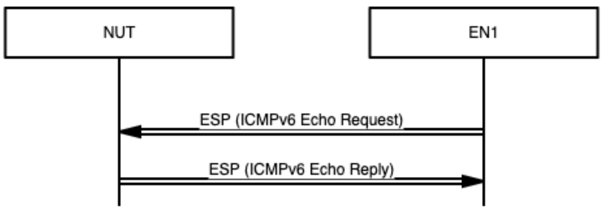
IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
	Padding	Sequential
	Padding Length	**255**
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP 2

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
	Padding Length	7+8n (0 <= n <= 31)
ICMP	Type	129 (Echo Reply)
	Data Length	7

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
4.	Initialize the NUT	
5.	EN1 transmits ICMP Echo Request with ESP 2	
6.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP

Part C: TFC enabled Transport Mode Padding
Packets:

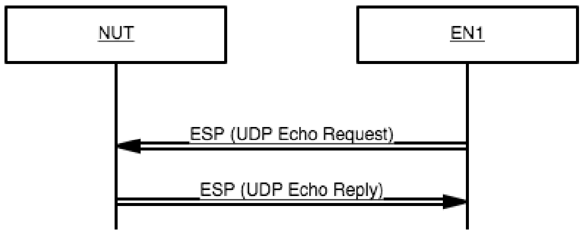
IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
UDP	Source Port	10000
	Destination Port	7 (echo)

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

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
UDP	Source Port	7 (echo)
	Destination Port	10000

UDP Echo Reply with SA-O's ESP

Procedure:



Step	Action	Expected Result
7.	Initialize the NUT	
8.	EN1 transmits UDP Echo Request with SA-I's ESP (TFC Padded)	
9.	Observe the packets transmitted on Link0	The NUT transmits UDP Echo Reply with SA-O's ESP

Possible Problems:

- None

IPsec.Conf.2.1.9. Invalid SPI

Purpose:

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

References:

- [RFC 4301] 4.4.2, 4.4.2.1
- [RFC 4303] 2.1
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	EN1_Link1
Mode	Transport
Remote Address	EN1_Link1
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP 1

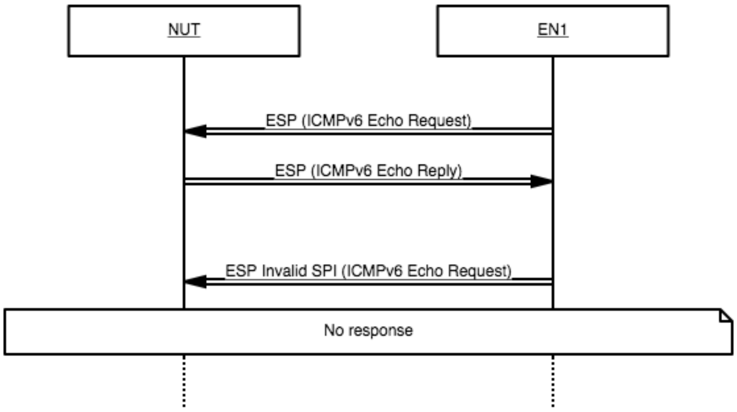
IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Different from SA-I's SPD
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

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

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits ICMP Echo Request with ESP 1	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP
4.	EN1 transmits ICMP Echo Request with ESP 2 (Non-Registered)	
5.	Observe the packets transmitted on Link0	The NUT never transmits ICMP Echo Reply with ESP

Possible Problems:

- None

IPsec.Conf.2.1.10. Invalid ICV

Purpose:

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

References:

- [RFC 4301]
- [RFC 4303] 2.8
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	EN1_Link1
Mode	Transport
Remote Address	EN1_Link1
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)
	Data	"EchoData"

ICMP Echo Request with ESP 1

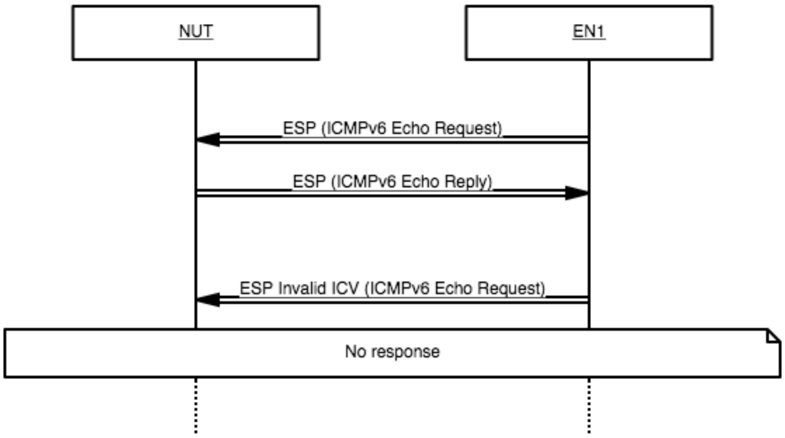
IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)
	Data	"EchoData"

ICMP Echo Reply with ESP

IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	2
	Encrypted Data/ICV	SA-I
	ICV	aaaaaaaaaaaaaaaaaa......
ICMP	Type	128 (Echo Request)
	Data	"cracked"

ICMP Echo Request with ESP 2 (ICV is modified)

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits ICMP Echo Request with ESP 1	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP
4.	EN1 transmits ICMP Echo Request with ESP 2 (ICV is modified)	
5.	Observe the packets transmitted on Link0	The NUT never transmits ICMP Echo Reply with ESP

Possible Problems:

- None

2.2. IPsec/ESP Architecture (Tunnel Mode)

IPsec.Conf.2.2.1. Tunnel Mode with SGW

Purpose:

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

References:

- [RFC 4301] 4.4.1.1
- [RFC 4303]
- [RFC 7296] 2.9

Initialization:

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

Databases

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

Policy 1	
Peer	SGW1_Link1
Mode	Tunnel
Remote Address	Link2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

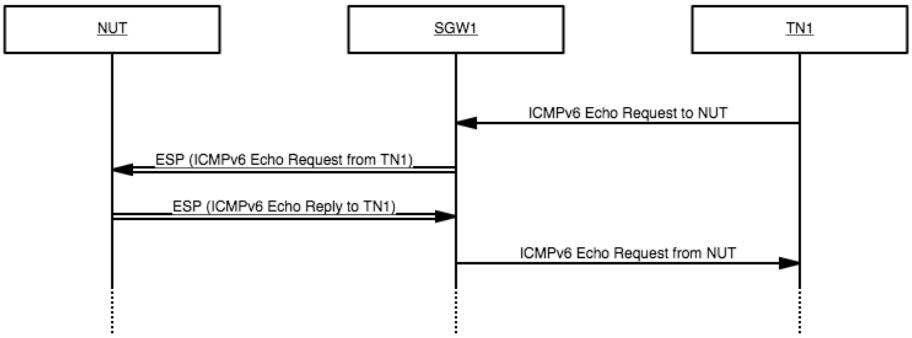
IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP

Possible Problems:

- None

IPsec.Conf.2.2.2. Tunnel Mode Select SPD

Purpose:

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

References:

- [RFC 4301] 4.4.1.1
- [RFC 4303]
- [RFC 7296] 2.9

Initialization:

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

Databases:

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

Policy 1	
Peer	SGW1_Link1
Mode	Tunnel
Remote Traffic Selector	TN1_Link2
Local Traffic Selector	NUT_Link0
Protocol/Port	ANY/ANY
Policy 2	
Peer	SGW1_Link1
Mode	Tunnel
Remote Address	TN2_Link2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP 1

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP 1

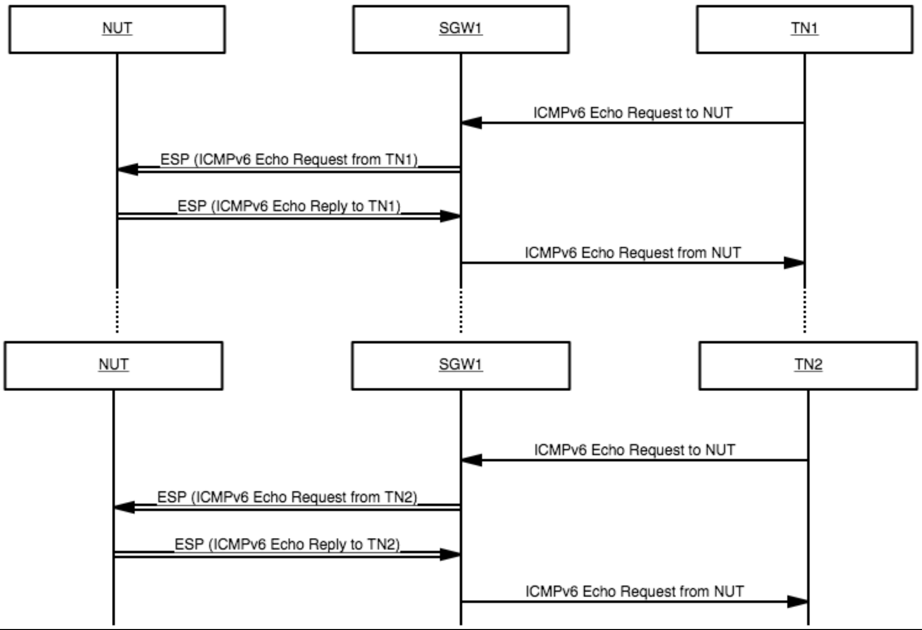
IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic3
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN2_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP 2

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic4
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN2_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP 2

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 transmits ICMP Echo Request with ESP 1	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP 1
4.	SGW1 transmits ICMP Echo Request with ESP 2	
5.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP 2

Possible Problems:

- None

IPsec.Conf.2.2.3. Tunnel Mode Sequence Number Increment

Purpose:

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

References:

- [RFC 4301] 4.4.2.1
- [RFC 4303] 2.2
- [RFC 7296]

Initialization:

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

Databases

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

Policy 1	
Peer	SGW1_Link1
Mode	Tunnel
Remote Address	Link2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

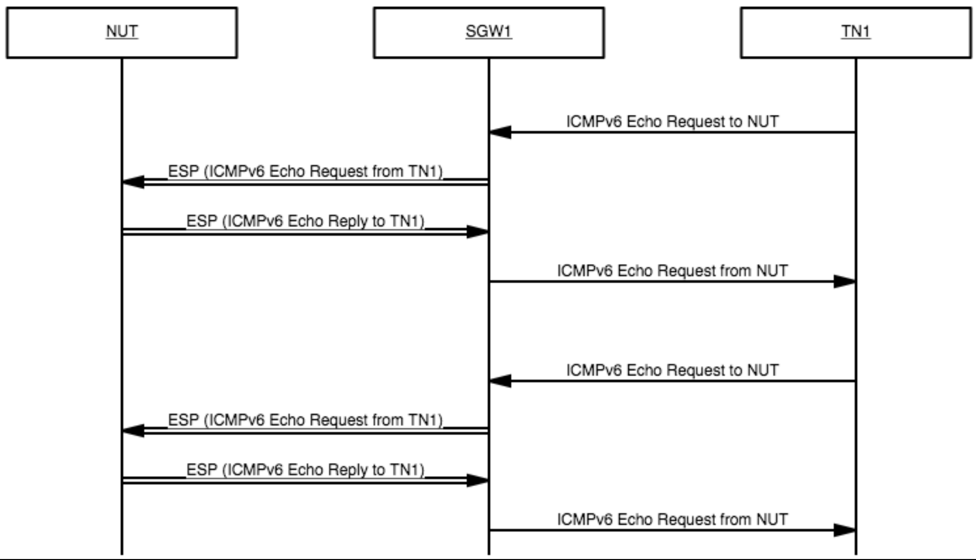
IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 transmits ICMP Echo Request with ESP	

3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP with an ESP Sequence Number of 1.
4.	SGW1 transmits ICMP Echo Request with ESP	
5.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP with an ESP Sequence Number of 2.

Possible Problems:

- None

IPsec.Conf.2.2.4. Tunnel Mode Packet Too Big Reception

Purpose:

Verify that a NUT (End-Node) can fragment and reassemble fragments correctly in Tunnel Mode

References:

- [RFC 4301] 7
- [RFC 4303] 3.3.4
- [RFC 7296]

Initialization:

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

Databases

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

Policy 1	
Peer	SGW1_Link1
Mode	Tunnel
Remote Address	Link2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
	Payload Length	1240
Fragment Header	Offset	0
	More	1
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP 1

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
Fragment Header	Offset	154
	More	0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	Rest of ICMP Echo Request with ESP

Fragmented ICMP Echo Request with ESP 2

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	TR1_Link1
	Destination Address	NUT_Link0
ICMP	Type	2 (Packet Too Big)
	MTU	1280
	Data	1232 Byte of ICMP Echo Reply with ESP

ICMP Error Message (Packet Too Big)

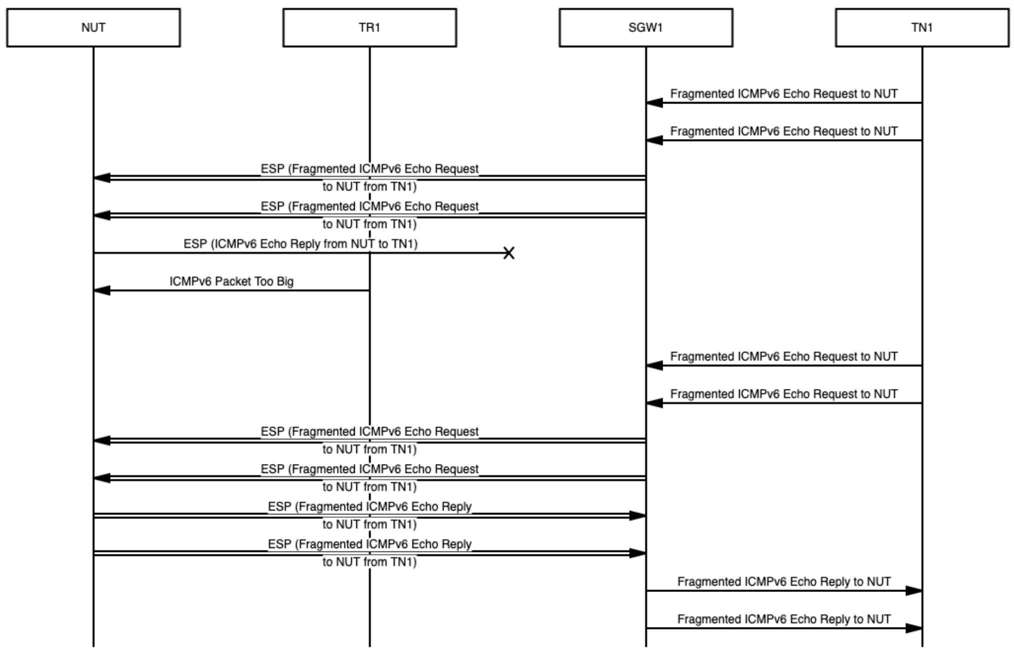
IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
Fragment Header	Offset	0
	More	1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

Fragmented ICMP Echo Reply with ESP 1

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
Fragment Header	Offset	154
	More	0
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	Rest of ICMP Echo Reply with ESP

Fragmented ICMP Echo Reply with ESP 1

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 transmits Fragmented ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP
4.	TR1 transmits ICMP Error Message (Packet Too Big) to the NUT	
5.	SGW1 transmits Fragmented ICMP Echo Request with ESP	
6.	Observe the packets transmitted on Link0	The NUT transmits Fragmented ICMP Echo Reply with ESP 1 and Fragmented ICMP Echo Reply with ESP 2.

Possible Problems:

- When fragmenting the Echo Reply, the DUT may choose to apply fragmentation on the cleartext, or ciphertext side of the IPsec threshold. If the DUT fragments on the ciphertext side, the IPv6 Fragmentation headers will not be visible until decrypted by SGW1.

IPsec.Conf.2.2.5. Tunnel Mode Receipt of No Next Header

Purpose:

Verify that a NUT (End-Node) processes the dummy packet (the protocol value 59) correctly in Tunnel Mode with SGW

References:

- [RFC 4301] 3.2
- [RFC 4303] 2.6
- [RFC 7296]

Initialization:

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

Databases

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

Policy 1	
Peer	SGW1_Link1
Mode	Tunnel
Remote Address	Link2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

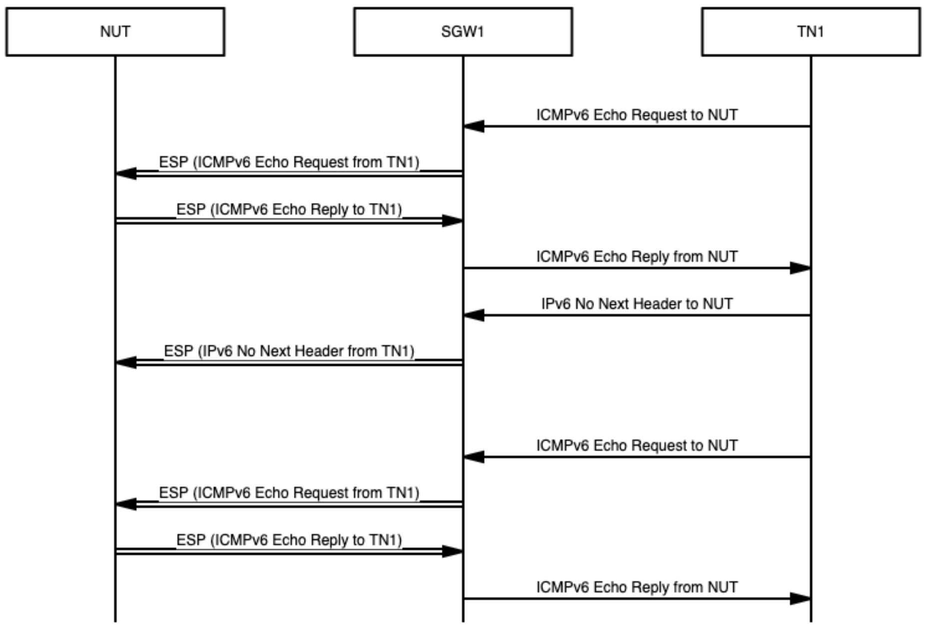
IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
Upper Layer	Data	empty

No Next Header with ESP

Procedure:



Part A: No Next Header

Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP
4.	SGW1 transmits No Next Header with ESP (The ESP sequence number must be incremented according to the packet transmitted at step 2)	
5.	SGW1 transmits ICMP Echo Request with ESP	
6.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP

Part B: TFC Padding with No Next Header

Step	Action	Expected Result
7.	Initialize the NUT	
8.	SGW1 transmits ICMP Echo Request with ESP	

9.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP
10.	SGW1 transmits No Next Header with ESP (The ESP sequence number must be incremented according to the packet transmitted at step 9 and the data in the upper layer consists of random bytes as the plaintext portion)	
11.	SGW1 transmits ICMP Echo Request with ESP	
12.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP

Possible Problems:

- None

IPsec.Conf.2.2.6. Tunnel Mode Bypass Policy

Purpose:

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

References:

- [RFC 4301] 4.4.1
- [RFC 4303]
- [RFC 7296]

Initialization:

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

Databases

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

Policy 1	
Peer	SGW1_Link1
Mode	Tunnel
Remote Address	Link2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Policy 2	
Peer	TN2
Mode	BYPASS
Remote Address	TN2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

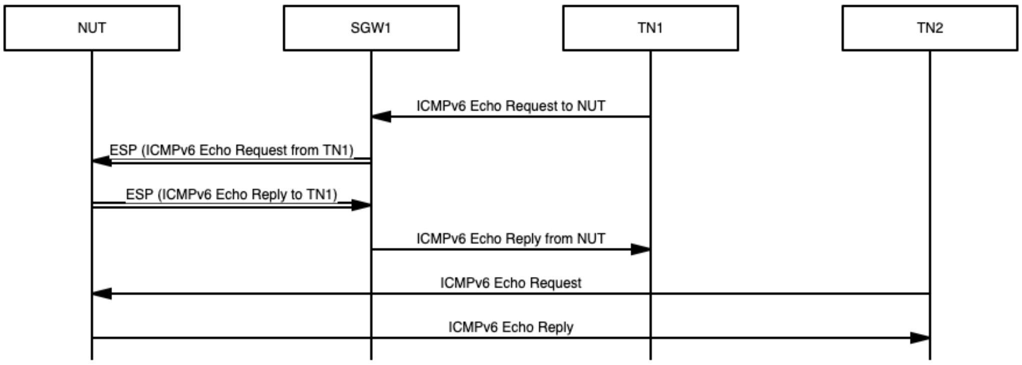
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 transmits ICMP Echo Request with ESP from TN1	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP
4.	TN2 transmits ICMP Echo Request	
5.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply to TN2

Possible Problems:

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

IPsec.Conf.2.2.7. Tunnel Mode Discard Policy

Purpose:

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

References:

- [RFC 4301] 4.4.1
- [RFC 4303]
- [RFC 7296]

Initialization:

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

Databases

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

Policy 1	
Peer	SGW1_Link1
Mode	Tunnel
Remote Address	Link2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Policy 2	
Peer	TN2
Mode	DISCARD
Remote Address	TN2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

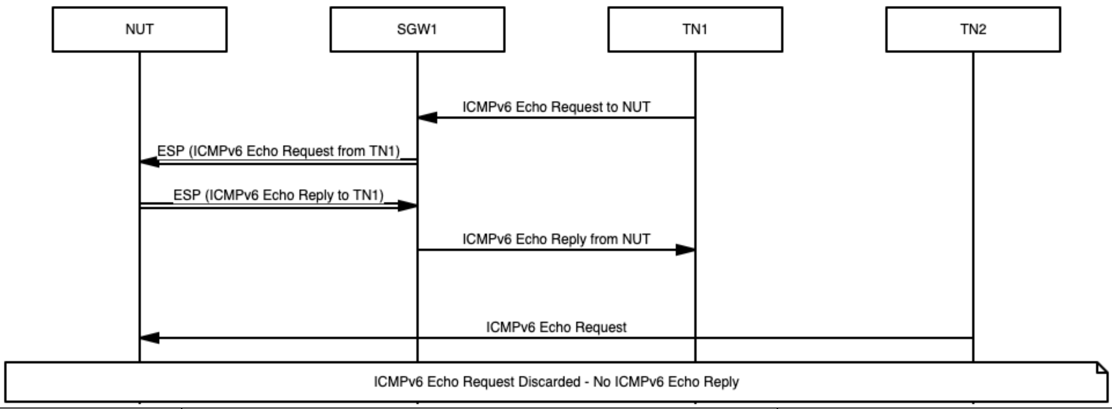
IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 transmits ICMP Echo Request with ESP from TN1	

3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP
4.	TN2 transmits ICMP Echo Request	
5.	Observe the packets transmitted on Link0	The NUT never transmits ICMP Echo Reply to TN2

Possible Problems:

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

IPsec.Conf.2.2.8. Tunnel Mode Padding

Purpose:

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

References:

- [RFC 4301] 4.2
- [RFC 4303] 2.4
- [RFC 7296]

Initialization:

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

Databases

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

Policy 1	
Peer	SGW1_Link1
Mode	Tunnel
Remote Address	Link2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Part A: Tunnel Mode Padding (PadLen 7)

Packets:

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
	Padding	sequential
	Padding Length	7
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP 1

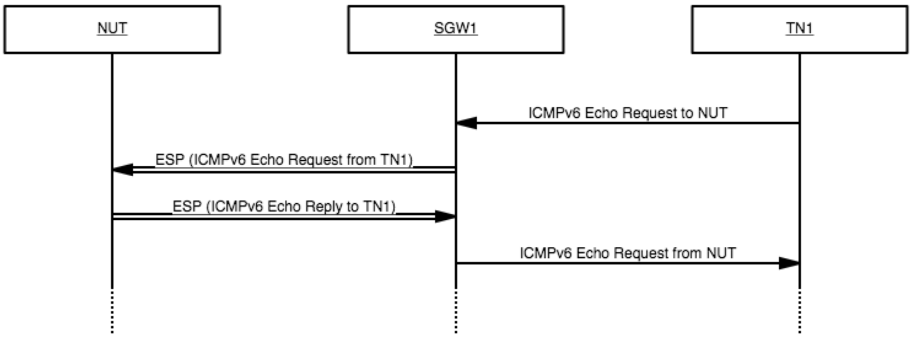
GWIP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
	Padding	sequential
	Padding Length	255
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP 2

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
	Padding Length	7+8n (0 <= n <= 31)
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)
	Data Length	7

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 transmits ICMP Echo Request with ESP 1	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP 1

Part B: Tunnel Mode Padding (PadLen 255)
Packets:

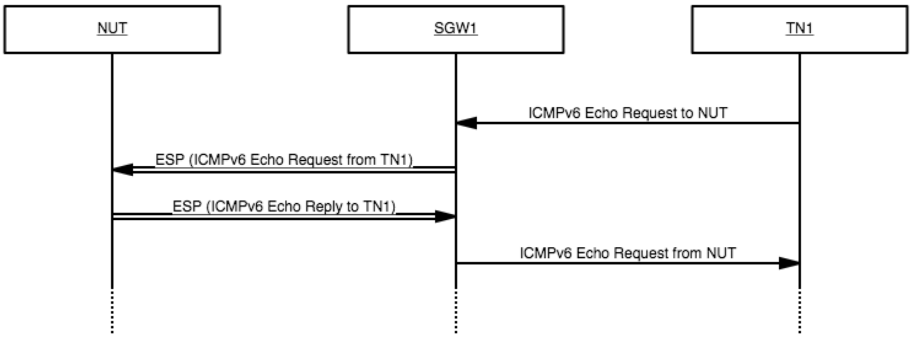
IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
	Padding	sequential
	Padding Length	255
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP 2

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
	Padding Length	7+8n (0 <= n <= 31)
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)
	Data Length	7

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
4.	Initialize the NUT	
5.	SGW1 transmits ICMP Echo Request with ESP 2	
6.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP 2

Part C: TFC enabled Tunnel Mode Padding

Packets:

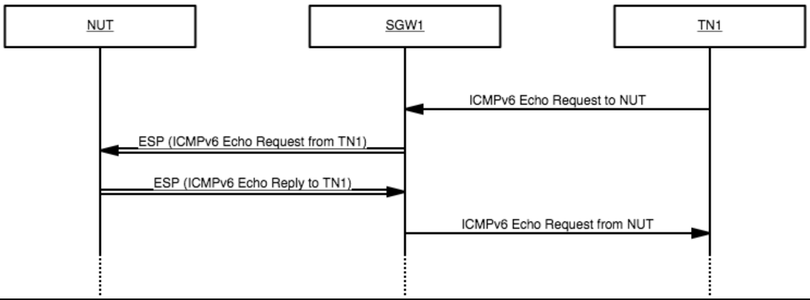
IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP (TFC Padded)

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
7.	Initialize the NUT	
8.	SGW1 transmits ICMP Echo Request with ESP (TFC Padded)	
9.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP

Possible Problems:

- None

IPsec.Conf.2.2.9. Tunnel Mode Invalid SPI

Purpose:
Verify that a NUT (End-Node) correctly processes an, otherwise valid, packet with an invalid SPI in Tunnel Mode

References:

- [RFC 4301] 4.4.2, 4.4.2.1
- [RFC 4303] 2.1
- [RFC 7296]

Initialization:

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

Databases

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

Policy 1	
Peer	SGW1_Link1
Mode	Tunnel
Remote Address	Link2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

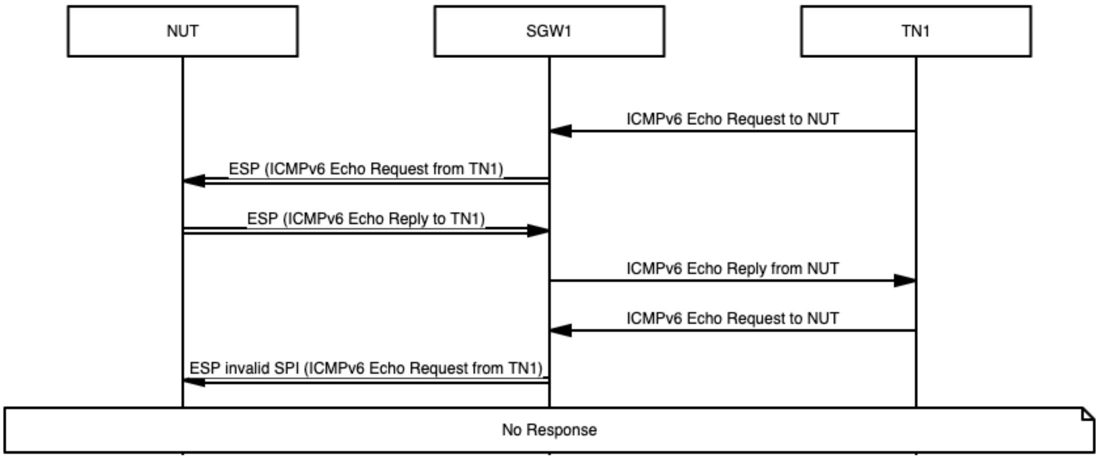
IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Different from SA-I's SPD
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

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

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP with an ESP Sequence Number of 1.
4.	SGW1 transmits ICMP Echo Request with ESP 2 (Non-registered SPI)	
5.	Observe the packets transmitted on Link0	The NUT never transmits ICMP Echo Reply with ESP.

Possible Problems:

- None

IPsec.Conf.2.2.10. Tunnel Mode Invalid ICV

Purpose:
Verify that a NUT (End-Node) correctly processes an, otherwise valid, packet with an invalid ICV in Tunnel Mode.

References:

- [RFC 4301]
- [RFC 4303] 2.8
- [RFC 7296]

Initialization:

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

Databases

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

Policy 1	
Peer	SGW1_Link1
Mode	Tunnel
Remote Address	Link2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

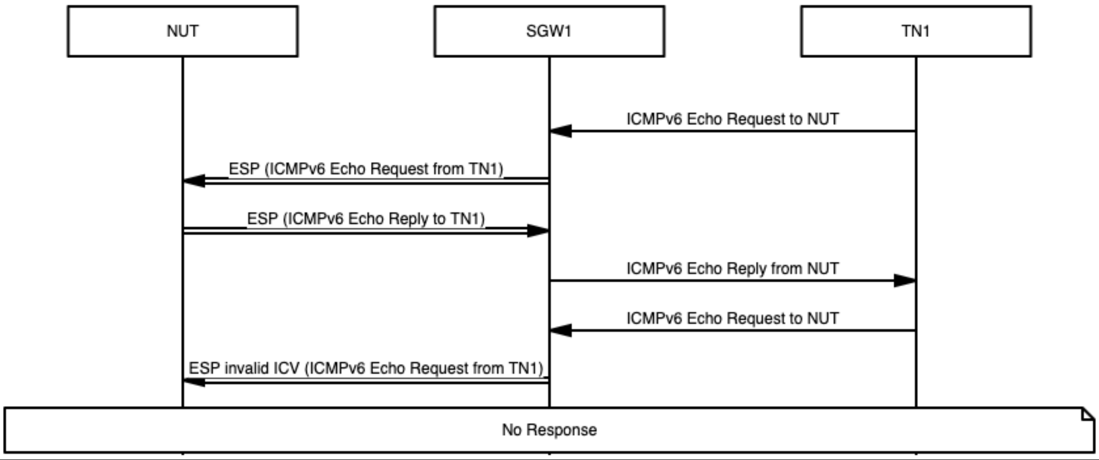
IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Different from SA-I's SPD
	Sequence	1
	Encrypted Data/ICV	SA-I
	ICV	aaaaaaaaaaaaaaaaa......
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)
	Data	"cracked"

ICMP Echo Request with ESP 2 (ICV is modified)

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP with an ESP Sequence Number of 1.
4.	SGW1 transmits ICMP Echo Request with ESP 2 (Invalid ICV)	
5.	Observe the packets transmitted on Link0	The NUT never transmits ICMP Echo Reply with ESP.

Possible Problems:

- None

IPsec.Conf.2.2.11. Tunnel Mode Encrypted PTB Message

Purpose:
Verify that a NUT (End-Node) correctly processes an ICMPv6 Packet Too Big message that has been received encrypted on a Tunnel Mode IPsec SA.

References:

- [RFC 4301] 7
- [RFC 4303] 3.3.4
- [RFC 7296]

Initialization:

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

Databases

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

Policy 1	
Peer	SGW1_Link1
Mode	Tunnel
Remote Address	Link2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
	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	TN1_Link2
	Destination Address	NUT_Link0
	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	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
	Payload Length	*1stPL*
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP 1

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
	Payload Length	*2ndPL*
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of ICMP Echo Request

Fragmented ICMP Echo Request with ESP 2

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	SGW1_LINK1
	Destination Address	NUT_Link0
ICMP	Type	2 (Packet Too Big)
	MTU	1280 <= n <= 1430 (e.g., 1280)
	Data	1232 Byte of ICMP Echo Reply B

ICMP Packet Too Big with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
	Payload Length	1stPL
Fragment	Offset	0
	More Flag	1
ICMP	Type	129 (Echo Reply)

Fragmented ICMP Echo Reply with ESP 1

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
	Payload Length	2ndPL
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of ICMP Echo Reply

Fragmented ICMP Echo Reply with ESP 2

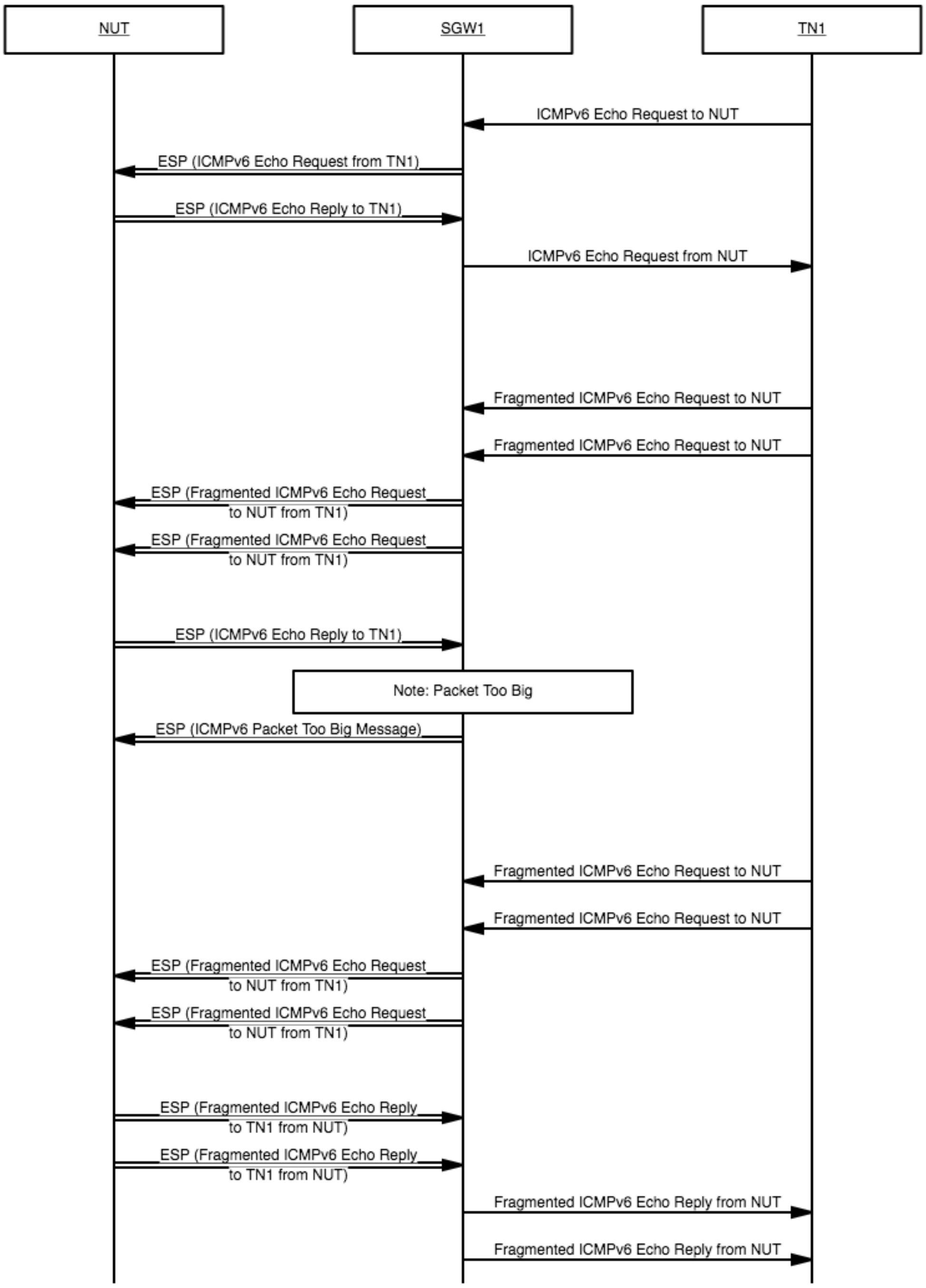
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
	Payload Length	1stPL(=MTU-40) (e.g., 1240)
Fragment	Offset	0
	More Flag	1
ICMP	Type	129 (Echo Reply)

Fragmented ICMP Echo Reply 1

IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
	Payload Length	2ndPL(=1476-1stPL)
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of ICMP Echo Reply

Fragmented ICMP Echo Reply 2

Procedure:



NUT
SGW1
TN1
ICMPv6 Echo Request to NUT
ESP (ICMPv6 Echo Request from TN1)
ESP (ICMPv6 Echo Reply to TN1)
ICMPv6 Echo Request from NUT
Fragmented ICMPv6 Echo Request to NUT
Fragmented ICMPv6 Echo Request to NUT
ESP (Fragmented ICMPv6 Echo Request to NUT from TN1)
ESP (Fragmented ICMPv6 Echo Request to NUT from TN1)
ESP (ICMPv6 Echo Reply to TN1)
Note: Packet Too Big
ESP (ICMPv6 Packet Too Big Message)
Fragmented ICMPv6 Echo Request to NUT
Fragmented ICMPv6 Echo Request to NUT
ESP (Fragmented ICMPv6 Echo Request to NUT from TN1)
ESP (Fragmented ICMPv6 Echo Request to NUT from TN1)
ESP (Fragmented ICMPv6 Echo Reply to TN1 from NUT)
ESP (Fragmented ICMPv6 Echo Reply to TN1 from NUT)
Fragmented ICMPv6 Echo Reply from NUT
Fragmented ICMPv6 Echo Reply from NUT

Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 sends ICMP Echo Request with ESP from TN1 to NUT	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP to TN1
4.	SGW1 sends Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2 from TN1 to the NUT	
5.	Observe the packets transmitted on Link0	The NUT reassembles ICMP Echo Request and transmits fully assembled ICMP Echo Reply with ESP to TN1
6.	SGW1 sends ICMP Packet Too Big Message with ESP to the NUT	
7.	SGW1 sends ICMP Echo Request with ESP 1 and ICMP Echo Request with ESP 2 from TN1 to the NUT	
8.	Observe the packets transmitted on Link0	The NUT reassembles ICMP Echo Request and transmits Fragmented ICMP Echo Reply with ESP 1 and Fragmented ICMP Echo Reply with ESP 2 to TN1

Possible Problems:

- None

IPsec.Conf.2.2.12. Tunnel Mode with End-Node

Purpose:

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

References:

- [RFC 4301] 4.4.1.1
- [RFC 4303]
- [RFC 7296] 2.9

Initialization:

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

Databases:

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

Policy 1	
Peer	EN1_Link1
Mode	Tunnel
Remote Address	EN1_Link1
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

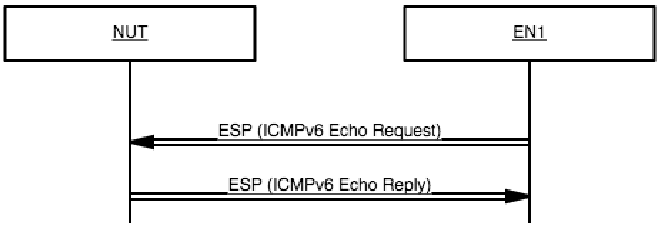
IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP

Possible Problems:

- None

Section 3: IPsec SGW

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

3.1. IPsec/ESP Architecture

IPsec.Conf.3.1.1. Select SPD (2 SGW Peers)

Purpose:

Verify that a NUT (SGW) selects appropriate SPD

References:

- [RFC 4301] 4.4.1.1
- [RFC 4303]
- [RFC 7296] 2.9

Initialization:

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

Databases

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

Policy 1	
Peer	SGW1_Link2
Mode	Tunnel
Remote Traffic Selector	Link3
Local Traffic Selector	Link0
Protocol/Port	ANY/ANY

Policy 2	
Peer	SGW2_Link2
Mode	Tunnel
Remote Address	Link4
Local Address	Link0
Protocol/Port	ANY/ANY

Packets

IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 1

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA1-I
IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA1-I's ESP

IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply 1

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA1-O
IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA1-O's ESP

IP Header	Source Address	TN4_Link4
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 2

IP Header	Source Address	SGW2_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic3
	Sequence	1
	Encrypted Data/ICV	SA2-I
IP Header	Source Address	TN4_Link4
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA2-I's ESP

IP Header	Source Address	TN1_Link0

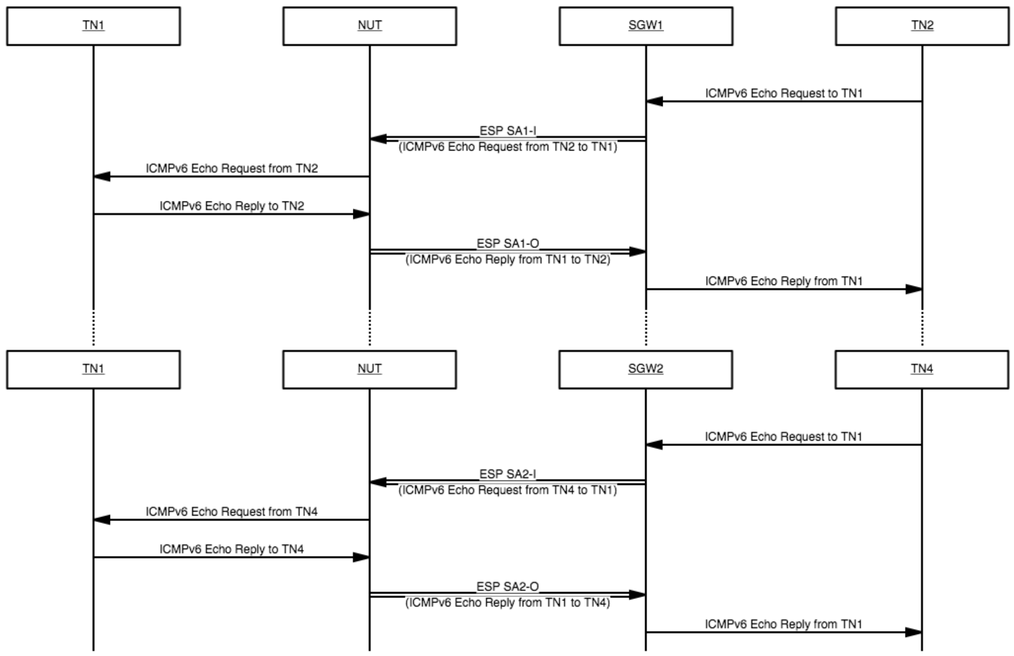
	Destination Address	TN4_Link4
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply 2

IP Header	Source Address	NUT_Link1
	Destination Address	SGW2_Link2
ESP	SPI	Dynamic4
	Sequence	1
	Encrypted Data/ICV	SA2-O
IP Header	Source Address	TN1_Link0
	Destination Address	TN4_Link4
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA2-O's ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 transmits ICMP Echo Request with SA1-I's ESP (originally from TN2)	
3.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request 1
4.	TN1 sends ICMP Echo Reply 1	
5.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Reply with SA1-O's ESP
6.	SGW2 transmits ICMP Echo Request with SA2-I's ESP (originally from TN4)	
7.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request 2
8.	TN1 sends ICMP Echo Reply 2	
9.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Reply with SA2-O's ESP

Possible Problems:

- None

IPsec.Conf.3.1.2. Select SPD (2 Hosts behind same Peer)

Purpose:

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

References:

- [RFC 4301] 4.4.1.1
- [RFC 4303]
- [RFC 7296] 2.9

Initialization:

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

Databases:

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

Policy 1	
Peer	SGW1_Link2
Mode	Tunnel
Remote Traffic Selector	TN2_Link3
Local Traffic Selector	Link0
Protocol/Port	ANY/ANY

Policy 2	
Peer	SGW1_Link2
Mode	Tunnel
Remote Address	TN3_Link3
Local Address	Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 1

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA1-I
IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA1-I's ESP

IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply 1

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA1-O
IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA1-O's ESP

IP Header	Source Address	TN3_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 2

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic3
	Sequence	1
	Encrypted Data/ICV	SA2-I
IP Header	Source Address	TN3_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with SA2-I's ESP

IP Header	Source Address	TN1_Link0

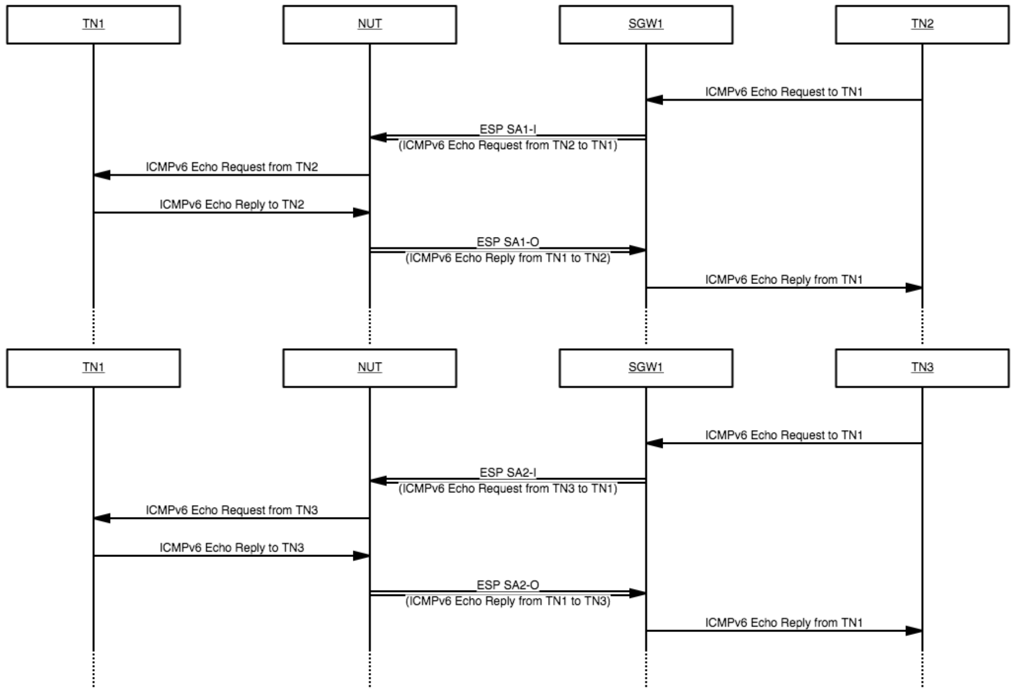
	Destination Address	TN3_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply 2

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	Dynamic4
	Sequence	1
	Encrypted Data/ICV	SA2-O
IP Header	Source Address	TN1_Link0
	Destination Address	TN3_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with SA2-O's ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits ICMP Echo Request with SA1-I's ESP (originally from TN2)	
3.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request 1
4.	TN1 sends ICMP Echo Reply 1	
5.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Reply with SA1-O's ESP
6.	EN1 sends ICMP Echo Request with SA2-I's ESP (originally from TN3)	
7.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request 2
8.	TN1 sends ICMP Echo Reply 2	
9.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Reply with SA2-O's ESP

Possible Problems:

- None

IPsec.Conf.3.1.3. Sequence Number Increment

Purpose:

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

References:

- [RFC 4301] 4.4.2.1
- [RFC 4303] 2.2
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	SGW1_Link2
Mode	Tunnel
Remote Traffic Selector	Link3
Local Traffic Selector	Link0
Protocol/Port	ANY/ANY

Packets:

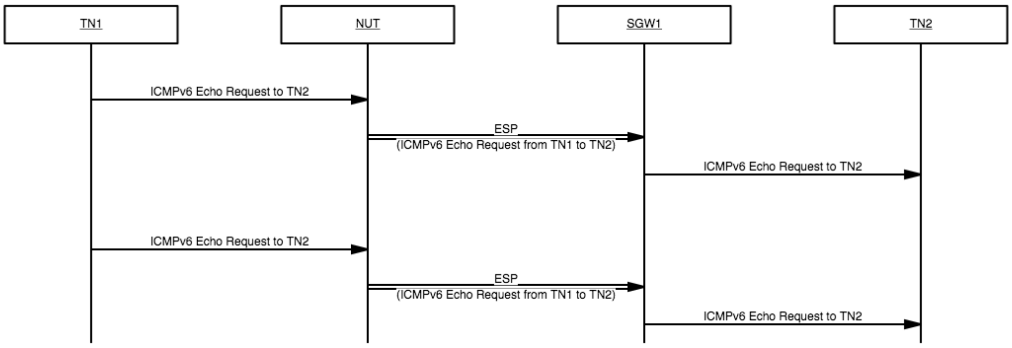
IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	Dynamic2
	Sequence	1st = 1, 2nd = 2
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 sends ICMP Echo Request	
3.	Observe the packets transmitted on Link0 and Link1	The NUT transmits an ICMP Echo Request with ESP with an ESP Sequence number of 1
4.	TN1 sends ICMP Echo Request	
5.	Observe the packets transmitted on Link0 and Link1	The NUT transmits an ICMP Echo Request with ESP with an ESP Sequence number of 2

Possible Problems:

- None

IPsec.Conf.3.1.4. Packet Too Big Transmission

Purpose:

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

References:

- [RFC 4301] 7
- [RFC 4303] 3.3.4
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	SGW1_Link2
Mode	Tunnel
Remote Traffic Selector	Link3
Local Traffic Selector	Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
	Payload Length	1460
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link0
ICMP	Type	2 (Packet Too Big)
	MTU	1280 <= n <= 1430 (e.g., 1280)
	Data	1232 Byte of ICMP Echo Request

ICMP Error Message (Packet Too Big)

IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
	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	TN1_Link0
	Destination Address	TN2_Link3
	Payload Length	2ndPL(=1476-1stPL)
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of ICMP Echo Request

Fragmented ICMP Echo Request 2

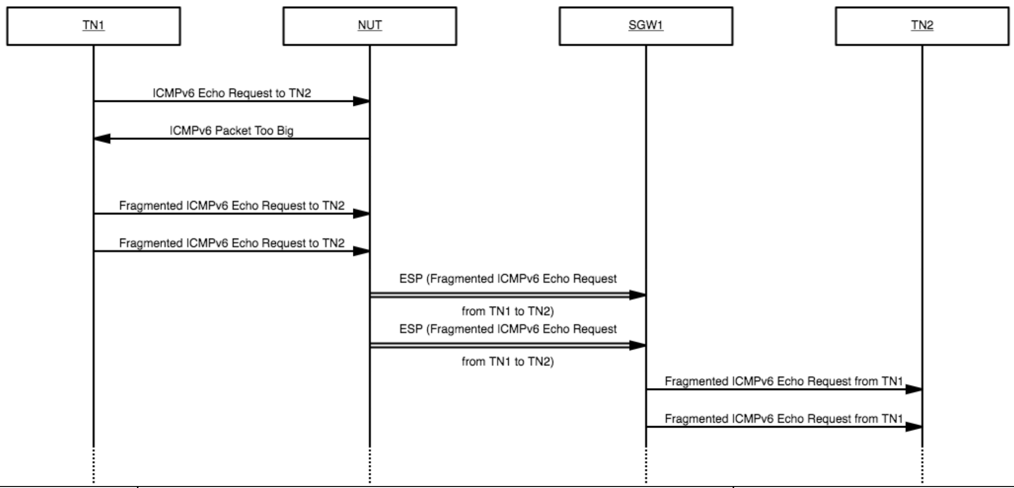
IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
	Payload Length	1stPL
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP 1

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
	Payload Length	2ndPL
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of ICMP Echo Request

Fragmented ICMP Echo Request with ESP 2

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 sends ICMP Echo Request	
3.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Error Message (Packet Too Big)
4.	TN1 sends Fragmented ICMP Echo Request 1 and Fragmented ICMP Echo Request 2	
5.	Observe the packets transmitted on Link0 and Link1	The NUT transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2

Possible Problems:

- None

IPsec.Conf.3.1.5. Packet Too Big Forwarding

Purpose:

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

References:

- [RFC 4301] 7
- [RFC 4303] 3.3.4
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	SGW1_Link2
Mode	Tunnel
Remote Traffic Selector	Link3
Local Traffic Selector	Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
	Payload Length	1360
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
	Payload Length	1360
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	TR1_Link2
	Destination Address	NUT_Link1
ICMP	Type	2 (Packet Too Big)
	MTU	1356
	Data	1232 Byte of ICMP Echo Request

ICMP Error Message to NUT (Packet Too Big)

IP Header	Source Address	TR1_Link2 or NUT_Link1
	Destination Address	TN1_Link0
ICMP	Type	2 (Packet Too Big)
	MTU	1280 – 1286
	Data	1232 Byte of ICMP Echo Request

ICMP Error Message to TN1 (Packet Too Big)

IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
	Payload Length	1240
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request 1

IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
	Payload Length	136
Fragment	Offset	154
	More Flag	0
Data	Data	Rest of ICMP Echo Request

Fragmented ICMP Echo Request 2

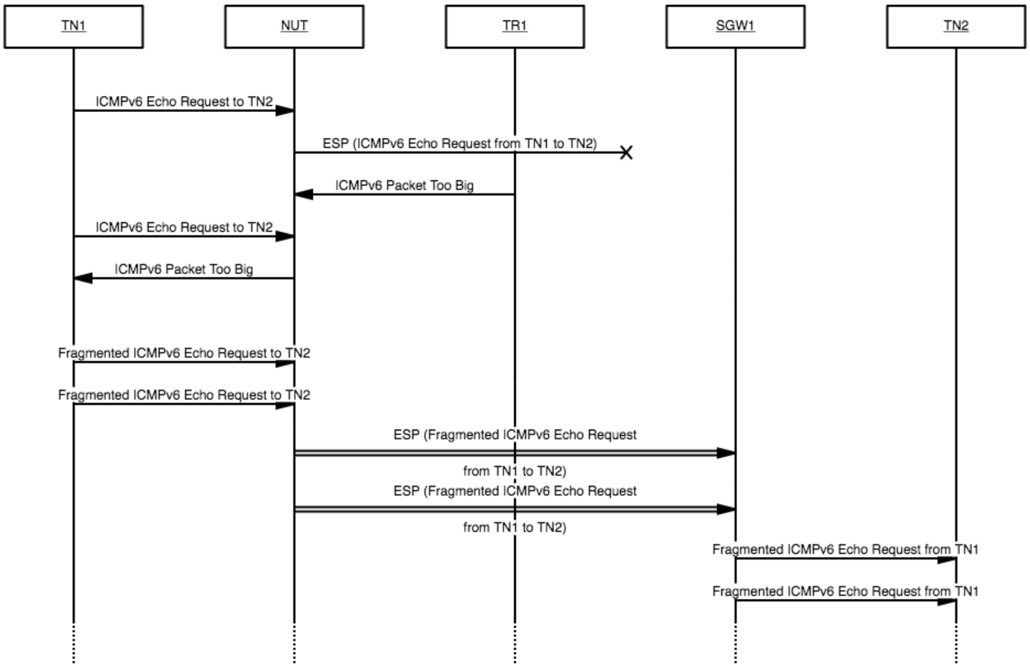
IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
	Payload Length	1240
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request with ESP 1

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
	Payload Length	136
Fragment	Offset	154
	More Flag	0
Data	Data	Rest of ICMP Echo Request

Fragmented ICMP Echo Request with ESP 2

Procedure:

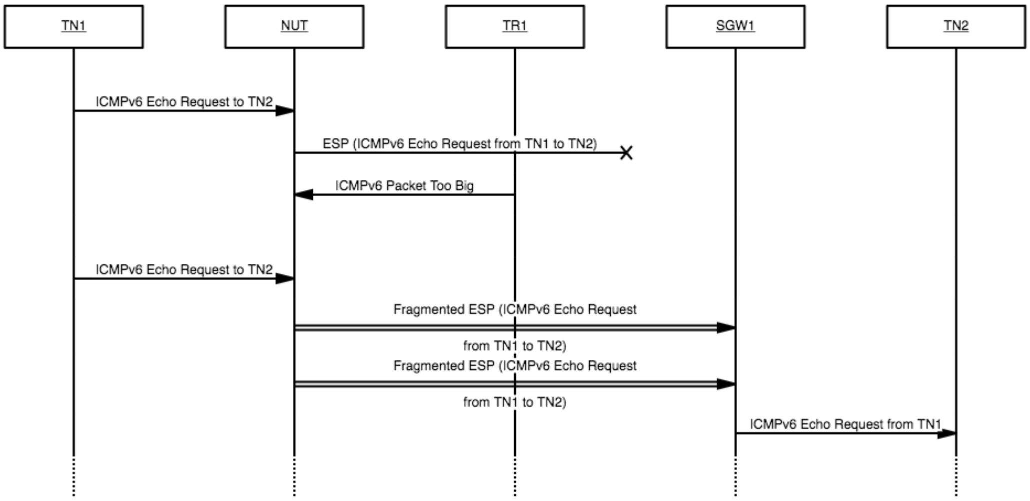


TN1
NUT
TR1
SGW1
TN2
ICMPv6 Echo Request to TN2
ESP (ICMPv6 Echo Request from TN1 to TN2)
ICMPv6 Packet Too Big
ICMPv6 Echo Request to TN2
ICMPv6 Packet Too Big
Fragmented ICMPv6 Echo Request to TN2
Fragmented ICMPv6 Echo Request to TN2
ESP (Fragmented ICMPv6 Echo Request
from TN1 to TN2)
ESP (Fragmented ICMPv6 Echo Request
from TN1 to TN2)
Fragmented ICMPv6 Echo Request from TN1
Fragmented ICMPv6 Echo Request from TN1

Step	Action	Expected Result
1.	Initialize the NUT	
2.	TN1 sends ICMP Echo Request	
3.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request with ESP
4.	TR1 sends ICMP Error Message to NUT (Packet Too Big)	
5.	TN1 sends ICMP Echo Request	The NUT transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2
6.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Error Message to TN1 (Packet Too Big)
7.	TN1 sends Fragmented ICMP Echo Request 1 and Fragmented ICMP Echo Request 2	
8.	Observe the packets transmitted on Link0 and Link1	The NUT transmits Fragmented ICMP Echo Request with ESP 1 and Fragmented ICMP Echo Request with ESP 2

Possible Problems:

- The NUT (SGW) may choose to process the ICMPv6 Packet Too Big PMTU information on the ciphertext side of the interface. In this case, the NUT will not generate and send a Packet Too Big Message to TN1 but will instead transmit fragmented ESP Packets from after tunneling and applying ESP to the Echo Request from TN1. TN1 will continue to transmit whole packets. See RFC 4301 Section 2.1, and reference diagram below.


TN1
NUT
TR1
SGW1
TN2
ICMPv6 Echo Request to TN2
ESP (ICMPv6 Echo Request from TN1 to TN2)
ICMPv6 Packet Too Big
ICMPv6 Echo Request to TN2
Fragmented ESP (ICMPv6 Echo Request
from TN1 to TN2)
Fragmented ESP (ICMPv6 Echo Request
from TN1 to TN2)
ICMPv6 Echo Request from TN1

IPsec.Conf.3.1.6. Receipt of No Next Header

Purpose:

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

References:

- [RFC 4301] 3.2
- [RFC 4303] 2.6
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	SGW1_Link2
Mode	Tunnel
Remote Traffic Selector	Link3
Local Traffic Selector	Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

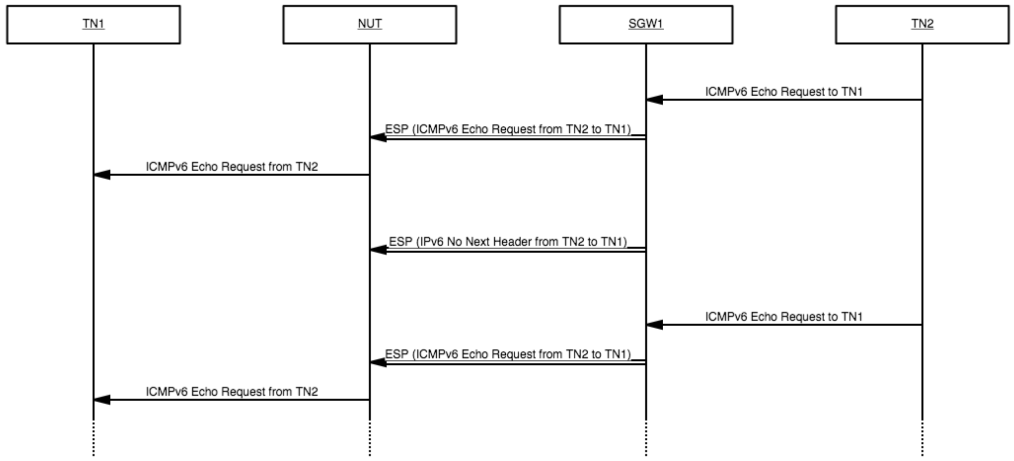
ICMP Echo Request with ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
	Next Header	no next header (59)
Upper Layer	Data	See below

No Next Header with ESP

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

Procedure:



Part A: No Next Header

Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 sends ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request
4.	EN1 sends No Next Header with ESP	
5.	The ESP sequence number must be 1 greater than the packet transmitted at step 2	
6.	Observe the packets transmitted on Link0 and Link1	The NUT does not transmit any packets
7.	EN1 sends ICMP Echo Request with ESP	
8.	The ESP sequence number must be 1 greater than the packet transmitted at step 4	
9.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request

Part B: TFC Padding with No Next Header

Step	Action	Expected Result
10.	Initialize the NUT	
11.	EN1 sends ICMP Echo Request with ESP	

12.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request
13.	EN1 sends No Next Header with ESP	
14.	The ESP sequence number must be 1 greater than the packet transmitted at step 2	
15.	Observe the packets transmitted on Link0 and Link1	The NUT does not transmit any packets
16.	EN1 sends ICMP Echo Request with ESP	
17.	The ESP sequence number must be 1 greater than the packet transmitted at step 4	
18.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request

Possible Problems:

- None

IPsec.Conf.3.1.7. Bypass Policy

Purpose:

Verify that a NUT (SGW) can utilize Bypass Policy

References:

- [RFC 4301] 4.4.1
- [RFC 4303]
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	SGW1_Link2
Mode	Tunnel
Remote Traffic Selector	Link3
Local Traffic Selector	Link0
Protocol/Port	ANY/ANY

Policy 2	
Peer	N/A
Mode	BYPASS
Remote Address	Link4
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 1

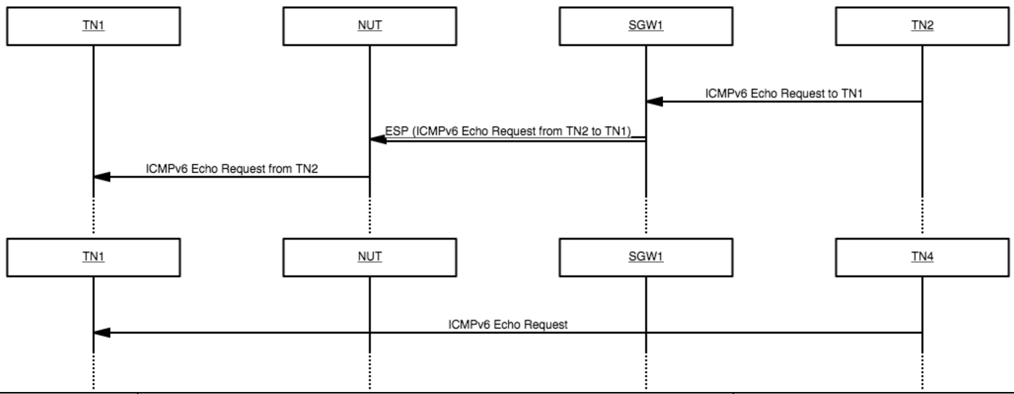
IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	TN4_Link4
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 2

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 sends ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request 1
4.	SGW1 forwards ICMP Echo Request 2	
5.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request 2

Possible Problems:

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

IPsec.Conf.3.1.8. Discard Policy

Purpose:

Verify that a NUT (SGW) can utilize Discard Policy

References:

- [RFC 4301] 4.4.1
- [RFC 4303]
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	SGW1_Link2
Mode	Tunnel
Remote Traffic Selector	Link3
Local Traffic Selector	Link0
Protocol/Port	ANY/ANY

Policy 2	
Peer	N/A
Mode	DISCARD
Remote Address	Link4
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 1

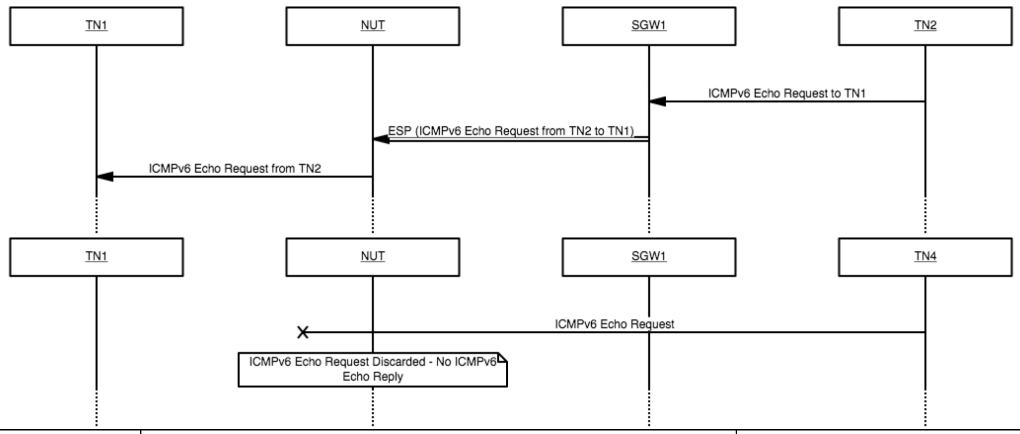
IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	TN4_Link4
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request 2

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 sends ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request 1
4.	TN4 sends ICMP Echo Request 2	
5.	Observe the packets transmitted on Link0 and Link1	The NUT never transmits ICMP Echo Request 2

Possible Problems:

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

IPsec.Conf.3.1.9. Tunnel Mode Padding

Purpose:

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

References:

- [RFC 4301] 4.2
- [RFC 4303] 2.4
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	SGW1_Link2
Mode	Tunnel
Remote Traffic Selector	Link3
Local Traffic Selector	Link0
Protocol/Port	ANY/ANY

Part A: Tunnel Mode Padding (PadLen 7)

Packets:

IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
	Padding	Sequential
	Padding Length	7+8n (0 <= n <= 31)
IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP

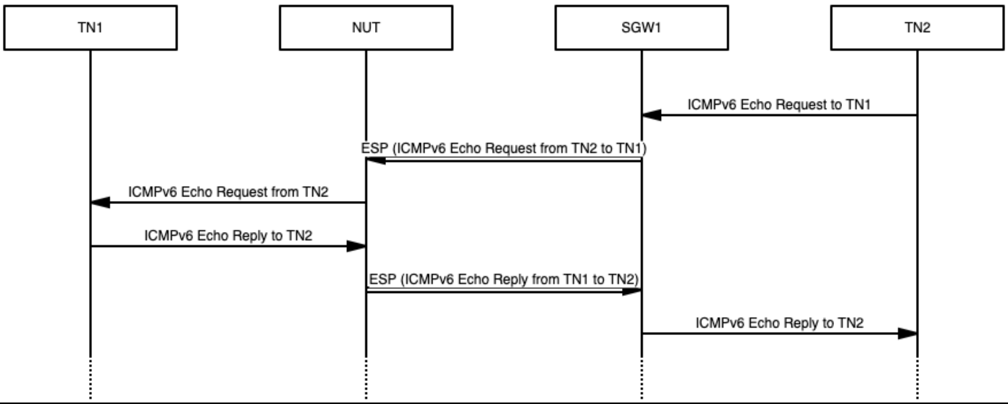
IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
	Padding	Sequential
	Padding Length	7+8n (0 <= n <= 31)
IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
ICMP	Type	129 (Echo Reply)
	Data Length	7

ICMP Echo Reply with ESP

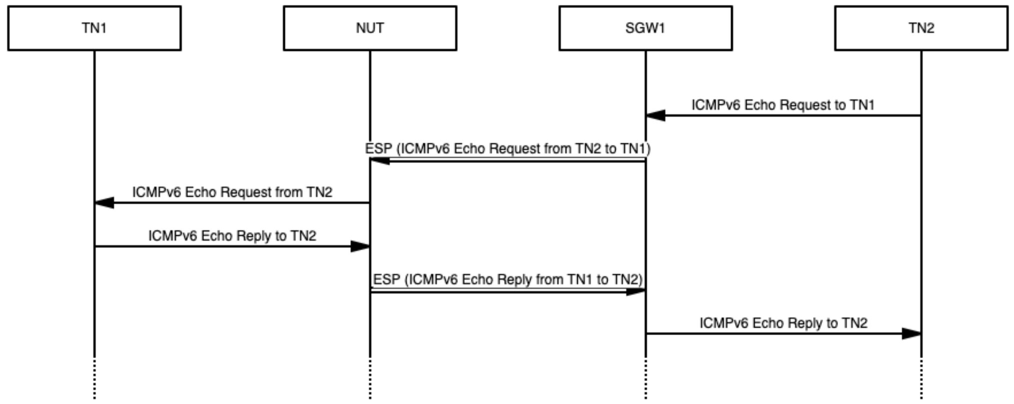
Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 sends ICMP Echo Request with ESP (Padding length=7)	
3.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request
4.	TN1 sends ICMP Echo Reply	
5.	Observe the packet transmitted by NUT	The NUT transmits ICMP Echo Reply with ESP
6.	SGW1 sends ICMP Echo Request with ESP (Padding length=255)	
7.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request
8.	TN1 sends ICMP Echo Reply	
9.	Observe the packet transmitted by NUT	The NUT transmits ICMP Echo Reply with ESP

Part B: Tunnel Mode Padding (PadLen 255)

Packets:
Same as Part A with Padding Length 255



Step	Action	Expected Result
10.	Initialize the NUT	
11.	SGW1 sends ICMP Echo Request with ESP (Padding length=255)	
12.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request
13.	TN1 sends ICMP Echo Reply	
14.	Observe the packet transmitted by NUT	The NUT transmits ICMP Echo Reply with ESP

Part C: TFC enabled Tunnel Mode Padding

Packets:

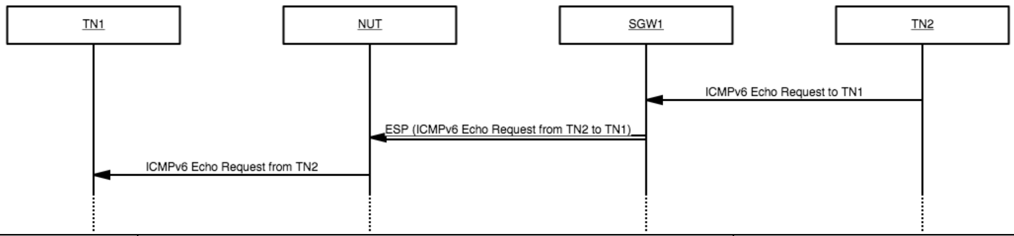
IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP (TFC Padded)

IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

Procedure:



Step	Action	Expected Result
15.	Initialize the NUT	
16.	SGW1 sends ICMP Echo Request with ESP (TFC Padded)	
17.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request

Possible Problems:

- None

IPsec.Conf.3.1.10. Invalid SPI

Purpose:

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

References:

- [RFC 4301] 4.4.2, 4.4.2.1
- [RFC 4303] 2.1
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	SGW1_Link2
Mode	Tunnel
Remote Traffic Selector	Link3
Local Traffic Selector	Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

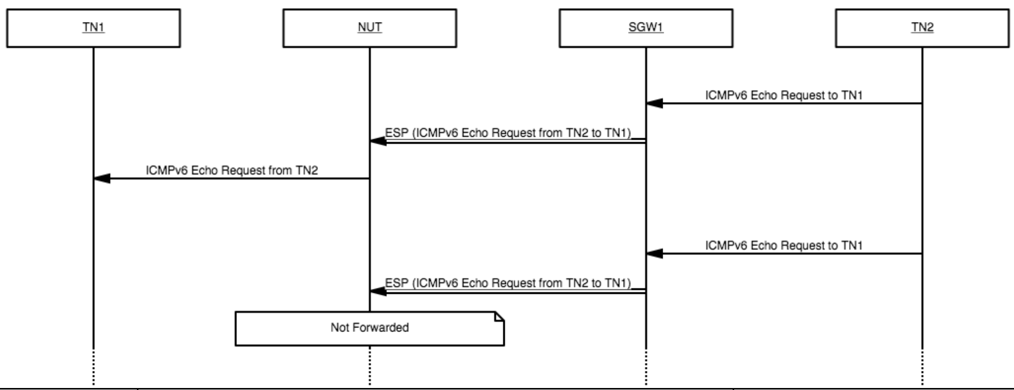
IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic1
	Sequence Number	1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Different from SA-I's SPD
	Sequence Number	1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP (Non-registered SPI)

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 sends ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request
4.	SGW1 sends ICMP Echo Request with ESP (Non-registered SPI)	
5.	Observe the packets transmitted on Link0 and Link1	The NUT never transmits ICMP Echo Request

Possible Problems:

- None

IPsec.Conf.3.1.11. Invalid ICV

Purpose:

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

References:

- [RFC 4301]
- [RFC 4303] 2.8
- [RFC 7296]

Initialization:

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

Databases:

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

Policy 1	
Peer	SGW1_Link2
Mode	Tunnel
Remote Traffic Selector	Link3
Local Traffic Selector	Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)
	Data	"PadLen is zero"

ICMP Echo Request

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

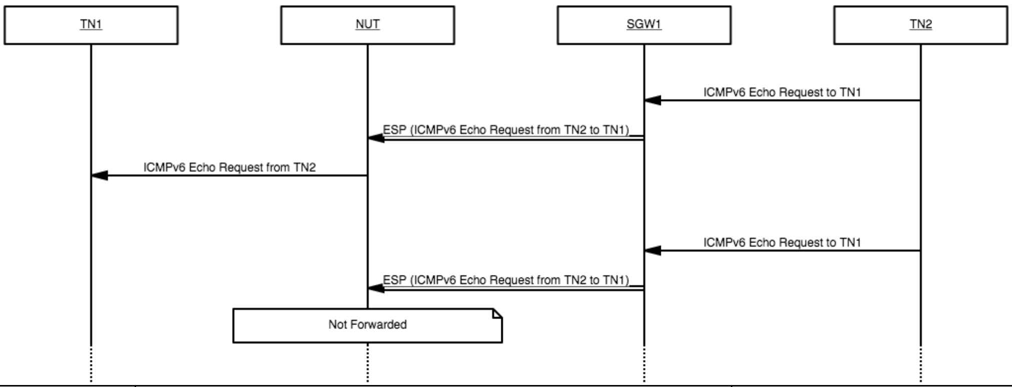
	Data	"PadLen is zero"

ICMP Echo Request with ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic1
	Sequence	2
	Encrypted Data/ICV	SA-I
	ICV	aaaaaaaaa........
IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)
	Data	"cracked"

ICMP Echo Request with ESP (Incorrect ICV)

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 sends ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request
4.	SGW1 sends ICMP Echo Request with ESP (Incorrect ICV)	
5.	Observe the packets transmitted on Link0 and Link1	The NUT never transmits ICMP Echo Request

Possible Problems:

- None

IPsec.Conf.3.1.12. Tunnel Mode with End-Node

Purpose:

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

References:

- [RFC 4301] 4.4.1.1
- [RFC 4303]
- [RFC 7296] 2.9

Initialization:

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

Databases:

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

Policy 1	
Peer	EN1_Link2
Mode	Tunnel
Remote Traffic Selector	EN1_Link2
Local Traffic Selector	Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	EN1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	SGW1_Link2
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	EN1_Link2
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

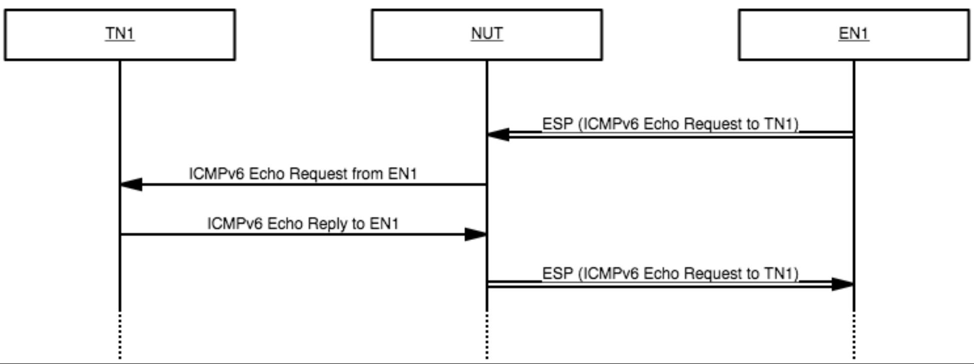
IP Header	Source Address	TN1_Link0
	Destination Address	EN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

IP Header	Source Address	NUT_Link1
	Destination Address	EN1_Link2
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	TN1_Link0
	Destination Address	EN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Request
4.	TN1 transmits ICMP Echo Reply	
5.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Reply with ESP

Possible Problems:

- None

Section 4: Algorithms

This Chapter reviews the test cases for the various algorithms that are used with IKEv2 and IPsec/ESP.

4.1. ESP Algorithms

ESP Common Configurations

Algorithm List

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

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

IPsec.Conf.4.1.1. End-Node ESP Algorithms (Transport Mode)

Purpose:

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

References:

- [RFC 4301]
- [RFC 4303]
- [RFC 7296]
- [RFC 8221]

Initialization:

- Topology
    - Connect the devices according to Common Topology
- Configuration
    - Use ESP Common Configurations combined with the below configurations.
    - In addition, use the algorithms specified in each part.

Databases:

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

Policy 1	
Peer	EN1_Link1
Mode	Transport
Remote Address	EN1_Link1
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

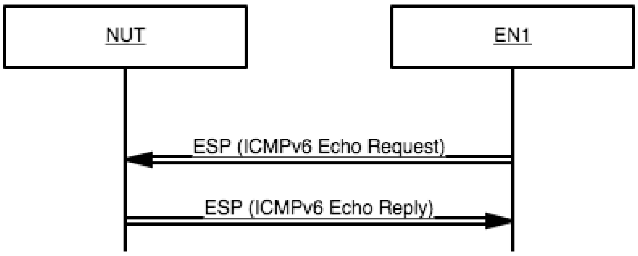
IP Header	Source Address	EN1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	EN1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



All Parts: Algorithms

Step	Action	Expected Result
1.	Initialize the NUT	
2.	EN1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP

Possible Problems:

- None

IPsec.Conf.4.1.2. End-Node ESP Algorithms (Tunnel Mode)

Purpose:

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

References:

- [RFC 4301]
- [RFC 4303]
- [RFC 7296]
- [RFC 8221]

Initialization:

- Topology
  - Connect the devices according to Common Topology
- Configuration
  - Use ESP Common Configurations combined with the below configurations
  - In addition, use the algorithms specified in each part.

Databases:

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

Policy 1	
Peer	SGW1_Link1
Mode	Tunnel
Remote Address	Link2
Local Address	NUT_Link0
Protocol/Port	ANY/ANY

Packets:

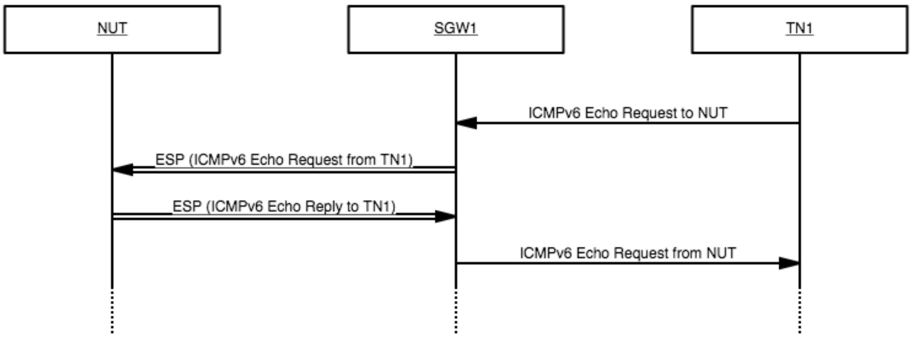
IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	NUT_Link0
	Destination Address	TN1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



All Parts: Algorithms

Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0	The NUT transmits ICMP Echo Reply with ESP

Possible Problems:

- None

IPsec.Conf.4.1.3. SGW ESP Algorithms

Purpose:

Verify that an SGW device can correctly utilize various algorithms

References:

- [RFC 4301]
- [RFC 4303]
- [RFC 7296]
- [RFC 8221]

Initialization:

- Topology
    - Connect the devices according to Common Topology 4
- Configuration
    - Use ESP Common Configurations combined with the below configurations
    - In addition, use the algorithms specified in each part.

Databases:

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

Policy 1	
Peer	SGW1_Link2
Mode	Tunnel
Remote Traffic Selector	Link3
Local Traffic Selector	Link0
Protocol/Port	ANY/ANY

Packets:

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	Dynamic1
	Sequence	1
	Encrypted Data/ICV	SA-I
IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request with ESP

IP Header	Source Address	TN2_Link3
	Destination Address	TN1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

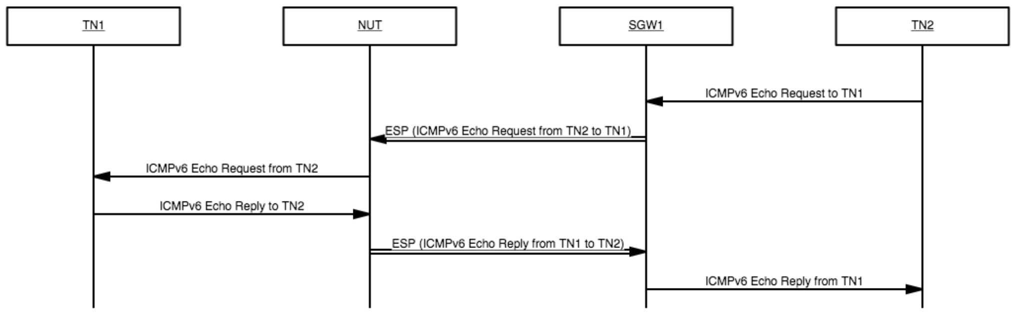
IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	Dynamic2
	Sequence	1
	Encrypted Data/ICV	SA-O
IP Header	Source Address	TN1_Link0
	Destination Address	TN2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply with ESP

Procedure:



All Parts: Algorithms

Step	Action	Expected Result
1.	Initialize the NUT	
2.	SGW1 transmits ICMP Echo Request with ESP	
3.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Request
4.	TN1 transmits ICMP Echo Reply	
5.	Observe the packets transmitted on Link0 and Link1	The NUT transmits ICMP Echo Reply with ESP

Possible Problems:

- None

Modification Record

Version	Date	Editor	Modification
2.0.1	2024-01-05	Christopher Brown	• Removed SHA1 • Updated Requirements in to match with USGv6 Revision 1. • Added Definitions section • Added Possible Problems section • Removed Manual Keys.
2.0.1	2022-04-01	Timothy Carlin	• Bug fixes and Edits • Update Required Tests Table • Update Topology Diagrams • Add/Update References
2.0.0	2021-08-04	Timothy Carlin	Reorganized sections Separated ESP from Architecture tests Common Configuration for Manual Keys and Policies Updated Algorithm Requirements according to RFC7321bis Added CHAHA20-POLY1305 to ADVANCED encryption algorithms Changed AES-CBC(128-bit) and NULL from ADVANCED to BASIC encryption algorithms Changed 3DES-CBC from BASIC to ADVANCED encryption algorithms Added AES-GCM(128-bit) to BASIC encryption algorithms Added AES-CBC (192-bit), AES-CBC(256-bit), AES-GCM(192-bit), and AES-GCM(256-bit) to ADVANCED encryption algorithms Changed HMAC-SHA-256 from ADVANCED to BASIC Integrity algorithms Added AES-GMAC(128-bit) to BASIC Integrity algorithms Added HMAC-SHA-384, HMAC-SHA-512, AES-GMAC(192-bit), and AES-GMAC(256-bit) to ADVANCED Integrity algorithms Added test cases for AES-CBC(128-bit) HMAC-SHA-256 (Section 5.2.9, 6.2.9) Added test cases for AES-CBC HMAC-SHA-384 (Section 5.2.10, 6.2.10) Added test cases for AES-CBC(256-bit) HMAC-SHA-512 (Section 5.2.11, 6.2.11) Added test cases for AES-GCM NULL (Section 5.2.12, 6.2.12), RFC 4106 "The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)" Added test cases for NULL AES-GMAC (Section 5.2.13, 6.2.13), RFC 4543 "The Use of Galois Message Integrity Code (GMAC) in IPsec ESP and AH Added IKEv2-Specific test cases Modified formatting and fixed typos

1.11.0	2011-10-05	Timothy Carlin	Added Section 5.3.6 to verify that End-Node can process a tunneled ICMPv6 Packet Too Big Message and correctly reassemble/fragment packet Modified Section 5.1 End-Node Transport Mode Packet Too Big Reception to fragment inbound Echo Request. Removed ESP Null Authentication Tests Typos and Bug Fixes
1.10.0	2010-05-31	Timothy Carlin	Support Authentication Algorithm HMAC-SHA-256 in RFC 4868 (Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec) (Section 5.2.8, and 6.2.8) Added the description to Section 6.1.6 Possible Problems
1.9.2	2010-02-03		Corrected pre-shared key at subsection 5.1.5 Corrected packet format of dummy packet at subsection 6.1.7 Clarified relationship between steps in procedure and Observable Result at all subsections.
1.9.1	2009-01-07		Support the passive node which doesn't have ping6 application (as Possible Problems in Section 5.1.2)
1.9.0	2008-12-09		
1.8.1	2007-10-11		Support RFC 4312 (The Camellia Cipher Algorithm and Its Use With IPsec) (Section 5.2.7, 6.2.7) Use IPv6 prefix defined in RFC 3849 for the documentation Remove ESN test cases (Section 5.1.12, 6.1.14)
1.8.0	2007-05-27		Support IPsec v3
1.7.7	2006-05-06		Correct 5.3.4 Category
1.7.6	2005-12-22		Correct expected MTU value in ICMP Packet Too Big message for 6.1.5 Packet Too Big Forwarding
1.7.5	2005-09-20		Correct the maximum MTU value for 6.1.4 Packet Too Big Transmission.
1.7.4	2005-06-13		Fix typos
1.7.3	2005-06-07		Removed test for Packet Too Big Forwarding (Known Original Host) for SGW.
1.7.2	2005-05-20		Fix typos
1.7.1	2005-05-18		Change Security Policy for 5.3.2.
1.7	2005-05-08		Add Sequence Number Increment Test. Add ICMP Error Test.
1.6	2005-03-01		Change Keys Add Select SPD test for tunnel mode
1.5	2004-11-26		Change packet description of 5.1.4
1.4	2004-11-19		Change Host to End-Node, Default algorithms changed to (3DES-CBC, HMAC-SHA1) for Architecture test. Editorial fix
1.3	2004-09-24		
1.2	2004-09-22		
1.1	2004-09-13		
1.0	2004-09-08		

Copyright