IPv6 Ready Logo

Phase II Test Specification
IPsec

Technical Document

Revision 1.7.6

IPv6 Forum
IPv6 Logo Committee

Modification Record

Version 1.7.6	December 22, 2005 Correct expected MTU value in ICMP Packet Too Big message for 6.1.5 Packet Too Big Forwarding.
Version 1.7.5	September 20, 2005 Correct the maximum MTU value for 6.1.4 Packet Too Big Transmission.
Version 1.7.4	June 13, 2005 Fix typos.
Version 1.7.3	June 7, 2005 Removed test for Packet Too Big Forwarding (Known Original Host) for SGW.
Version 1.7.2	April 20, 2005 Fix typos.
Version 1.7.1	April 18, 2005 Change Security Policy for 5.3.2.
Version 1.7	April 8, 2005 Add Sequence Number Increment Test. Add ICMP Error Test.
Version 1.6	March 1, 2005 Change Keys Add Select SPD test for tunnel mode
Version 1.5	November 26, 2004 Change packet description of 5.1.4
Version 1.4	November 19, 2004 Change Host to End-Node, Default algorithms changed to (3DES-CBC, HMAC-SHA1) for Architecture test. Editorial fix
Version 1.3	September 24, 2004
Version 1.2	September 22, 2004
Version 1.1	September 13, 2004
Version 1.0	September 8, 2004

Acknowledgement

IPv6 Forum would like to acknowledge the efforts of the following organizations in the development of this test specification.

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

Introduction

The IPv6 forum plays a major role in bringing 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.

The IPv6 Logo Program consists of three phases:

Phase 1 :

In a first stage, the Logo will indicate that the product includes IPv6 mandatory core protocols and can interoperate with other IPv6 implementations.

Phase 2 :

The "IPv6 ready" step implies a proper care, technical consensus and clear technical references. The IPv6 ready logo will indicate that a product has successfully satisfied strong requirements stated by the IPv6 Logo Committee (v6LC).

To avoid confusion, the logo "IPv6 Ready" will be generic. The v6LC will define the test profiles with associated requirements for specific functionalities.

Phase 3 :

Same as Phase 2 with IPsec mandated.

Requirements

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

Equipment Type:

We define two possibilities for equipment types, they are as follows:

End-Node:

A node who can use IPsec only for itself. Host and Router can be an End-Node.

SGW (Security Gateway):

A node who can provide IPsec tunnel mode for nodes behind it. Router can be a SGW.

Security Protocol:

A NUT is required to pass all of the ESP tests regardless the equipment type. The IPv6 Ready Logo Program does not focus on AH.

Mode:

The mode requirement depends on the type of NUT.

End-Node:

If the NUT is an End-Node, it must pass all the Transport mode tests. If the NUT supports the Tunnel mode, it also must pass all the Tunnel mode tests. (i.e., Tunnel mode is ADVANCED functionality for End-Node)

SGW:

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

Encryption Algorithm:

IPv6 Logo Committee had defined 2 encryption algorithm categories: BASE ALGORITHM and ADVANCED ALGORITHM. All NUTs must pass the BASE ALGORITHM tests to obtain an IPsec Logo. A NUT which supports algorithms listed as ADVANCED

ALGORITHM, must pass all corresponding tests.

The algorithm requirement is independent from NUT type.

BASE ALGORITHM:
3DES-CBC

ADVANCED ALGORITHM:
AES-CBC
NULL
DES-CBC

Authentication Algorithm:

IPv6 Logo Committee had defined BASE ALGORITHM and ADVANCED ALGORITHM.
All NUTs have to pass all the test of BASE ALGORITHM to obtain the IPsec Logo.
The NUTs, which support the algorithms that are listed as ADVANCED ALGORITHM, have to pass all the corresponding tests.

The algorithm requirement is independent from NUT type.

BASE ALGORITHM:
HMAC-SHA1

ADVANCED ALGORITHM:
AES-XCBC-MAC-96
NULL
HMAC-MD5

Category:

All NUTs are required to support BASIC. ADVANCED is required for all NUTs which support ADVANCED encryption and/or authentication algorithms. Each test description contains a Category section which lists the requirements to satisfy the test.

References

This test specification focus on following IPsec related RFCs.

RFC1829 : The ESP DES-CBC Transform

RFC1851 : The ESP Triple DES Transform

RFC2401 : Security Architecture for the Internet Protocol

RFC2403 : The Use of HMAC-MD5-96 within ESP and AH

RFC2404 : The Use of HMAC-SHA-1-96 within ESP and AH

RFC2405 : The ESP DES-CBC Cipher Algorithm With Explicit IV

RFC2406 : IP Encapsulating Security Payload (ESP)

RFC2410 : The NULL Encryption Algorithm and Its Use With IPsec

RFC2463 : Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)

RFC3602 : The AES-CBC Cipher Algorithm and Its Use with IPsec

RFC3566 : The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec

---TOC---

1. Test Details

This chapter contains detailed information, including terminology, which is described below.

Terminology:

TN : Tester Node
NUT : Node Under Test (Target Implementation)
SGW : Security Gateway

Required Application:

All tests use ICMP Echo Request and Echo Reply messages by default. ICMP is independent from any implemented application and this adds clarity to the test. If the NUT can not apply IPsec for ICMPv6 packets, it is acceptable to use other protocols rather than ICMPv6. In this case, the device must support either ICMPv6, TCP or UDP. The application and port number are unspecified when TCP or UDP packets are used. The test coordinator should support any ports associated with an application used for the test. Applicants must mention the specific protocol and port that was used to execute the tests.

IPsec Configuration:

Manual key configuration is used by default and is a minimal requirement. IKE is an acceptable alternative to use when IPsec is tested. When IKE is used, the encryption key and authentication key are negotiated dynamically. In that case, dynamic keys are used rather than the static keys specified in this document. The tester should support the alternative of using IKE with dynamic keys to execute the tests.

Topology:

In "2. Test Topology" the network topology for the test is shown.

2. Test Topology

Below are logical Network Topologies for test samples.

For End-Node: Transport and Tunnel Mode with End-Node Test

1. Set global address to NUT by RA (NUT_Link0)
2. Set MTU to NUT by RA (MTU value is 1500 for Link0)
3. Make IPsec transport mode between NUT and HOST1 and HOST2

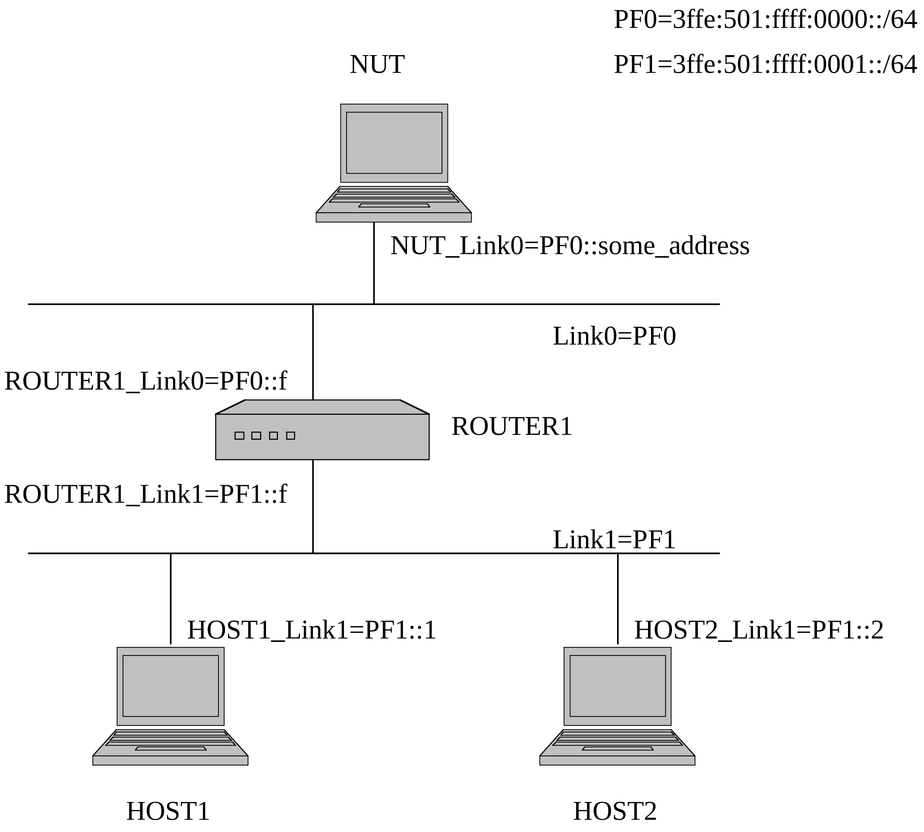

Fig. 1 Topology for End-Node: Transport and Tunnel mode with End-Node

For End-Node: Tunnel Mode with SGW Test

1. Set global address to NUT by RA (NUT_Link0)
2. Set MTU to NUT by RA (MTU value is 1500 for Link0)
3. Make IPsec tunnel mode between NUT and SGW1.

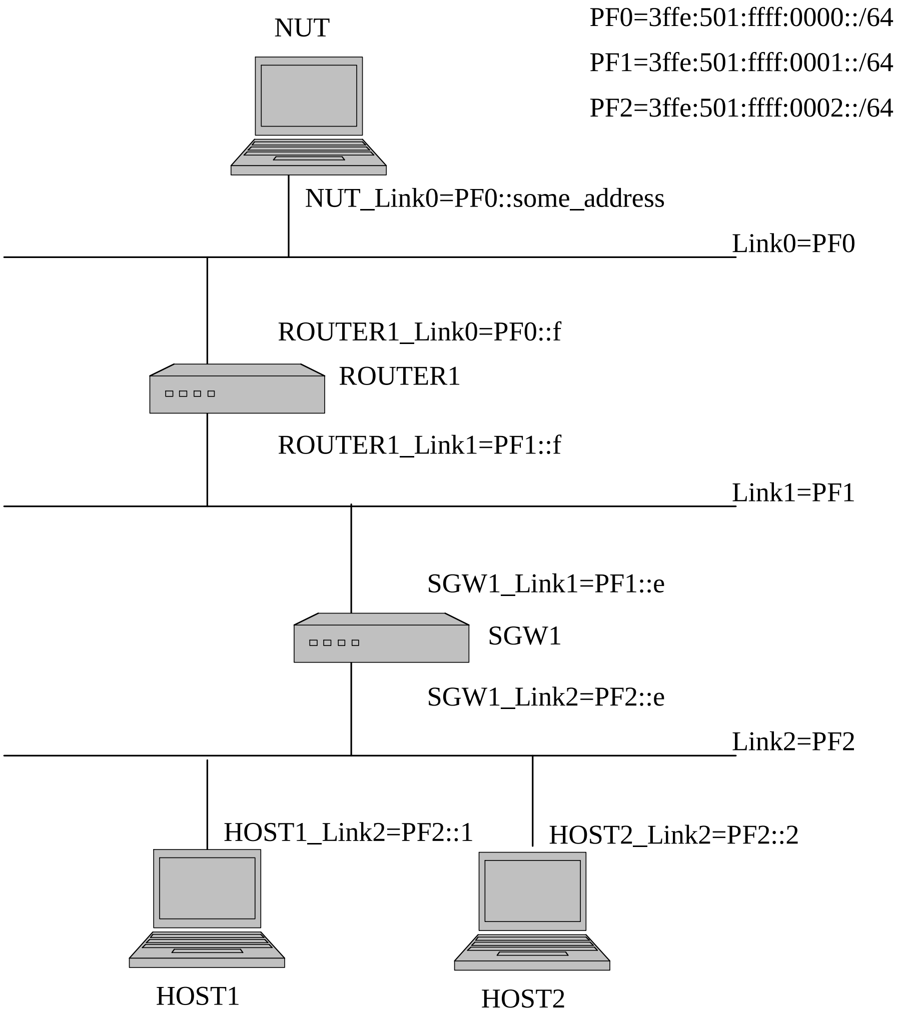

Fig. 2 Topology for End-Node: Tunnel mode with SGW

For SGW: Tunnel Mode with End-Node Test

1. Set global address to NUT manually (NUT_Link0, NUT_Link1)
2. Set routing table to NUT manually (ROUTER1_Link1 for Link2)
3. Set MTU to NUT manually for Link0 and Link1 (MTU value is 1500 for Link0 and Link1)
4. Make IPsec tunnel mode between NUT and HOST2.

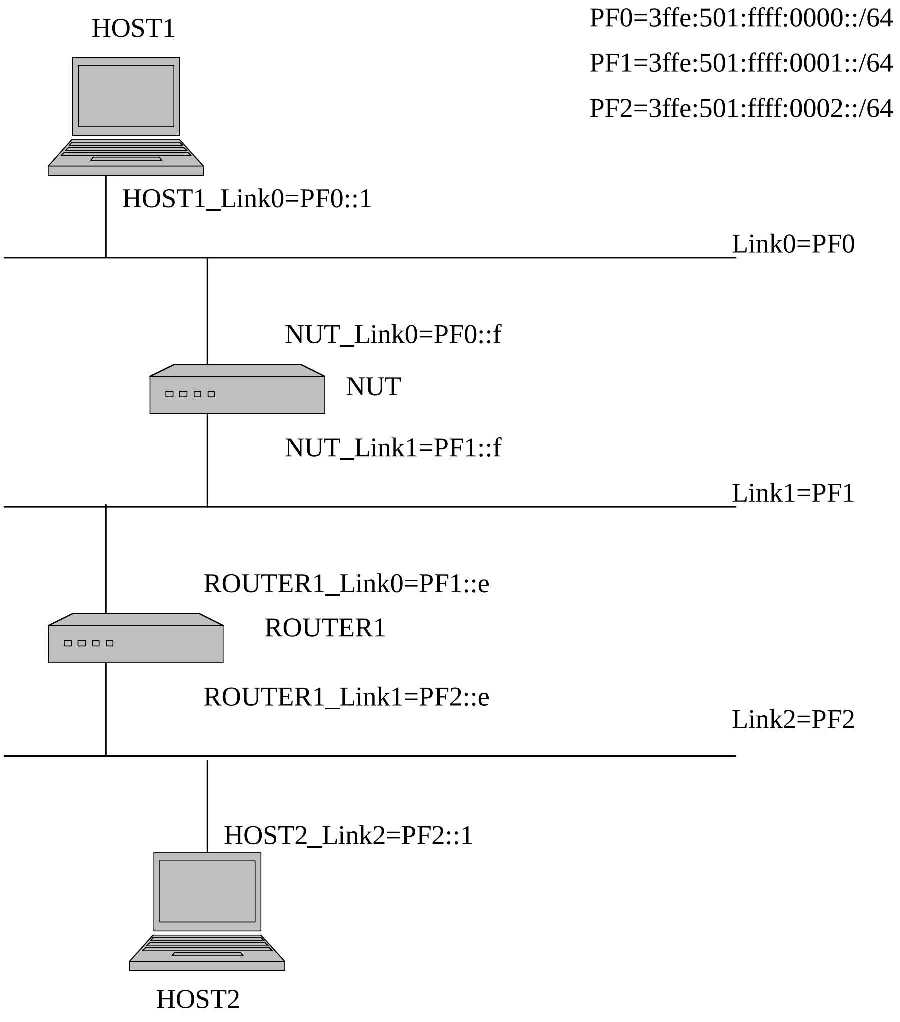

Fig. 3 Topology for SGW: Tunnel mode with End-Node

For SGW: Tunnel Mode Test

1. Set global address to NUT manually (NUT_Link0, NUT_Link1)
2. Set routing table to NUT manually (ROUTER1_Link1 for Link2, Link3 and Link4)
3. Set MTU to NUT manually for Link0 and Link1 (MTU value is 1500 for Link0 and Link1)
4. Make IPsec tunnel mode between NUT and SGW1 and SGW2

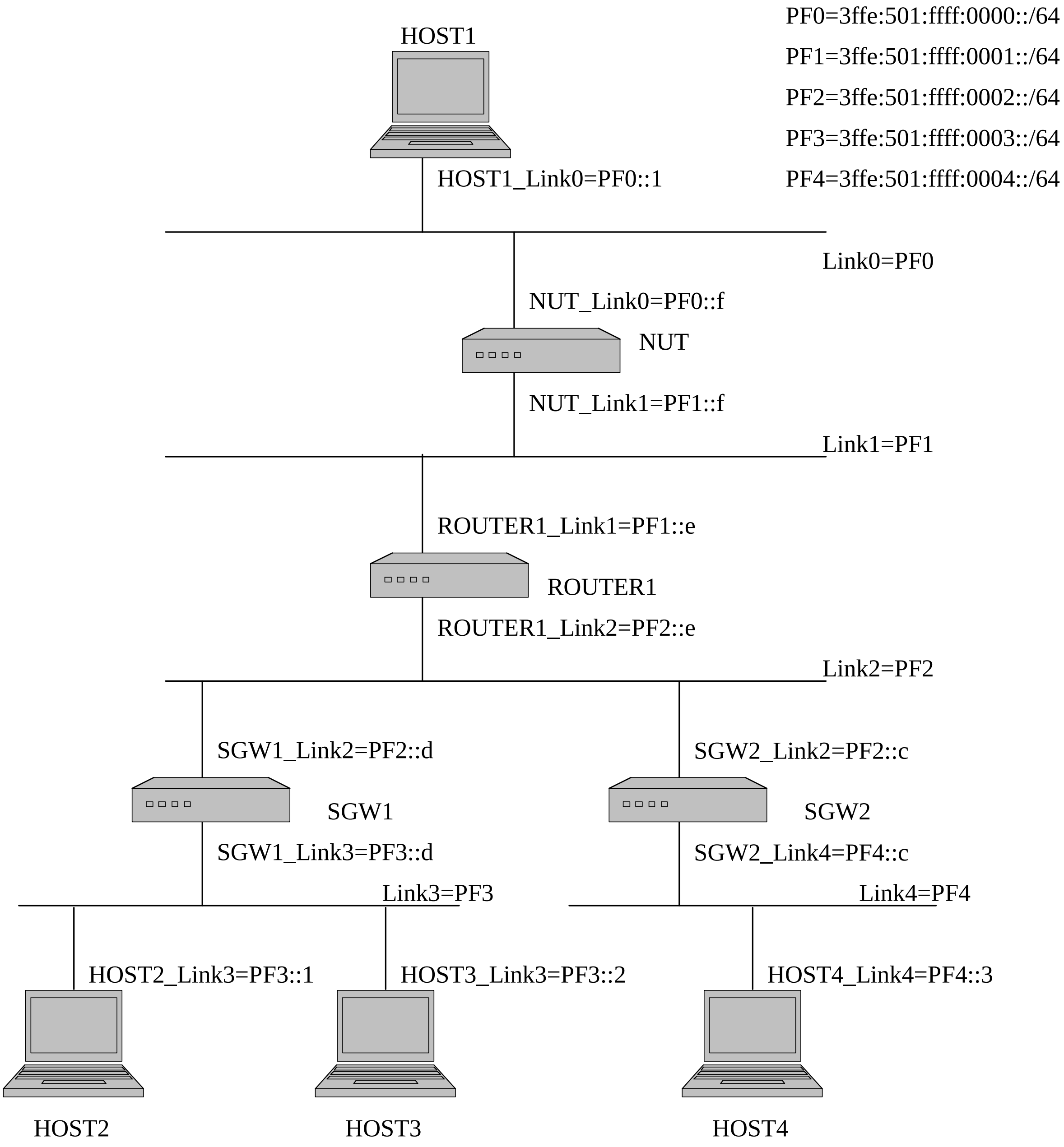

Fig. 4 Topology for SGW: Tunnel mode with SGW

3. Description

Each test specification consists of following parts.

Purpose: The Purpose is the short statement describing what the test attempts to achieve. It is usually phrased as a simple assertion of the future or capability to be tested.

Category: The Category shows what classification of device must satisfy the test.

Initialization: The Initialization 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.

Packets: The Packets describes the simple figure of packets which is used in the test. In this document, the packet name is represented in *Italic* style font.

Procedure: The Procedure describes step-by-step instructions for carrying out the test.

Judgment: The Judgment describes expected result. If we can observe as same result as the description of Judgment, the NUT passes the test.

References: The References section contains some parts of specification related to the tests. It also shows the document names and section numbers.

4. Required Tests

The following table lists which tests a device is required to pass based on category.

For End-Node:

Test Title	Category	Note
Select SPD	BASIC	
Sequence Number Increment	BASIC	
Packet Too Big Reception	BASIC	
Bypass Policy	ADVANCED	Either of Bypass or Discard Policy is required
Discard Policy	ADVANCED	
Transport Mode Padding	BASIC	
Non-Registered SPI	BASIC	
ICV	BASIC	
Transport Mode ESP=3DES-CBC HMAC-SHA1	BASIC	
Transport Mode ESP=3DES-CBC AES-XCBC	ADVANCED	
Transport Mode ESP=3DES-CBC NULL	ADVANCED	
Transport Mode ESP=3DES-CBC HMAC-MD5	ADVANCED	
Transport Mode ESP=AES-CBC (128-bit) HMAC-SHA1	ADVANCED	
Transport Mode ESP=NULL HMAC-SHA1	ADVANCED	
Transport Mode ESP=DES-CBC HMAC-SHA1	ADVANCED	
Tunnel Mode with End-Node	ADVANCED	
Tunnel Mode with SGW	ADVANCED	
Select SPD for 2 Hosts behind 1 SGW	ADVANCED	
Tunnel Mode Padding	ADVANCED	

For SGW:

Test Title	Category	Note
Select SPD	BASIC	
Select SPD for 2 Hosts behind 1 SGW	BASIC	
Sequence Number Increment	BASIC	
Packet Too Big Transmission	BASIC	
Packet Too Big Forwarding(Unknown Original Host)	BASIC	
Bypass Policy	ADVANCED	Either of Bypass or Discard Policy is required
Discard Policy	ADVANCED	
Tunnel Mode Padding	BASIC	
Non-Registered SPI	BASIC	
ICV	BASIC	
Tunnel Mode with End-Node	BASIC	
Tunnel Mode ESP=3DES-CBC HMAC-SHA1	BASIC	
Tunnel Mode ESP=3DES-CBC AES-XCBC	ADVANCED	
Tunnel Mode ESP=3DES-CBC NULL	ADVANCED	
Tunnel Mode ESP=3DES-CBC HMAC-MD5	ADVANCED	
Tunnel Mode ESP=AES-CBC (128-bit) HMAC-SHA1	ADVANCED	
Tunnel Mode ESP=NULL HMAC-SHA1	ADVANCED	
Tunnel Mode ESP=DES-CBC HMAC-SHA1	ADVANCED	

5. End-Node Test

This Chapter describes the test specification for End-Node.
The test specification consists of 2 sections. One is regarding "IPsec Architecture" and another part is regarding "Encryption and Authentication Algorithms".

Architecture

Scope:

Following tests focus on IPsec Architecture.

Overview:

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

5.1.1. Select SPD

Purpose:

Verify that a NUT (End-Node) selects appropriate SPD
(End-Node transport mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : BASIC (A requirement for all End-Node NUTs)
SGW : N/A

Initialization:

Use common topology described as Fig.1

Set NUT's SAD and SPD as following:

```
    HOST1_Link1 --------------------- NUT
                -------------------->  SA1-I
                <-------------------- SA1-O

    HOST2_Link1 --------------------- NUT
                -------------------->  SA2-I
                <-------------------- SA2-O
```

Security Association Database (SAD) for SA1-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA1-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA1-O

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA1-O

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Security Association Database (SAD) for SA2-I

source address	HOST2_Link1
destination address	NUT_Link0
SPI	0x3000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin02
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in02

Security Policy Database (SPD) for SA2-I

source address	HOST2_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA2-O

source address	NUT_Link0
destination address	HOST2_Link1
SPI	0x4000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout2
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out2

Security Policy Database (SPD) for SA2-O

source address	NUT_Link0
destination address	HOST2_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with SA1' s ESP

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with SA1’ s ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
ICMP	Type	129 (Echo Request)

ICMP Echo Request with SA2’ s ESP

IP Header	Source Address	HOST2_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x3000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin02
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in02
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with SA2’ s ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST2_Link1
ESP	SPI	0x4000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout2
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out2
ICMP	Type	129 (Echo Request)

Procedure:

```
HOST2_Link1(TN)  HOST1_Link1(TN)  Target(NUT)
     |                |             |
     |                |------------>| ICMP Echo Request with SA1's ESP
     |                |             |
     |                |<------------| ICMP Echo Reply with SA1's ESP
     |                |             |      (Judgment  #1)
     |                              |
     |----------------------------->| ICMP Echo Request with SA2's ESP
     |                              |
     |<-----------------------------| ICMP Echo Reply with SA2's ESP
     |                              |      (Judgment  #2)
```

PartA: SA1

1. HOST1 sends *"ICMP Echo Request with SA1's ESP"*
2. Observe the packet transmitted by NUT

PartB: SA2

3. Host2 sends *"ICMP Echo Request with SA2's ESP"*
4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
 Step-2: NUT transmits *"ICMP Echo Reply with SA1's ESP"*.

PartB: Judgment #2
 Step-4: NUT transmits *"ICMP Echo Reply with SA2's ESP"*.

References:

RFC2401 : Security Architecture for the Internet Protocol
        Section 4.4.1

5.1.2. Sequence Number Increment

Purpose:

Verify that a NUT (End-Node) increases sequence number correctly, starting with 1. (End-Node transport mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : BASIC (A requirement for all End-Node NUTs)
SGW   : N/A

Initialization:

Use common topology described as Fig.1
Set NUT's SAD and SPD as following:

```
    HOST1_Link1 ------------------- NUT
                ------------------> SA-I
                <------------------ SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-0

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-0

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Sequence	1st = 1, 2nd = 2
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ESP	SPI	0x2000
	Sequence	1st = 1, 2nd = 2
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
ICMP	Type	129 (Echo Reply)

Procedure:

```
HOST1_Link1(TN)                Target(NUT)
      |                              |
      |----------------------------->|  ICMP Echo Request with ESP
      |                              |
      |<-----------------------------|  ICMP Echo Reply with ESP
      |                              |       (Judgment  #1)
      |                              |
      |----------------------------->|  ICMP Echo Request with ESP
      |                              |
      |<-----------------------------|  ICMP Echo Reply with ESP
      |                              |       (Judgment  #2)
```

1. HOST1 sends *"ICMP Echo Request with ESP"*
2. Observe the packet transmitted by NUT
3. HOST1 sends *"ICMP Echo Request with ESP"*
4. Observe the packet transmitted by NUT

Judgment:

Judgment #1
Step-2: NUT transmits an *"ICMP Echo Reply with ESP" with an ESP Sequence Number of 1.*

Judgment #2
Step-4: NUT transmits an *"ICMP Echo Reply with ESP" with an ESP Sequence Number of 2.*

References:

RFC2401 : Security Architecture for the Internet Protocol
RFC2406 : IP Encapsulating Security Payload (ESP), Section 3.3.3

5.1.3. Packet Too Big Reception

Purpose:

Verify that a NUT (End-Node) process the ICMP Error Message (Packet Too Big) correctly.(End-Node transport mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : BASIC (A requirement for all End-Node NUTs)
SGW : N/A

Initialization:

Use common topology described as Fig.1. Router1’s interface to Link1 has an MTU value of 1280.
Set NUT's SAD and SPD as following:

```
    HOST1_Link1 ------------------- NUT
                ------------------> SA-I
                <------------------ SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-0

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-0

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
	Payload Length	1460
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
	Payload Length	1460
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
ICMP	Type	129 (Echo Reply)

ICMP Error Message (Packet Too Big)

IP Header	Source Address	Router_Link1
	Destination Address	NUT_Link0
ICMP	Type	2 (Packet Too Big)
	MTU	1280
	Data	*1232Byte of ICMP Echo Reply with ESP*

Fragmented ICMP Echo Reply with ESP 1

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
	Payload Length	1240
Fragment	Offset	0
	More Flag	1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
ICMP	Type	129 (Echo Reply)

Fragmented ICMP Echo Reply with ESP 2

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
	Payload Length	236
Fragment	Offset	154
	More Flag	0
Data	Data	Rest of *ICMP Echo Reply with ESP*

Procedure:

```
HOST1_Link1(TN)  ROUTER1_Link0  Target(NUT)
     |                 |             |
     |------------------------------>| ICMP Echo Request with ESP
     |                 |             |
     |<------------------------------| ICMP Echo Reply with ESP
     |                 |             |      (Judgment  #1)
     |                 |             |
     |                 |------------>| ICMP Error Message(Packet Too Big)
     |                 |             |
     |------------------------------>| ICMP Echo Request with ESP
     |                 |             |
     |<------------------------------| Fragmented ICMP Echo Reply with ESP 1
     |<------------------------------| Fragmented ICMP Echo Reply with ESP 2
     |                 |             |      (Judgment  #2)
```

1. HOST1 sends *"ICMP Echo Request with ESP"*
2. Observe the packet transmitted by NUT
3. ROUTER1 sends *"ICMP Error Message (Packet Too Big)*
4. HOST1 sends *"ICMP Echo Request with ESP"*
5. Observe the packet transmitted by NUT

Judgment:

Judgment #1
Step-2: NUT transmits *"ICMP Echo Reply with ESP"*

Judgment #2
Step-5: NUT transmits *"Fragmented ICMP Echo Reply with ESP 1"*
and *"Fragmented ICMP Echo Reply with ESP 2"*

References:

RFC2401 : Security Architecture for the Internet Protocol
Section 6.1.2

5.1.4. Bypass Policy

Purpose:

Verify that a NUT (End-Node) select bypass or discard policies

Category:

End-Node : ADVANCED (This test is required for all End-Node NUTs which support Bypass Policy, regardless of explicitly or implicitly)
SGW : N/A

NOTE: NUT needs to pass at least either of "Bypass Policy" or "Discard Policy" tests.

Initialization:

Use common topology described as Fig.1
Set NUT's SAD and SPD as following:

```
    HOST1_Link1 -------------------- NUT
                -------------------> SA-I
                <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-0

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-0

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
	Payload Length	1460
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
	Payload Length	1460
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
ICMP	Type	129 (Echo Reply)

ICMP Echo Request

IP Header	Source Address	HOST2_Link1
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply

IP Header	Source Address	NUT_Link0
	Destination Address	HOST2_Link1
ICMP	Type	129 (Echo Reply)

Procedure:

```
HOST2_Link1(TN)      HOST1_Link1(TN)       Target(NUT)
     |                     |                     |
     |                     |-------------------->| ICMP Echo Request with ESP
     |                     |                     |
     |                     |<--------------------| ICMP Echo Reply with ESP
     |                     |                     |      (Judgment  #1)
     |                     |                     |
   ============================================== Set Bypass policy to NUT.
     |                     |                     |
     |------------------------------------------>| ICMP Echo Request
     |                     |                     |
     |←------------------------------------------| ICMP Echo Reply
     |                     |                     |      (Judgment  #2)
     |                     |                     |
```

PartA: Confirmation

1. Host1 sends *"ICMP Echo Request with ESP"*
2. Observe the packet transmitted by NUT

PartB: Bypass policy

3. Set Bypass policy for above ICMP Echo Request to NUT as following example

Example 1: Security Policy Database (SPD) for policy=Bypass

source address	HOST2_Link1
destination address	NUT_Link0
upper spec	any
direction	in
policy	bypass(none)

Example 2: Security Policy Database (SPD) for policy=Bypass as default policy

source address	any
destination address	any
upper spec	any
direction	in
policy	bypass(none)

4. HOST1 sends *"ICMP Echo Request"*
5. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1.
Step-2: NUT transmits *"ICMP Echo Reply with ESP"*

PartB: Judgment #2.
Step-5: NUT transmits *"ICMP Echo Reply"*

References:

RFC2401 : Security Architecture for the Internet Protocol
Section 4.4.1

5.1.5. Discard Policy

Purpose:

Verify that a NUT (End-Node) select bypass or discard policies

Category:

End-Node : ADVANCED (This test is required for all End-Node NUTs which support
           Discard Policy, regardless of explicitly or implicitly)
SGW   : N/A

NOTE: NUT need to pass at least either of "Bypass Policy" or "Discard Policy" tests.

Initialization:

Use common topology described as Fig.1
Set NUT's SAD and SPD as following:

```
    HOST1_Link1 -------------------- NUT
                --------------------> SA-I
                <-------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
	Payload Length	1460
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
	Payload Length	1460
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
ICMP	Type	129 (Echo Reply)

ICMP Echo Request

IP Header	Source Address	HOST2_Link1
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply

IP Header	Source Address	NUT_Link0
	Destination Address	HOST2_Link1
ICMP	Type	129 (Echo Reply)

Procedure:

```
HOST2_Link1(TN)       HOST1_Link1(TN)        Target(NUT)
     |                      |                       |
     |                      |--------------------->| ICMP Echo Request with ESP
     |                      |                       |
     |                      |<---------------------| ICMP Echo Reply with ESP
     |                      |                       |      (Judgment  #1)
     |                      |                       |
   ============================================== Set Discard policy to NUT.
     |                      |                       |
     |                      |                       |
     |------------------------------------------->| ICMP Echo Request
     |                      |                       |
     |(<-----------------------------------------)| No Reply
     |                      |                       |      (Judgment  #2)
     |                      |                       |
```

PartA: Confirmation

1. Host1 sends *"ICMP Echo Request with ESP"*

2. Observe the packet transmitted by NUT

PartB: Discard policy

3. Set Discard policy for above ICMP Echo Request to NUT as following example

Example 1: Security Policy Database (SPD) for policy=Discard

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
policy	discard

Example 2: Security Policy Database (SPD) for policy=Discard as default policy

source address	any
destination address	any
upper spec	any
direction	in
policy	discard

4. HOST1 sends *"ICMP Echo Request"*
5. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1.

Step-2: NUT transmits *"ICMP Echo Reply with ESP"*

PartB: Judgment #2.

Step-5: NUT transmits nothing

References:

RFC2401 : Security Architecture for the Internet Protocol
Section 4.4.1

5.1.6. Transport Mode Padding

Purpose:

Verify that a NUT (End-Node) supports padding & padding byte handling
(End-Node transport mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : BASIC (A requirement for all End-Node NUTs)
SGW : N/A

Initialization:

Use common topology described as Fig.1

Set NUT's SAD and SPD as following:

```
    HOST1_Link1 ------------------- NUT
                ------------------> SA-I
                <------------------ SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP 1

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
	Padding	Sequential
	Padding Length	7
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request with ESP 2

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
	Padding	Sequential
	Padding Length	255
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
	Padding Length	7+8n (0 <= n <= 31)
ICMP	Type	129 (Echo Reply)
	Data Length	7

Procedure:

```
 HOST1_Link1(TN)                  Target(NUT)
      |                            |
      |--------------------------->| ICMP Echo Request with ESP 1
      |                            |
      |<---------------------------| ICMP Echo Reply with ESP
      |                            |      (Judgment  #1)
      |                            |
      |                            |
      |--------------------------->| ICMP Echo Request with ESP 2
      |                            |
      |<---------------------------| ICMP Echo Reply with ESP
      |                            |      (Judgment  #2)
```

PartA: Padding 7

1. HOST1 sends *"ICMP Echo Request with ESP 1"*
2. Observe the packet transmitted by NUT

PartB: Padding 255

3. HOST1 sends *"ICMP Echo Request with ESP 2"*
4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
Step-2: NUT transmits *"ICMP Echo Reply with ESP"*

PartB: Judgment #2
Step-4: NUT transmits *"ICMP Echo Reply with ESP"*

References:

RFC2406 : IP Encapsulating Security Payload (ESP)
Section 2.4

5.1.7. Non-Registered SPI

Purpose:

Verify that a NUT (End-Node) can behave when No valid Security Association is configured.

Category:

End-Node : BASIC (A requirement for all End-Node NUTs)
SGW : N/A

Initialization:

Use common topology described as Fig.1

Set NUT's SAD and SPD as following:

```
    HOST1_Link1 -------------------- NUT
                -------------------> SA-I
                <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP 1

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with ESP 1

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
ICMP	Type	129 (Echo Reply)

ICMP Echo Request with ESP 2 with non-registered SPI

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x9000 (Different from SA-I's SPD)
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMP	Type	128 (Echo Request)

ICMP Echo Reply

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ICMP	Type	129 (Echo Reply)

Procedure:

```
HOST1_Link1(TN)       Target(NUT)
     |                    |
     |------------------->| ICMP Echo Request with ESP 1
     |                    |
     |<-------------------| ICMP Echo Reply with ESP 1
     |                    |      (Judgment  #1)
     |                    |
     |                    |
     |------------------->| ICMP Echo Request with ESP 2 with non-registered SPI
     |                    |      (different SPI)
     |                    |
     |(<-----------------)| No ICMP Echo Reply
     |                    |      (Judgment  #2)
```

PartA: valid SA exists
 1. HOST1 sends *"ICMP Echo Request with ESP 1"*
 2. Observe the packet transmitted by NUT

PartB: no valid SA exists
 3. HOST1 sends *"ICMP Echo Request with ESP 2"*(different SPI)
 4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
 Step-2: NUT transmits *"ICMP Echo Reply with ESP 1"*

PartB: Judgment #2
 Step-4: NUT does not transmit *"ICMP Echo Reply with ESP 1" nor "ICMP Echo Reply"*

References:

RFC2406 : IP Encapsulating Security Payload (ESP)
 Section 3.4.2

5.1.8. ICV

Purpose:

Verify that a NUT (End-Node) can detect the modification by examining the ICV (End-Node transport mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : BASIC (A requirement for all End-Node NUTs)
SGW : N/A

Initialization:

Use common topology described as Fig.1

Set NUT's SAD and SPD as following:

```
    HOST1_Link1 -------------------- NUT
                -------------------> SA-I
                <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP 1

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Sequence number	1
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMP	Type	128 (Echo Request)
	Data	″EchoData″

ICMP Echo Reply with ESP 1

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
ICMP	Type	129 (Echo Reply)
	Data	"EchoData"

ICMP Echo Request with ESP 2

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Sequence number	2
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
	ICV	aaaaaaaaaaaaaaaaaa……
ICMP	Type	128 (Echo Request)
	Data	"cracked" ("EchoData" is original)

ICMP Echo Reply

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ICMP	Type	129 (Echo Reply)

Procedure:

```
HOST1_Link1(TN)                Target(NUT)
     |                           |
     |-------------------------->| ICMP Echo Request with ESP 1
     |                           |
     |<--------------------------| ICMP Echo Reply with ESP 1
     |                           |      (Judgment  #1)
     |                           |
     |-------------------------->| ICMP Echo Request with ESP 2
     |                           |      (ICV is modified)
     |                           |
     |(<------------------------)| No ICMP Echo Reply
     |                           |      (Judgment  #2)
```

PartA: send correct packet

1. HOST1 sends *"ICMP Echo Request with ESP 1"*
2. Observe the packet transmitted by NUT

PartB: send modified packet

3. HOST1 sends *"ICMP Echo Request with ESP 2"* (ICV is modified)
4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1

Step-2: NUT transmits *"ICMP Echo Reply with ESP 1"*

PartB: Judgment #2

Step-4: NUT does not transmit *"ICMP Echo Reply with ESP 1" nor "ICMP Echo Reply"*

References:

RFC2406 : IP Encapsulating Security Payload (ESP)
Section 3.4.3

5.2. Algorithm Test

Scope:

Following tests focus on Encryption and Authentication Algorithms.

Overview:

Tests in this section verify that the NUT properly decrypt the received packets and encrypts the transmitting packets using Encryption algorithms specified in the SAD.
And they verify that the NUT properly processes the authentication algorithms specified in the SAD.

5.2.1. Transport Mode ESP=3DES-CBC HMAC-SHA1

Purpose:

End-Node transport mode, ESP=3DES-CBC HMAC-SHA1

Category:

End-Node : BASIC (A requirement for all End-Node NUTs)
SGW   : N/A

Initialization:

Use common topology described as Fig.1

Set NUT's SAD and SPD as following:

```
    HOST1_Link1 -------------------- NUT
                -------------------> SA-I
                <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-0

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-0

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
ICMP	Type	129 (Echo Reply)

Procedure:

```
HOST1_Link1(TN)              Target(NUT)
     |                          |
     |------------------------->| ICMP Echo Request with ESP
     |                          |
     |<-------------------------| ICMP Echo Reply with ESP
     |                          |     (Judgment  #1)
```

1. HOST1 sends *"ICMP Echo Request with ESP"*
2. Observe the packet transmitted by NUT

Judgment:

Judgment #1
Step-2: NUT transmits *"ICMP Echo Reply with ESP"*

References:

RFC1851 : The ESP Triple DES Transform
RFC2401 : Security Architecture for the Internet Protocol
RFC2404 : The Use of HMAC-SHA-1-96 within ESP and AH
RFC2405 : The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC2406 : IP Encapsulating Security Payload (ESP)

5.2.2. Transport Mode ESP=3DES-CBC AES-XCBC

Purpose:

End-Node transport mode, ESP=3DES-CBC AES-XCBC

Category:

End-Node : ADVANCED (This test is required for all End-Node NUTs which support
AES-XCBC as an authentication algorithm)
SGW : N/A

Initialization:

Use common topology described as Fig.1

Set NUT's SAD and SPD as following:

```
    HOST1_Link1 -------------------- NUT
                -------------------> SA-I
                <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	AES-XCBC-MAC-96
ESP authentication key	ipv6readaesxin01

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	AES-XCBC-MAC-96
ESP authentication key	ipv6readaesxout1

Security Policy Database (SPD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcin01
	Authentication Algorithm	AES-XCBC-MAC-96
	Authentication Key	ipv6readaesxin01
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcout1
	Authentication Algorithm	AES-XCBC-MAC-96
	Authentication Key	ipv6readaesxout1
ICMP	Type	129 (Echo Reply)

Procedure:

```
HOST1_Link1(TN)               Target(NUT)
     |                           |
     |-------------------------->| ICMP Echo Request with ESP
     |                           |
     |<--------------------------| ICMP Echo Reply with ESP
     |                           |      (Judgment  #1)
```

1. HOST1 sends *"ICMP Echo Request with ESP"*
2. Observe the packet transmitted by NUT

Judgment:

Judgment #1
Step-2: NUT transmits *"ICMP Echo Reply with ESP"*

References:

RFC1851 : The ESP Triple DES Transform
RFC2401 : Security Architecture for the Internet Protocol
RFC2405 : The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC2406 : IP Encapsulating Security Payload (ESP)
RFC3566 : The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec

5.2.3. Transport Mode ESP=3DES-CBC NULL

Purpose:

End-Node transport mode, ESP=3DES-CBC NULL

Category:

End-Node : ADVANCED (This test is required for all End-Node NUTs which support
           NULL as an authentication algorithm)
SGW   : N/A

Initialization:

Use common topology described as Fig.1

Set NUT's SAD and SPD as following:

```
    HOST1_Link1 -------------------- NUT
                -------------------> SA-I
                <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	NULL
ESP authentication key	

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	NULL
ESP authentication key	

Security Policy Database (SPD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	NULL
	Authentication Key	
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	NULL
	Authentication Key	
ICMP	Type	129 (Echo Reply)

Procedure:

```
HOST1_Link1(TN)              Target(NUT)
     |                           |
     |-------------------------->|  ICMP Echo Request with ESP
     |                           |
     |<--------------------------|  ICMP Echo Reply with ESP
     |                           |       (Judgment  #1)
```

1. HOST1 sends *"ICMP Echo Request with ESP"*
2. Observe the packet transmitted by NUT

Judgment:

Judgment #1
Step-2: NUT transmits *"ICMP Echo Reply with ESP"*

References:

RFC1851 : The ESP Triple DES Transform
RFC2401 : Security Architecture for the Internet Protocol
RFC2405 : The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC2406 : IP Encapsulating Security Payload (ESP)
RFC2410 : The NULL Encryption Algorithm and Its Use With IPsec

5.2.4. Transport Mode ESP=3DES-CBC HMAC-MD5

Purpose:

End-Node transport mode, ESP=3DES-CBC HMAC-MD5

Category:

End-Node : ADVANCED (This test is required for all End-Node NUTs which support
HMAC-MD5 as an authentication algorithm)
SGW : N/A

Initialization:

Use common topology described as Fig.1

Set NUT's SAD and SPD as following:

```
    HOST1_Link1 -------------------- NUT
                -------------------> SA-I
                <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-MD5
ESP authentication key	ipv6readymd5in01

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	HMAC-MD5
ESP authentication key	ipv6readymd5out1

Security Policy Database (SPD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-MD5
	Authentication Key	ipv6readymd5in01
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	KEY	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-MD5
	Authentication Key	ipv6readymd5out1
ICMP	Type	129 (Echo Reply)

Procedure:

```
HOST1_Link1(TN)                Target(NUT)
     |                           |
     |-------------------------->|  ICMP Echo Request with ESP
     |                           |
     |<--------------------------|  ICMP Echo Reply with ESP
     |                           |       (Judgment  #1)
```

1. HOST1 sends *"ICMP Echo Request with ESP"*
2. Observe the packet transmitted by NUT

Judgment:

Judgment #1
Step-2: NUT transmits *"ICMP Echo Reply with ESP"*

References:

RFC1851 : The ESP Triple DES Transform
RFC2401 : Security Architecture for the Internet Protocol
RFC2403 : The Use of HMAC-MD5-96 within ESP and AH
RFC2405 : The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC2406 : IP Encapsulating Security Payload (ESP)

5.2.5. Transport Mode ESP=AES-CBC (128-bit) HMAC-SHA1

Purpose:

End-Node transport mode, ESP=AES-CBC (128-bit) HMAC-SHA1

Category:

End-Node : ADVANCED (This test is required for all End-Node NUTs which support
AES-CBC (128-bit) as an encryption algorithm)
SGW : N/A

Initialization:

Use common topology described as Fig.1

Set NUT’s SAD and SPD as following:

```
    HOST1_Link1 -------------------- NUT
                -------------------> SA-I
                <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	AES-CBC(128-bit)
ESP algorithm key	ipv6readaescin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	AES-CBC(128-bit)
ESP algorithm key	ipv6readaescout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	AES-CBC(128-bit)
	Key	ipv6readaescin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ESP	SPI	0x2000
	Algorithm	AES-CBC(128-bit)
	Key	ipv6readaescout1
	Authentication Algorithm	HMAC-MD5
	Authentication Key	ipv6readylogsha1out1
ICMP	Type	129 (Echo Reply)

Procedure:

```
HOST1_Link1(TN)                Target(NUT)
     |                              |
     |----------------------------->| ICMP Echo Request with ESP
     |                              |
     |<-----------------------------| ICMP Echo Reply with ESP
     |                              |      (Judgment  #1)
```

1. HOST1 sends *"ICMP Echo Request with ESP"*
2. Observe the packet transmitted by NUT

Judgment:

Judgment #1
Step-2: NUT transmits *"ICMP Echo Reply with ESP"*

References:

RFC2401 : Security Architecture for the Internet Protocol
RFC2404 : The Use of HMAC-SHA-1-96 within ESP and AH
RFC2406 : IP Encapsulating Security Payload (ESP)
RFC3602 : The AES-CBC Cipher Algorithm and Its Use with IPsec

5.2.6. Transport Mode ESP=NULL HMAC-SHA1

Purpose:

End-Node transport mode, ESP=NULL HMAC-SHA1

Category:

End-Node : ADVANCED (This test is required for all End-Node NUTs which support
NULL as an encryption algorithm)
SGW : N/A

Initialization:

Use common topology described as Fig.1

Set NUT' s SAD and SPD as following:

```
    HOST1_Link1 -------------------- NUT
                -------------------> SA-I
                <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	NULL
ESP algorithm key	
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	NULL
ESP algorithm key	
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	NULL
	Key	
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ESP	SPI	0x2000
	Algorithm	NULL
	Key	
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
ICMP	Type	129 (Echo Reply)

Procedure:

```
HOST1_Link1(TN)              Target(NUT)
     |                            |
     |--------------------------->| ICMP Echo Request with ESP
     |                            |
     |<---------------------------| ICMP Echo Reply with ESP
     |                            |      (Judgment  #1)
```

1. HOST1 sends *"ICMP Echo Request with ESP"*
2. Observe the packet transmitted by NUT

Judgment:

Judgment #1
 Step-2: NUT transmits *"ICMP Echo Reply with ESP"*

References:

RFC2401 : Security Architecture for the Internet Protocol
RFC2404 : The Use of HMAC-SHA-1-96 within ESP and AH
RFC2406 : IP Encapsulating Security Payload (ESP)
RFC2410 : The NULL Encryption Algorithm and Its Use With IPsec

5.2.7. Transport Mode ESP=DES-CBC HMAC-SHA1

Purpose:

End-Node transport mode, ESP=DES-CBC HMAC-SHA1

Category:

End-Node : ADVANCED (This test is required for all End-Node NUTs which support DES-CBC as an encryption algorithm)
SGW : N/A

Initialization:

Use common topology described as Fig.1

Set NUT’s SAD and SPD as following:

```
    HOST1_Link1 -------------------- NUT
                -------------------> SA-I
                <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	transport
protocol	ESP
ESP algorithm	DES-CBC
ESP algorithm key	idesin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	transport

Security Association Database (SAD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	transport
protocol	ESP
ESP algorithm	DES-CBC
ESP algorithm key	idesout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	transport

Packets:

ICMP Echo Request with ESP

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	DES-CBC
	Key	idesin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
ICMP	Type	128 (Echo Request)

ICMP Echo Reply with ESP

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ESP	SPI	0x2000
	Algorithm	DES-CBC
	Key	idesout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
ICMP	Type	129 (Echo Reply)

Procedure:

```
HOST1_Link1(TN)                Target(NUT)
      |                             |
      |---------------------------->| ICMP Echo Request with ESP
      |                             |
      |<----------------------------| ICMP Echo Reply with ESP
      |                             |      (Judgment  #1)
```

1. HOST1 sends *"ICMP Echo Request with ESP"*
2. Observe the packet transmitted by NUT

Judgment:

Judgment #1
Step-2: NUT transmits *"ICMP Echo Reply with ESP"*

References:

RFC1829 : The ESP DES-CBC Transform
RFC2401 : Security Architecture for the Internet Protocol
RFC2404 : The Use of HMAC-SHA-1-96 within ESP and AH
RFC2405 : The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC2406 : IP Encapsulating Security Payload (ESP)

5.3. Tunnel Mode

5.3.1. Tunnel Mode with End-Node

Purpose:

Verify that a NUT (End-Node) can build IPsec tunnel mode with End-Node correctly.
(End-Node tunnel mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : ADVANCED (This test is required for all End-Node NUTs which support Tunnel Mode)
SGW : N/A

Initialization:

Use common topology described as Fig.1
Set NUT's SAD and SPD as following:

```
   HOST1_Link1 -------------------- NUT
               -------------------> SA-I
               <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

tunnel source address	HOST1_Link1
tunnel destination address	NUT_Link0
source address	HOST1_Link1
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link0
destination address	HOST1_Link1
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

tunnel source address	NUT_Link0
tunnel destination address	HOST1_Link1
source address	NUT_Link0
destination address	HOST1_Link1
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP tunnel

IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST1_Link1
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply within ESP tunnel

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link1
ICMP	Type	129 (Echo Reply)

Procedure:

```
 HOST1_Link1(TN)        Target(NUT)
     |                       |
     |---------------------->| ICMP Echo Request within ESP tunnel
     |                       |
     |<----------------------| ICMP Echo Reply within ESP tunnel
     |                       |      (Judgment  #1)
```

1. HOST1 sends *"ICMP Echo Request with ESP tunnel"*
2. Observe the packet transmitted by NUT

Judgment:

Judgment #1
Step2: NUT transmits the packet *"ICMP Echo Reply within ESP tunnel"*.

References:

RFC2401 : Security Architecture for the Internet Protocol
Section 4.1

5.3.2. Tunnel Mode with SGW

Purpose:

Verify that a NUT (End-Node) can build IPsec tunnel mode with SGW correctly (End-Node tunnel mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : ADVANCED (This test is required for all End-Node NUTs which support
           Tunnel Mode)
SGW   : N/A

Initialization:

Use common topology described as Fig.2
Set NUT's SAD and SPD as following:

```
   HOST1 -- SGW1 -------------------- NUT
                 -------------------> SA-I
                 <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

tunnel source address	SGW1_Link1
tunnel destination address	NUT_Link0
source address	Link2
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link0
destination address	SGW1_Link1
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

tunnel source address	NUT_Link0
tunnel destination address	SGW1_Link1
source address	NUT_Link0
destination address	Link2
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP tunnel

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply within ESP tunnel

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link2
ICMP	Type	129 (Echo Reply)

Procedure:

```
 SGW1_Link1(TN)        Target(NUT)
     |                      |
     |--------------------->| ICMP Echo Request from HOST1 within ESP tunnel
     |                      |
     |<---------------------| ICMP Echo Reply to HOST1 within ESP tunnel
     |                      |      (Judgment  #1)
```

1. SGW1 sends *"ICMP Echo Request from HOST1 within ESP tunnel"*
2. Observe the packet transmitted by NUT

Judgment:

Judgment #1

Step2: NUT transmits the packet *"ICMP Echo Reply within ESP tunnel"*.

References:

RFC2401 : Security Architecture for the Internet Protocol
Section 4.1

5.3.3. Select SPD for 2 Hosts behind 1 SGW

Purpose:

Verify that a NUT (End-Node) can build IPsec tunnel mode with SGW correctly
(End-Node tunnel mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : ADVANCED (This test is required for all End-Node NUTs which support
Tunnel Mode)
SGW : N/A

Initialization:

Use common topology described as Fig.2
Set NUT's SAD and SPD as following:

```
  HOST1_Link2 -- SGW1 -------------------- NUT_Link0
                      -------------------> SA1-I
                      <------------------- SA1-O

  HOST2_Link2 -- SGW1 -------------------- NUT_Link0
                      -------------------> SA2-I
                      <------------------- SA2-O
```

Security Association Database (SAD) for SA1-I

source address	SGW1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA1-I

tunnel source address	SGW1_Link1
tunnel destination address	NUT_Link0
source address	HOST1_Link2
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA1-O

source address	NUT_Link0
destination address	SGW1_Link1
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA1-O

tunnel source address	NUT_Link0
tunnel destination address	SGW1_Link1
source address	NUT_Link0
destination address	HOST1_Link2
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA2-I

source address	SGW1_Link1
destination address	NUT_Link0
SPI	0x3000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin02
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in02

Security Policy Database (SPD) for SA2-I

tunnel source address	SGW1_Link1
tunnel destination address	NUT_Link0
source address	HOST2_Link2
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA2-O

source address	NUT_Link0
destination address	SGW1_Link1
SPI	0x4000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout2
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out2

Security Policy Database (SPD) for SA2-O

tunnel source address	NUT_Link0
tunnel destination address	SGW1_Link1
source address	NUT_Link0
destination address	HOST2_Link2
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request from HOST1 within ESP tunnel

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply to HOST1 within ESP tunnel

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Request from HOST2 within ESP tunnel

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x3000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin02
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in02
IP Header	Source Address	HOST2_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply to HOST2 within ESP tunnel

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	0x4000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout2
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out2
IP Header	Source Address	NUT_Link0
	Destination Address	HOST2_Link2
ICMP	Type	129 (Echo Reply)

Procedure:

```
 SGW1_Link1(TN)         Target(NUT)
     |                       |
     |---------------------->| ICMP Echo Request from HOST1 within ESP tunnel
     |                       |
     |<----------------------| ICMP Echo Reply to HOST1 within ESP tunnel
     |                       |      (Judgment  #1)
     |                       |
     |---------------------->| ICMP Echo Request from HOST2 within ESP tunnel
     |                       |
     |<----------------------| ICMP Echo Reply to HOST2 within ESP tunnel
     |                       |      (Judgment  #2)
```

PartA: SA1

1. SGW1 sends *"ICMP Echo Request from HOST1 within ESP tunnel"*
2. Observe the packet transmitted by NUT

PartB: SA2

3. SGW1 sends *"ICMP Echo Request from HOST2 within ESP tunnel"*
4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
Step-2: NUT transmits the packet *"ICMP Echo Reply to HOST1 within ESP tunnel"*.

PartB: Judgment #2
Step-4: NUT transmits the packet *"ICMP Echo Reply to HOST2 within ESP tunnel"*.

References:

RFC2401 : Security Architecture for the Internet Protocol
    Section 4.1

5.3.4. Tunnel Mode Padding

Purpose:

Verify that a NUT (End-Node) supports padding & padding byte handling
(End-Node Tunnel mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : BASIC (A requirement for all End-Node NUTs)
SGW : N/A

Initialization:

Use common topology described as Fig.2
Set NUT's SAD and SPD as following:

```
   HOST1 -- SGW1 -------------------- NUT
                 -------------------> SA-I
                 <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link1
destination address	NUT_Link0
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

tunnel source address	SGW1_Link1
tunnel destination address	NUT_Link0
source address	Link2
destination address	NUT_Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link0
destination address	SGW1_Link1
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

tunnel source address	NUT_Link0
tunnel destination address	SGW1_Link1
source address	NUT_Link0
destination address	Link2
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP tunnel 1

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
	Padding	sequential
	Padding Length	7
IP Header	Source Address	HOST1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request within ESP tunnel 2

IP Header	Source Address	SGW1_Link1
	Destination Address	NUT_Link0
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
	Padding	sequential
	Padding Length	255
IP Header	Source Address	HOST1_Link2
	Destination Address	NUT_Link0
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Reply within ESP tunnel

IP Header	Source Address	NUT_Link0
	Destination Address	SGW1_Link1
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
	Padding Length	7+8n (0 <= n <= 31)
IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link2
ICMP	Type	129 (Echo Reply)
	Data Length	7

Procedure:

```
 SGW1_Link1(TN)              Target(NUT)
       |                        |
       |----------------------->| ICMP Echo Request from HOST1 within ESP tunnel
       |                        |
       |<-----------------------| ICMP Echo Reply within ESP
       |                        |      (Judgment  #1)
       |                        |
       |                        |
       |----------------------->| ICMP Echo Request from HOST1 within ESP tunnel
       |                        |
       |<-----------------------| ICMP Echo Reply within ESP
       |                        |      (Judgment  #2)
```

PartA: Padding 7

1. SGW1 sends *"ICMP Echo Request from HOST1 within ESP tunnel"*
2. Observe the packet transmitted by NUT

PartB: Padding 255

3. SGW1 sends *"ICMP Echo Request from HOST1 within ESP tunnel"*
4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1

Step-2: NUT transmits the packet *"ICMP Echo Reply to HOST1 within ESP tunnel"*.

PartB: Judgment　#2
 Step-4: NUT transmits the packet *"ICMP Echo Reply to HOST1 within ESP tunnel".*

References:

RFC2406 : IP Encapsulating Security Payload
　　Section 2.4

6. SGW Test

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

6.1. Architecture

Scope:

Following tests focus on IPsec Architecture.

Overview:

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

6.1.1. Select SPD

Purpose:

Verify that a NUT (SGW) selects appropriate SPD
(SGW tunnel mode, ESP=3DES-CBC)

Category:

End-Node : N/A
SGW : BASIC (A requirement for all SGW NUTs)

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
  HOST2_Link3 -- SGW1 -------------------- NUT -- HOST1_Link0
                      -------------------> SA1-I
                      <------------------- SA1-O

  HOST4_Link4 -- SGW2 -------------------- NUT -- HOST1_Link0
                      -------------------> SA2-I
                      <------------------- SA2-O
```

Security Association Database (SAD) for SA1-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA1-I

tunnel source address	SGW1_Link2
tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA1-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA1-O

tunnel source address	NUT_Link1
tunnel destination address	SGW1_Link2
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA2-I

source address	SGW2_Link2
destination address	NUT_Link1
SPI	0x3000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin02
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in02

Security Policy Database (SPD) for SA2-I

tunnel source address	SGW2_Link2
tunnel destination address	NUT_Link1
source address	Link4
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA2-O

source address	NUT_Link1
destination address	SGW2_Link2
SPI	0x4000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout2
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out2

Security Policy Database (SPD) for SA2-O

tunnel source address	NUT_Link1
tunnel destination address	SGW2_Link2
source address	Link0
destination address	Link4
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within SA1's ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request from HOST2

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply to HOST2

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply within SA1's ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Request within SA2’s ESP

IP Header	Source Address	SGW2_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x3000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin02
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in02
IP Header	Source Address	HOST4_Link4
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request from HOST4

IP Header	Source Address	HOST4_Link4
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply to HOST4

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST4_Link4
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply within SA2’s ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW2_Link2
ESP	SPI	0x4000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout2
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out2
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST4_Link4
ICMP	Type	129 (Echo Reply)

Procedure:

```
SGW1_Link2(TN)    Target(NUT)    HOST1_Link0(TN)
    |                 |              |
    |---------------->|              | ICMP Echo Request within SA1's ESP
    |                 |              |
    |                 |------------->| ICMP Echo Request from HOST2
    |                 |              | (SRC=HOST2_Link3/DST=HOST1_Link0)
    |                 |              |      (Judgment  #1)
    |                 |              |
    |                 |<-------------| ICMP Echo Reply to HOST2
    |                 |              | (SRC=HOST1_Link0/DST=HOST2_Link3)
    |                 |              |
    |<----------------|              | ICMP Echo Reply within SA1's ESP
    |                 |              |      (Judgment  #2)

SGW2_Link2(TN)    Target(NUT)    HOST1_Link0(TN)
    |                 |              |
    |---------------->|              | ICMP Echo Request within SA2's ESP
    |                 |              |
    |                 |------------->| ICMP Echo Request from HOST4
    |                 |              | (SRC=HOST4_Link4/DST=HOST1_Link0)
    |                 |              |      (Judgment  #3)
    |                 |              |
    |                 |<-------------| ICMP Echo Reply to HOST4
    |                 |              | (SRC=HOST1_Link0/DST=HOST4_Link4)
    |                 |              |
    |<----------------|              | ICMP Echo Reply within SA2's ESP
    |                 |              |      (Judgment  #4)
```

PartA: SA1-I

1. SGW1 sends *"ICMP Echo Request within SA1's ESP"* (originally from HOST2)
2. Observe the packet transmitted by NUT

PartB: SA1-O

3. HOST1 sends *"ICMP Echo Reply to HOST2"*
4. Observe the packet transmitted by NUT

PartC: SA2-I

5. SGW1 sends *"ICMP Echo Request within SA2's ESP"* (originally from HOST4)
6. Observe the packet transmitted by NUT

PartD: SA2-O

7. HOST1 sends *"ICMP Echo Reply to HOST4"*

8. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
Step-2: NUT transmits *"ICMP Echo Request from HOST2"*

PartB: Judgment #2
Step-4: NUT transmits *"ICMP Echo Reply within SA1's ESP"*

PartC: Judgment #3
Step-6: NUT transmits *"ICMP Echo Request from HOST4"*

PartD: Judgment #4
Step-8: NUT transmits *"ICMP Echo Reply within SA2's ESP"*

References:

RFC2401: Security Architecture for IP
Section 4.4.1

6.1.2. Select SPD for 2 Hosts behind 1 SGW

Purpose:

Verify that a NUT (SGW) selects appropriate SPD
(SGW tunnel mode, ESP=3DES-CBC)

Category:

End-Node : N/A
SGW : BASIC (A requirement for all SGW NUTs)

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
  HOST2_Link3 -- SGW1 -------------------- NUT -- HOST1_Link0
                      -------------------> SA1-I
                      <------------------- SA1-O

  HOST3_Link3 -- SGW1 -------------------- NUT -- HOST1_Link0
                      -------------------> SA2-I
                      <------------------- SA2-O
```

Security Association Database (SAD) for SA1-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA1-I

tunnel source address	SGW1_Link2
tunnel destination address	NUT_Link1
source address	HOST2_Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA1-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA1-O

tunnel source address	NUT_Link1
tunnel destination address	SGW1_Link2
source address	Link0
destination address	HOST2_Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA2-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x3000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin02
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in02

Security Policy Database (SPD) for SA2-I

tunnel source address	SGW1_Link2
tunnel destination address	NUT_Link1
source address	HOST3_Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA2-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x4000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout2
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out2

Security Policy Database (SPD) for SA2-O

tunnel source address	NUT_Link1
tunnel destination address	SGW1_Link2
source address	Link0
destination address	HOST3_Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within SA1's ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request from HOST2

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply to HOST2

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply within SA1's ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Request within SA2's ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x3000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin02
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in02
IP Header	Source Address	HOST3_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request from HOST3

IP Header	Source Address	HOST3_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply to HOST3

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST3_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply within SA2's ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x4000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout2
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out2
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST3_Link3
ICMP	Type	129 (Echo Reply)

Procedure:

```
SGW1_Link2(TN)    Target(NUT)    HOST1_Link0(TN)
    |                |              |
    |--------------->|              | ICMP Echo Request within SA1's ESP
    |                |              |
    |                |------------->| ICMP Echo Request from HOST2
    |                |              | (SRC=HOST2_Link3/DST=HOST1_Link0)
    |                |              |      (Judgment  #1)
    |                |              |
    |                |<-------------| ICMP Echo Reply to HOST2
    |                |              | (SRC=HOST1_Link0/DST=HOST2_Link3)
    |                |              |
    |<---------------|              | ICMP Echo Reply within SA1's ESP
    |                |              |      (Judgment  #2)
    |                |              |
    |--------------->|              | ICMP Echo Request within SA2's ESP
    |                |              |
    |                |------------->| ICMP Echo Request from HOST3
    |                |              | (SRC=HOST3_Link3/DST=HOST1_Link0)
    |                |              |      (Judgment  #3)
    |                |              |
    |                |<-------------| ICMP Echo Reply to HOST3
    |                |              | (SRC=HOST1_Link0/DST=HOST3_Link3)
    |                |              |
    |<---------------|              | ICMP Echo Reply within SA2's ESP
    |                |              |      (Judgment  #4)
```

PartA: SA1-I

1. SGW1 sends *"ICMP Echo Request within SA1's ESP"* (originally from HOST2)
2. Observe the packet transmitted by NUT

PartB: SA1-O

3. HOST1 sends *"ICMP Echo Reply to HOST2"*
4. Observe the packet transmitted by NUT

PartC: SA2-I

5. SGW1 sends *"ICMP Echo Request within SA2's ESP"* (originally from HOST3)
6. Observe the packet transmitted by NUT

PartD: SA2-O

7. HOST1 sends *"ICMP Echo Reply to HOST3"*
8. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
 Step-2: NUT transmits *"ICMP Echo Request from HOST2"*

PartB: Judgment #2
 Step-4: NUT transmits *"ICMP Echo Reply within SA1's ESP"*

PartC: Judgment #3
 Step-6: NUT transmits *"ICMP Echo Request from HOST3"*

PartD: Judgment #4
 Step-8: NUT transmits *"ICMP Echo Reply within SA2's ESP"*

References:

RFC2401: Security Architecture for IP
 Section 4.4.1

6.1.3. Sequence Number Increment

Purpose:

Verify that a NUT (SGW) increases sequence number correctly, starting with 1.
(SGW tunnel mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : N/A
SGW : BASIC (A requirement for all SGW NUTs)

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
  HOST2_Link3 -- SGW1 ------------------- NUT -- HOST1_Link0
                     -------------------> SA-I
                     <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

Tunnel source address	NUT_Link0
Tunnel destination address	HOST1_Link1
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets：

ICMP Echo Request

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	128 (Echo Request)

ICMP Echo Request within ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Sequence	1st = 1, 2nd = 2
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	128 (Echo Request)
	Data Length	7

Procedure：

```
SGW1_Link2(TN)    Target(NUT)    HOST1_Link0(TN)
     |                |              |
     |                |<-------------| ICMP Echo Request
     |                |              | (SRC=HOST1_Link0/DST=HOST2_Link3)
     |                |              |
     |<---------------|              | ICMP Echo Request within ESP
     |                |              |      (Judgment  #1)
     |                |              |
     |                |<-------------| ICMP Echo Request
     |                |              | (SRC=HOST1_Link0/DST=HOST2_Link3)
     |                |              |
     |<---------------|              | ICMP Echo Request within ESP
     |                |              |      (Judgment  #2)
```

PartA： SA-I

1. HOST1 sends *"ICMP Echo Request"*
2. Observe the packet transmitted by NUT

PartB： SA-O

3. HOST1 sends *"ICMP Echo Request"*
4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
Step-2: NUT transmits an *"ICMP Echo Request within ESP"* with an ESP Sequence number of 1

PartB: Judgment #2
Step-4: NUT transmits an *"ICMP Echo Request within ESP"* with an ESP Sequence number of 2

References:

RFC2401 : Security Architecture for the Internet Protocol
RFC2406 : IP Encapsulating Security Payload (ESP), Section 3.3.3

6.1.4. Packet Too Big Transmission

Purpose:

Verify that a NUT (SGW) transmits the ICMP Error Message (Packet Too Big) correctly. (SGW tunnel mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : N/A
SGW : BASIC (A requirement for all SGW NUTs)

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
  HOST2_Link3 -- SGW1 --------------------- NUT -- HOST1_Link0
                      --------------------> SA-I
                      <-------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

Tunnel source address	NUT_Link0
Tunnel destination address	HOST1_Link1
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
	Payload Length	1460
ICMP	Type	128 (Echo Request)

ICMP Error Message (Packet Too Big)

IP Header	Source Address	NUT_Link0
	Destination Address	HOST1_Link0
ICMP	Type	2 (Packet Too Big)
	MTU	1280 <= n <= 1430 (e.g., 1280)
	Data	1232Byte of *ICMP Echo Request*

Fragmented ICMP Echo Request to Host2 1

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
	Payload Length	*1stPL*(=MTU-40) (e.g., 1240)
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request to Host2 2

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_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 to Host2 within ESP 1

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
	Payload Length	*1stPL*
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request to Host2 within ESP 2

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
	Payload Length	*2ndPL*
Fragment	Offset	(1stPL-8)/8
	More Flag	0
Data	Data	Rest of *ICMP Echo Request*

Procedure:

```
SGW1_Link2(TN)    Target(NUT)    HOST1_Link0(TN)
    |                |             |
    |                |<------------| ICMP Echo Request from HOST1
    |                |             |
    |                |------------>| ICMP Error Message (Packet Too Big)
    |                |             |     (Judgment  #1)
    |                |             |
    |                |<------------| Fragmented ICMP Echo Request to HOST2 1
    |                |<------------| Fragmented ICMP Echo Request to HOST2 2
    |                |             |
    |<---------------|             | Fragmented ICMP Echo Request to
    |                |             | HOST2 within ESP 1
    |<---------------|             | Fragmented ICMP Echo Request to
    |                |             | HOST2 within ESP 2
    |                |             |      (Judgment  #2)
```

1. HOST1 sends *"ICMP Echo Request"*
2. Observe the packet transmitted by NUT
3. HOST1 sends *"Fragmented ICMP Echo Request to HOST2 1"* and *"Fragmented ICMP Echo Request to HOST2 2"*
4. Observe the packet transmitted by NUT

Judgment:

Judgment #1
Step-2: NUT transmits *"ICMP Error Message (Packet Too Big)"*

Judgment #2
Step-4: NUT transmits *"Fragmented ICMP Echo Request within ESP 1"*
and *"Fragmented ICMP Echo Request within ESP 2"*

References:

RFC1851 : The ESP Triple DES Transform
RFC2404 : The Use of HMAC-SHA-1-96 within ESP and AH

6.1.5. Packet Too Big Forwarding (Unknown Original Host)

Purpose:

Verify that a NUT (SGW) forwards the ICMP Error Message (Packet Too Big) correctly when NUT can not determine the original host. (SGW tunnel mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : N/A
SGW : BASIC (A requirement for all SGW NUTs)

Initialization:

Use common topology described as Fig.4. Router1’s interface to Link2 has an MTU value of 1356.
Set NUT's SAD and SPD as following:

```
  HOST2_Link3 -- SGW1 -------------------- NUT -- HOST1_Link0
                      -------------------> SA-I
                      <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

Tunnel source address	NUT_Link1
Tunnel destination address	SGW1_Link2
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
	Payload Length	1360
ICMP	Type	128 (Echo Request)

ICMP Error Message to NUT (Packet Too Big)

IP Header	Source Address	ROUTER1_Link2
	Destination Address	NUT_Link1
ICMP	Type	2 (Packet Too Big)
	MTU	1356
	Data	1232Byte of *ICMP Echo Request*

ICMP Error Message to HOST1 (Packet Too Big)

IP Header	Source Address	ROUTER1_Link2 or NUT_Link1
	Destination Address	HOST1_Link0
ICMP	Type	2 (Packet Too Big)
	MTU	1280 - 1286
	Data	1232Byte of *ICMP Echo Request*

Fragmented ICMP Echo Request 1

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
	Payload Length	1240
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request 2

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
	Payload Length	136
Fragment	Offset	154
	More Flag	0
Data	Data	Rest of *ICMP Echo Request*

ICMP Echo Request within ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
	Payload Length	1360
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request within ESP 1

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
	Payload Length	1240
Fragment	Offset	0
	More Flag	1
ICMP	Type	128 (Echo Request)

Fragmented ICMP Echo Request within ESP 2

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
	Payload Length	136
Fragment	Offset	154
	More Flag	0
Data	Data	Rest of *ICMP Echo Request*

Procedure:

```
SGW1_Link2(TN)  ROUTER1_Link2(TN) Target(NUT)   HOST1_Link0(TN)
    |                |               |              |
    |                |               |<-------------| ICMP Echo Request
    |                |               |              |
    |<-------------------------------|              | ICMP Echo Request
    |                |               |              | to HOST2 within ESP
    |                |               |              |     (Judgment  #1)
    |                |               |              |
    |                |-------------->|              | ICMP Error Message to NUT
    |                |               |              | (Packet Too Big)
    |                |               |              |
    |                |               |<-------------| ICMP Echo Request
    |                |               |              |
    |                |               |------------->| ICMP Error Message
    |                |               |              | to HOST1(Packet Too Big)
    |                |               |              |     (Judgment  #2)
    |                |               |              |
    |                |               |<-------------| Fragmented ICMP Echo
    |                |               |              | Request 1
    |                |               |<-------------| Fragmented ICMP Echo
    |                |               |              | Request 2
    |                |               |              |
    |<-------------------------------|              | Fragmented ICMP Echo
    |                |               |              | Request within ESP 1
    |<-------------------------------|              | Fragmented ICMP Echo
    |                |               |              | Request within ESP 2
    |                |               |              |    (Judgment  #3)
```

1. HOST1 sends *"ICMP Echo Request"*
2. Observe the packet transmitted by NUT
3. ROUTER1 sends *"ICMP Error Message to NUT (Packet Too Big)"*
4. HOST1 sends *"ICMP Echo Request"*
5. Observe the packet transmitted by NUT
6. HOST1 sends *"Fragmented ICMP Echo Request 1"*
   and *"Fragmented ICMP Echo Request 2"*
7. Observe the packet transmitted by NUT

Judgment:

Judgment #1
Step-2: NUT transmits *"ICMP Echo Request within ESP"*

Judgment #2
Step-5: NUT transmits *"ICMP Error Message to HOST1 (Packet Too Big)"*

Judgment #3
Step-7: NUT transmits *"Fragmented ICMP Echo Request within ESP 1"*
and *"Fragmented ICMP Echo Request within ESP 2"*

References:

RFC2401 : Security Architecture for the Internet Protocol
Section 6.1.2

6.1.6. Bypass Policy

Purpose:

Verify that a NUT (SGW) select bypass or discard policies

Category:

End-Node : N/A
SGW : ADVANCED (This test is required for all SGW NUTs which support Bypass Policy, regardless of explicitly or implicitly)

NOTE: NUT need to pass at least either of "Bypass Policy" or "Discard Policy" tests.

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
  HOST2_Link3 -- SGW1 -------------------- NUT -- HOST1_Link0
                      -------------------> SA-I
                      <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

Tunnel source address	NUT_Link1
Tunnel destination address	SGW1_Link2
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request from HOST2

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request from HOST4

IP Header	Source Address	HOST4_Link4
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

Procedure:

```
SGW1_Link2(TN)   Target(NUT)   HOST1_Link0(TN)
    |                |              |
    |--------------->|              | ICMP Echo Request within ESP
    |                |              | (SRC=SGW1_Link2/DST=NUT_Link1)
    |                |              |
    |                |------------->| ICMP Echo Request
    |                |              |      (Judgment #1)
   ================================ Set Bypass Policy to NUT

HOST4_Link4(TN)   Target(NUT)   HOST1_Link0(TN)
    |                |              |
    |--------------->|              | ICMP Echo Request from HOST4
    |                |              | (SRC=HOST4_Link4/DST=HOST1_Link0)
    |                |              |
    |                |------------->| ICMP Echo Request from HOST4
    |                |              |      (Judgment #2)
```

PartA: Confirmation

1. SGW1 sends *"ICMP Echo Request within ESP"*
2. Observe the packet transmitted by NUT

PartB: Bypass Policy

3. Set Bypass Policy for above ICMP Echo Reply to NUT as following example
4. SGW1 sends "ICMP Echo Request from HOST4"
5. Observe the packet transmitted by NUT

Example 1: Security Policy Database (SPD) for policy=bypass (none)

source address	HOST4_Link4
destination address	HOST1_Link0
upper spec	any
direction	out
policy	bypass(none)

Example 2: Security Policy Database (SPD) for policy=bypass (none) as default policy

source address	any
destination address	any
upper spec	any
direction	out
policy	bypass(none)

Judgment:

PartA: Judgment #1
Step-2: NUT transmits *"ICMP Echo Request"*

PartB: Judgment #2
Step-5: NUT transmits *"ICMP Echo Request from HOST4"*

References:

RFC2401 : Security Architecture for the Internet Protocol
Section 4.4.1

6.1.7. Discard Policy

Purpose:

Verify that a NUT (SGW) select bypass or discard policies

Category:

End-Node : N/A
SGW : ADVANCED (This test is required for all SGW NUTs which support Discard Policy, regardless of explicitly or implicitly)

NOTE: NUT need to pass at least either of "Bypass Policy" or "Discard Policy" tests.

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
  HOST2_Link3 -- SGW1 --------------------- NUT -- HOST1_Link0
                      -------------------> SA-I
                      <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database （SPD） for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database （SAD） for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database （SPD） for SA-O

Tunnel source address	NUT_Link1
Tunnel destination address	SGW1_Link2
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request from HOST4

IP Header	Source Address	HOST4_Link4
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

Procedure:

```
SGW1_Link2(TN)    Target(NUT)   HOST1_Link0(TN)
    |                |             |
    |--------------->|             | ICMP Echo Request
    |                |             | (SRC=SGW1_Link2/DST=NUT_Link1)
    |                |             |
    |                |------------>| ICMP Echo Request
    |                |             |      (Judgment #1)
   ================================ Set Discard Policy to NUT

HOST4_Link4(TN)   Target(NUT)   HOST1_Link0(TN)
    |                |             |
    |--------------->|             | ICMP Echo Request from HOST4
    |                |             | (SRC=HOST4_Link4/DST=HOST1_Link0)
    |                |             |
    |                |(----------->)| No ICMP Echo Request from HOST4
    |                |             |      (Judgment #2)
```

PartA: Confirmation

1. SGW1 sends *"ICMP Echo Request"*
2. Observe the packet transmitted by NUT

PartB: discard policy

3. Set discard policy for above ICMP Echo Reply to NUT as following example
4. HOST4 sends *"ICMP Echo Request from HOST4"*
5. Observe the packet transmitted by NUT

Example 1: Security Policy Database (SPD) for policy=discard

source address	HOST4_Link4
destination address	HOST1_Link0
upper spec	any
direction	out
policy	discard

Example 2: Security Policy Database (SPD) for policy=discard as default policy

source address	any
destination address	any
upper spec	any
direction	out
policy	discard

Judgment:

PartA: Judgment #1

Step-2: NUT transmits *"ICMP Echo Request"*

PartB: Judgment #2

Step-5: NUT does not transmits *"ICMP Echo Request from HOST4"*

References:

RFC2401 : Security Architecture for the Internet Protocol
Section 4.4.1

6.1.8. Tunnel Mode Padding

Purpose:

Verify that a NUT (SGW) supports padding & padding byte handling
(SGW tunnel mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : N/A
SGW   : BASIC (A requirement for all SGW NUTs)

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
  HOST2_Link3 -- SGW1 ------------------- NUT -- HOST1_Link0
                     -------------------> SA-I
                     <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

Tunnel source address	NUT_Link0
Tunnel destination address	HOST1_Link1
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
	Padding	Sequential
	Padding Length	7+8n (0 <= n <= 31)
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)
	Data Length	7

ICMP Echo Request

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply within ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-MD5
	Authentication Key	ipv6readylogsha1out1
	Padding	Sequential
	Padding Length	7+8n (0 <= n <= 31)
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)
	Data Length	7

Procedure:

```
SGW1_Link2(TN)   Target(NUT)   HOST1_Link0(TN)
    |                |              |
    |--------------->|              | ICMP Echo Request within ESP
    |                |              | (padding length is 7)
    |                |              |
    |                |------------->| ICMP Echo Request
    |                |              | (SRC=HOST2_Link3/DST=HOST1_Link0)
    |                |              |      (Judgment  #1)
    |                |              |
    |                |<-------------| ICMP Echo Reply
    |                |              | (SRC=HOST1_Link0/DST=HOST2_Link3)
    |                |              |
    |<---------------|              | ICMP Echo Reply within ESP
    |                |              |      (Judgment  #2)
    |                |              |
    |--------------->|              | ICMP Echo Request within ESP
    |                |              | (padding length is 255)
    |                |              |
    |                |------------->| ICMP Echo Request
    |                |              | (SRC=HOST2_Link3/DST=HOST1_Link0)
    |                |              |      (Judgment  #3)
    |                |              |
    |                |<-------------| ICMP Echo Reply
    |                |              | (SRC=HOST1_Link0/DST=HOST2_Link3)
    |                |              |
    |<---------------|              | ICMP Echo Reply within ESP
    |                |              |      (Judgment  #4)
```

PartA: Padding Length is 7

1. SGW1 sends *"ICMP Echo Request within ESP"(Padding Length=7)*
2. Observe the packet transmitted by NUT
3. HOST1 sends *"ICMP Echo Reply"*
4. Observe the packet transmitted by NUT

PartB: Padding Length is 255

5. SGW1 sends *"ICMP Echo Request within ESP" (Padding Length=255)*
6. Observe the packet transmitted by NUT
7. HOST1 sends *"ICMP Echo Reply"*
8. Observe the packet transmitted by NUT

Judgment:

PartA: Padding Length is 7
Judgment #1
 Step-2: NUT transmits *"ICMP Echo Request"*
Judgment #2
 Step-4: NUT transmits *"ICMP Echo Reply within ESP"*

PartB: Padding Length is 255
Judgment #3
 Step-6: NUT transmits *"ICMP Echo Request"*
Judgment #4
 Step-8: NUT transmits *"ICMP Echo Reply within ESP"*

References:

RFC2406 : IP Encapsulating Security Payload (ESP)
Section 2.4

6.1.9. Non-Registered SPI

Purpose:

Verify that a NUT (SGW) can behave when No valid Security Association is configured.

Category:

End-Node : N/A
SGW : BASIC (A requirement for all SGW NUTs)

Initialization:

Use common topology described as Fig.4
Set NUT's SAD and SPD as following:

```
  HOST2_Link3 -- SGW1 ------------------- NUT -- HOST1_Link0
                     -------------------> SA-I
                     <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

Tunnel source address	NUT_Link1
Tunnel destination address	SGW1_Link2
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP 1

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Sequence Number	1
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request within ESP 2 with non-registered SPI

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x9000 (different from SA-I's SPD)
	Sequence Number	1
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

Procedure:

```
SGW1_Link2(TN)   Target(NUT)   HOST1_Link0(TN)
    |                |              |
    |--------------->|              | ICMP Echo Request within ESP 1
    |                |              |
    |                |------------->| ICMP Echo Request
    |                |              | (SRC=HOST2_Link3/DST=HOST1_Link0)
    |                |              |      (Judgment  #1)
    |                |              |
    |                |              |
    |--------------->|              | ICMP Echo Request within ESP 2
    |                |              | w/ unknown SPI value
    |                |              |
    |                |(----------->)| No Request
    |                |              |      (Judgment  #2)
```

PartA: valid SA exists
 1. SGW1 sends *"ICMP Echo Request within ESP 1"*
 2. Observe the packet transmitted by NUT

PartB: no valid SA exists
 3. SGW1 sends *"ICMP Echo Request within ESP 2"*
 4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
 Step-2: NUT transmits "ICMP Echo Request"

PartB: Judgment #2
 Step-4: NUT does not transmit *"ICMP Echo Request"*

References:

RFC2406 : IP Encapsulating Security Payload (ESP)
     Section 3.4.2

6.1.10. ICV

Purpose:

Verify that a NUT (SGW) can detect the modification by examining the ICV
(SGW tunnel mode, ESP=3DES-CBC HMAC-SHA1)

Category:

End-Node : N/A
SGW : BASIC (A requirement for all SGW NUTs)

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
  HOST2_Link3 -- SGW1 ------------------- NUT -- HOST1_Link0
                     -------------------> SA-I
                     <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-0

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-0

Tunnel source address	NUT_Link1
Tunnel destination address	SGW1_Link2
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP 1

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Sequence number	1
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)
	Data	"PadLen is zero"

ICMP Echo Request

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)
	Data	"PadLen is zero"

ICMP Echo Request within ESP 2

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Sequence number	2
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
	ICV	aaaaaaaaa........
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)
	Data	"crackedcracked" ("PadLen is zero" is original)

Procedure:

```
SGW1_Link2(TN)   Target(NUT)   HOST1_Link0(TN)
    |                |             |
    |--------------->|             | ICMP Echo Request within ESP 1
    |                |             |
    |                |------------>| ICMP Echo Request
    |                |             | (SRC=HOST2_Link3/DST=HOST1_Link0)
    |                |             |      (Judgment  #1)
    |                |             |
    |                |             |
    |--------------->|             | ICMP Echo Request within ESP 2
    |                |             | with INCORRECT ICV
    |                |             |
    |                |(----------->)| No ICMP Echo Request
    |                |             |      (Judgment  #2)
```

PartA: correct packet
1. SGW1 sends *"ICMP Echo Request within ESP 1"*
2. Observe the packet transmitted by NUT

PartB: modified packet
3. SGW1 sends *"ICMP Echo Request with ESP 2"* (with INCORRECT ICV)
4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
Step-2: NUT transmits *"ICMP Echo Request"*

PartB: Judgment #2
Step-4: NUT does not transmit *"ICMP Echo Request"*

References:

RFC2406 : IP Encapsulating Security Payload (ESP)
Section 3.4.3

6.1.11. Tunnel Mode with End-Node

Purpose:

Verify that a NUT (SGW) can build IPsec tunnel mode with End-Node correctly, ESP=3DES-CBC

Category:

End-Node : N/A
SGW : BASIC (A requirement for all SGW NUTs)

Initialization:

Use common topology described as Fig.3
Set NUT's SAD and SPD as following:

```
    HOST2 -------------------- NUT -- HOST1
          -------------------> SA-I
          <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	HOST2_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

tunnel source address	HOST2_Link2
tunnel destination address	NUT_Link1
source address	HOST2_Link2
destination address	HOST1_Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	HOST2_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP algorithm key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

tunnel source address	NUT_Link1
tunnel destination address	HOST2_Link2
source address	HOST1_Link0
destination address	HOST2_Link2
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP tunnel

IP Header	Source Address	HOST2_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST2_Link2
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	HOST2_Link2
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link2
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply within ESP tunnel

IP Header	Source Address	NUT_Link1
	Destination Address	HOST2_Link2
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link2
ICMP	Type	129 (Echo Reply)

Procedure:

```
HOST2_Link2(TN)   Target(NUT)   HOST1_Link0(TN)
    |               |              |
    |-------------->|              | ICMP Echo Request within ESP
    |               |              |
    |               |------------->| ICMP Echo Request
    |               |              | (SRC=HOST1_Link0/DST=HOST2_Link2)
    |               |              |      (Judgment  #1)
    |               |              |
    |               |<-------------| ICMP Echo Reply
    |               |              | (SRC=HOST2_Link2/DST=HOST1_Link0)
    |               |              |
    |<--------------|              | ICMP Echo Reply within ESP
    |               |              |      (Judgment  #2)
```

PartA: SA-I
 1. HOST2 sends *"ICMP Echo Request within ESP"*
 2. Observe the packet transmitted by NUT

PartB: SA-O
 3. HOST1 sends *"ICMP Echo Reply"*
 4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
 Step-2: NUT transmits *"ICMP Echo Request"*

PartB: Judgment #2
 Step-4: NUT transmits *"ICMP Echo Reply within ESP"*

References:

RFC2401: Security Architecture for IP
     Section 3.4.3

6.2. Algorithm Test

Scope:

Following tests focus on Encryption and Authentication Algorithms.

Overview:

Tests in this section verify that the NUT properly decrypt the received packets and encrypts the transmitting packets using Encryption algorithms specified in the SAD.
And they verify that the NUT properly processes the authentication algorithms specified in the SAD.

6.2.1. Tunnel Mode ESP=3DES-CBC HMAC-SHA1

Purpose:

SGW tunnel mode, ESP=3DES-CBC HMAC-SHA1

Category:

End-Node : N/A
SGW   : BASIC (A requirement for all SGW NUTs)

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
 HOST2_Link3 -- SGW1 -------------------- NUT -- HOST1_Link0
                     -------------------> SA-I
                     <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

Tunnel source address	NUT_Link1
Tunnel destination address	SGW1_Link2
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply within ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

Procedure:

```
SGW1_Link2(TN)    Target(NUT)    HOST1_Link0(TN)
     |                 |                |
     |---------------->|                | ICMP Echo Request within ESP
     |                 |                |
     |                 |--------------->| ICMP Echo Request
     |                 |                | (SRC=HOST2_Link3/DST=HOST1_Link0)
     |                 |                |      (Judgment  #1)
     |                 |                |
     |                 |<---------------| ICMP Echo Reply
     |                 |                | (SRC=HOST1_Link0/DST=HOST2_Link3)
     |                 |                |
     |<----------------|                | ICMP Echo Reply within ESP
     |                 |                |      (Judgment  #2)
```

PartA: SA-I

1. SGW1 sends *"ICMP Echo Request within ESP"*
2. Observe the packet transmitted by NUT

PartB: SA-0

3. HOST1 sends *"ICMP Echo Reply"*
4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1

Step-2: NUT transmits *"ICMP Echo Request"*

PartB: Judgment #2

Step-4: NUT transmits *"ICMP Echo Reply within ESP"*

References:

RFC1851 : The ESP Triple DES Transform
RFC2401 : Security Architecture for the Internet Protocol
RFC2404 : The Use of HMAC-SHA-1-96 within ESP and AH
RFC2405 : The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC2406 : IP Encapsulating Security Payload (ESP)

6.2.2. Tunnel Mode ESP=3DES-CBC AES-XCBC

Purpose:

SGW tunnel mode, ESP=3DES-CBC AES-XCBC

Category:

End-Node : N/A
SGW : ADVANCED (This test is required for all SGW NUTs which support AES-XCBC as an authentication algorithm)

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
 HOST2_Link3 -- SGW1 -------------------- NUT -- HOST1_Link0
                     -------------------> SA-I
                     <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	AES-XCBC
ESP authentication key	ipv6readaesxin01

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	AES-XCBC
ESP authentication key	ipv6readaesxout1

Security Policy Database (SPD) for SA-O

Tunnel source address	NUT_Link1
Tunnel destination address	SGW1_Link2
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	AES-XCBC
	Authentication Key	ipv6readaesxin01
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply within ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	AES-XCBC
	Authentication Key	ipv6readaesxout1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

Procedure:

```
SGW1_Link2(TN)   Target(NUT)   HOST1_Link0(TN)
    |               |              |
    |-------------->|              | ICMP Echo Request within ESP
    |               |              |
    |               |------------->| ICMP Echo Request
    |               |              | (SRC=HOST2_Link3/DST=HOST1_Link0)
    |               |              |      (Judgment  #1)
    |               |              |
    |               |<-------------| ICMP Echo Reply
    |               |              | (SRC=HOST1_Link0/DST=HOST2_Link3)
    |               |              |
    |<--------------|              | ICMP Echo Reply within ESP
    |               |              |      (Judgment  #2)
```

PartA: SA-I

1. SGW1 sends *"ICMP Echo Request within ESP"*
2. Observe the packet transmitted by NUT

PartB: SA-O

3. HOST1 sends *"ICMP Echo Reply"*
4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1

Step-2: NUT transmits *"ICMP Echo Request"*

PartB: Judgment #2

Step-4: NUT transmits *"ICMP Echo Reply within ESP"*

References:

RFC1851 : The ESP Triple DES Transform
RFC2401 : Security Architecture for the Internet Protocol
RFC2405 : The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC2406 : IP Encapsulating Security Payload (ESP)
RFC3566 : The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec

6.2.3. Tunnel Mode ESP=3DES-CBC NULL

Purpose:

SGW tunnel mode, ESP=3DES-CBC NULL

Category:

End-Node : N/A
SGW : ADVANCED (This test is required for all SGW NUTs which support NULL as an authentication algorithm)

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
 HOST2_Link3 -- SGW1 -------------------- NUT -- HOST1_Link0
                     -------------------> SA-I
                     <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	NULL
ESP authentication key	

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	NULL
ESP authentication key	

Security Policy Database (SPD) for SA-O

Tunnel source address	NUT_Link1
Tunnel destination address	SGW1_Link2
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	NULL
	Authentication Key	
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply within ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	NULL
	Authentication Key	
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

Procedure:

```
SGW1_Link2(TN)    Target(NUT)    HOST1_Link0(TN)
    |                |              |
    |--------------->|              | ICMP Echo Request within ESP
    |                |              |
    |                |------------->| ICMP Echo Request
    |                |              | (SRC=HOST2_Link3/DST=HOST1_Link0)
    |                |              |      (Judgment  #1)
    |                |              |
    |                |<-------------| ICMP Echo Reply
    |                |              | (SRC=HOST1_Link0/DST=HOST2_Link3)
    |                |              |
    |<---------------|              | ICMP Echo Reply within ESP
    |                |              |      (Judgment  #2)
```

PartA: SA-I

1. SGW1 sends *"ICMP Echo Request within ESP"*
2. Observe the packet transmitted by NUT

PartB: SA-O

3. HOST1 sends *"ICMP Echo Reply"*
4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
 Step-2: NUT transmits *"ICMP Echo Request"*

PartB: Judgment #2
 Step-4: NUT transmits *"ICMP Echo Reply within ESP"*

References:

RFC1851 : The ESP Triple DES Transform
RFC2401 : Security Architecture for the Internet Protocol
RFC2405 : The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC2406 : IP Encapsulating Security Payload (ESP)
RFC2410 : The NULL Encryption Algorithm and Its Use With IPsec

6.2.4. Tunnel Mode ESP=3DES-CBC HMAC-MD5

Purpose:

SGW tunnel mode, ESP=3DES-CBC HMAC-MD5

Category:

End-Node : N/A
SGW : ADVANCED (This test is required for all SGW NUTs which support HMAC-MD5 as an authentication algorithm)

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
 HOST2_Link3 -- SGW1 -------------------- NUT -- HOST1_Link0
                     -------------------> SA-I
                     <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcin01
ESP authentication	HMAC-MD5
ESP authentication key	ipv6readymd5in01

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	3DES-CBC
ESP key	ipv6readylogo3descbcout1
ESP authentication	HMAC-MD5
ESP authentication key	ipv6readymd5out1

Security Policy Database (SPD) for SA-O

Tunnel source address	NUT_Link1
Tunnel destination address	SGW1_Link2
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcin01
	Authentication Algorithm	HMAC-MD5
	Authentication Key	ipv6readymd5in01
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply within ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	3DES-CBC
	Key	ipv6readylogo3descbcout1
	Authentication Algorithm	HMAC-MD5
	Authentication Key	ipv6readymd5out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

Procedure:

```
SGW1_Link2(TN)    Target(NUT)    HOST1_Link0(TN)
    |                |              |
    |--------------->|              | ICMP Echo Request within ESP
    |                |              |
    |                |------------->| ICMP Echo Request
    |                |              | (SRC=HOST2_Link3/DST=HOST1_Link0)
    |                |              |      (Judgment  #1)
    |                |              |
    |                |<-------------| ICMP Echo Reply
    |                |              | (SRC=HOST1_Link0/DST=HOST2_Link3)
    |                |              |
    |<---------------|              | ICMP Echo Reply within ESP
    |                |              |      (Judgment  #2)
```

PartA: SA-I
 1. SGW1 sends *"ICMP Echo Request within ESP"*
 2. Observe the packet transmitted by NUT

PartB: SA-O
 3. HOST1 sends *"ICMP Echo Reply"*
 4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
 Step-2: NUT transmits *"ICMP Echo Request"*

PartB: Judgment #2
 Step-4: NUT transmits *"ICMP Echo Reply within ESP"*

References:

RFC1851 : The ESP Triple DES Transform
RFC2401 : Security Architecture for the Internet Protocol
RFC2403 : The Use of HMAC-MD5-96 within ESP and AH
RFC2405 : The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC2406 : IP Encapsulating Security Payload (ESP)

6.2.5.　　Tunnel Mode ESP=AES-CBC (128-bit) HMAC-SHA1

Purpose:

SGW tunnel mode, ESP=DES-CBC HMAC-SHA1

Category:

End-Node : N/A
SGW　　　: ADVANCED (This test is required for all SGW NUTs which support AES-CBC (128-bit) as an encryption algorithm)

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
 HOST2_Link3 -- SGW1 -------------------- NUT -- HOST1_Link0
                     -------------------> SA-I
                     <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	AES-CBC(128-bit)
ESP key	ipv6readaescin01
ESP authentication algorithm	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	AES-CBC(128-bit)
ESP key	ipv6readaescout1
ESP authentication algorithm	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

Tunnel source address	NUT_Link1
Tunnel destination address	SGW1_Link2
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Algorithm	AES-CBC(128-bit)
	Key	ipv6readaescin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply within ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	AES-CBC(128-bit)
	Key	ipv6readaescout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

Procedure:

```
SGW1_Link2(TN)    Target(NUT)    HOST1_Link0(TN)
    |                |              |
    |--------------->|              | ICMP Echo Request within ESP
    |                |              |
    |                |------------->| ICMP Echo Request
    |                |              | (SRC=HOST2_Link3/DST=HOST1_Link0)
    |                |              |      (Judgment  #1)
    |                |              |
    |                |<-------------| ICMP Echo Reply
    |                |              | (SRC=HOST1_Link0/DST=HOST2_Link3)
    |                |              |
    |<---------------|              | ICMP Echo Reply within ESP
    |                |              |      (Judgment  #2)
```

PartA: SA-I
1. SGW1 sends *"ICMP Echo Request within ESP"*
2. Observe the packet transmitted by NUT

PartB: SA-O
3. HOST1 sends *"ICMP Echo Reply"*
4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
Step-2: NUT transmits *"ICMP Echo Request"*

PartB: Judgment #2
Step-4: NUT transmits *"ICMP Echo Reply within ESP"*

References:

RFC2401 : Security Architecture for the Internet Protocol
RFC2404 : The Use of HMAC-SHA-1-96 within ESP and AH
RFC2406 : IP Encapsulating Security Payload (ESP)
RFC3602 : The AES-CBC Cipher Algorithm and Its Use with IPsec

6.2.6. Tunnel Mode ESP=NULL HMAC-SHA1

Purpose:

SGW tunnel mode, ESP=NULL HMAC-SHA1

Category:

End-Node : N/A
SGW : ADVANCED (This test is required for all SGW NUTs which support NULL as an encryption algorithm)

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
 HOST2_Link3 -- SGW1 -------------------- NUT -- HOST1_Link0
                     -------------------> SA-I
                     <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	NULL
ESP key	
ESP authentication algorithm	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	NULL
ESP key	
ESP authentication algorithm	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

Tunnel source address	NUT_Link1
Tunnel destination address	SGW1_Link2
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Algorithm	NULL
	Key	
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply within ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	NULL
	Key	
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

Procedure：

```
SGW1_Link2(TN)    Target(NUT)    HOST1_Link0(TN)
    |                |              |
    |--------------->|              | ICMP Echo Request within ESP
    |                |              |
    |                |------------->| ICMP Echo Request
    |                |              | (SRC=HOST2_Link3/DST=HOST1_Link0)
    |                |              |      (Judgment  #1)
    |                |              |
    |                |<-------------| ICMP Echo Reply
    |                |              | (SRC=HOST1_Link0/DST=HOST2_Link3)
    |                |              |
    |<---------------|              | ICMP Echo Reply within ESP
    |                |              |      (Judgment  #2)
```

PartA： SA-I
1. SGW1 sends *"ICMP Echo Request within ESP"*
2. Observe the packet transmitted by NUT

PartB： SA-O
3. HOST1 sends *"ICMP Echo Reply"*
4. Observe the packet transmitted by NUT

Judgment：

PartA： Judgment #1
Step-2： NUT transmits *"ICMP Echo Request"*

PartB： Judgment #2
Step-4： NUT transmits *"ICMP Echo Reply within ESP"*

References：

RFC2401 : Security Architecture for the Internet Protocol
RFC2404 : The Use of HMAC-SHA-1-96 within ESP and AH
RFC2406 : IP Encapsulating Security Payload (ESP)
RFC2410 : The NULL Encryption Algorithm and Its Use With IPsec

6.2.7. Tunnel Mode ESP=DES-CBC HMAC-SHA1

Purpose:

SGW tunnel mode, ESP=DES-CBC HMAC-SHA1

Category:

End-Node : N/A
SGW : ADVANCED (This test is required for all SGW NUTs which support DES-CBC as an encryption algorithm)

Initialization:

Use common topology described as Fig.4

Set NUT's SAD and SPD as following:

```
 HOST2_Link3 -- SGW1 -------------------- NUT -- HOST1_Link0
                     -------------------> SA-I
                     <------------------- SA-O
```

Security Association Database (SAD) for SA-I

source address	SGW1_Link2
destination address	NUT_Link1
SPI	0x1000
mode	tunnel
protocol	ESP
ESP algorithm	DES-CBC
ESP key	idesin01
ESP authentication algorithm	HMAC-SHA1
ESP authentication key	ipv6readylogsha1in01

Security Policy Database (SPD) for SA-I

Tunnel source address	SGW1_Link2
Tunnel destination address	NUT_Link1
source address	Link3
destination address	Link0
upper spec	any
direction	in
protocol	ESP
mode	tunnel

Security Association Database (SAD) for SA-O

source address	NUT_Link1
destination address	SGW1_Link2
SPI	0x2000
mode	tunnel
protocol	ESP
ESP algorithm	DES-CBC
ESP key	idesout1
ESP authentication algorithm	HMAC-SHA1
ESP authentication key	ipv6readylogsha1out1

Security Policy Database (SPD) for SA-O

Tunnel source address	NUT_Link1
Tunnel destination address	SGW1_Link2
source address	Link0
destination address	Link3
upper spec	any
direction	Out
protocol	ESP
mode	tunnel

Packets:

ICMP Echo Request within ESP

IP Header	Source Address	SGW1_Link2
	Destination Address	NUT_Link1
ESP	SPI	0x1000
	Algorithm	DES-CBC
	Key	idesin01
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1in01
IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Request

IP Header	Source Address	HOST2_Link3
	Destination Address	HOST1_Link0
ICMP	Type	128 (Echo Request)

ICMP Echo Reply

IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

ICMP Echo Reply within ESP

IP Header	Source Address	NUT_Link1
	Destination Address	SGW1_Link2
ESP	SPI	0x2000
	Algorithm	DES-CBC
	Key	idesout1
	Authentication Algorithm	HMAC-SHA1
	Authentication Key	ipv6readylogsha1out1
IP Header	Source Address	HOST1_Link0
	Destination Address	HOST2_Link3
ICMP	Type	129 (Echo Reply)

Procedure:

```
SGW1_Link2(TN)    Target(NUT)    HOST1_Link0(TN)
    |               |              |
    |-------------->|              | ICMP Echo Request within ESP
    |               |              |
    |               |------------->| ICMP Echo Request
    |               |              | (SRC=HOST2_Link3/DST=HOST1_Link0)
    |               |              |      (Judgment  #1)
    |               |              |
    |               |<-------------| ICMP Echo Reply
    |               |              | (SRC=HOST1_Link0/DST=HOST2_Link3)
    |               |              |
    |<--------------|              | ICMP Echo Reply within ESP
    |               |              |      (Judgment  #2)
```

PartA: SA-I
1. SGW1 sends *"ICMP Echo Request within ESP"*
2. Observe the packet transmitted by NUT

PartB: SA-O
3. HOST1 sends *"ICMP Echo Reply"*
4. Observe the packet transmitted by NUT

Judgment:

PartA: Judgment #1
Step-2: NUT transmits *"ICMP Echo Request"*

PartB: Judgment #2
Step-4: NUT transmits *"ICMP Echo Reply within ESP"*

References:

RFC1829 : The ESP DES-CBC Transform
RFC2401 : Security Architecture for the Internet Protocol
RFC2404 : The Use of HMAC-SHA-1-96 within ESP and AH
RFC2405 : The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC2406 : IP Encapsulating Security Payload (ESP)