IPSecME Working Group
Internet-Draft
Intended Status: Informational
Expires: March 27, 2016

A. Yamaya
Furukawa Network Solution
T. Ohya
NTT
T. Yamagata
KDDI
S. Matsushima
Softbank Telecom
September 24, 2015

Simple VPN solution using Multi-point Security Association
draft-yamaya-ipsecme-mpsa-06

Abstract

This document describes the over-lay network solution by utilizing dynamically established IPsec multi-point Security Association (SA) without individual connection.

Multi-point SA technology provides the simplified mechanism of the Auto Discovery and Configuration function. This is applicable for any IPsec tunnels such as IPv4 over IPv4, IPv4 over IPv6, IPv6 over IPv4 and IPv6 over IPv6.

Status of this Memo

Copyright and License Notice

Table of Contents

1. Introduction

As described in the problem statement document [ad-vpn-problem], dynamic, secure and scalable system for establishing SAs is needed.

With multi-point SA, an endpoint automatically discovers other endpoint. In this draft, an endpoint means an inexpensive CPE, which can hardly establish large number of IPsec sessions simultaneously. The CPEs also share a multi-point SA within the same group, and there is no individual connection between them.

Scalability issue becomes serious in the service, such as triple play which requires large number of sessions at the same time. MPSA enables large scale simultaneous sessions even with inexpensive CPEs, and can avoid scalability issue.

The latency between CPEs can be minimized because of stateless shared multipoint SA, MPSA is suitable for video and voice services which is very sensitive to latency.

It can avoid the exhaustive configuration for CPEs and controllers. No reconfiguration is needed when a new CPE is added, removed, or changed. It can avoid high load on the controllers.

1.1. Terminology

Multi-point SA - This is similar to Dynamic Full Mesh topology described in [ad-vpn-problem]; direct connections exist in a hub and spoke manner, but only one SA for data transfer is shared with all CPEs.

1.2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Motivation

There are two major topologies - Star topology and full-mesh topology - to communicate securely on over-lay network by using IPsec.

Figure.1 shows star topology. The number of IPsec connection is the same as the number of CPEs (CPE). Authentication, Authorization and Accounting (AAA) of each CPE can be achieved on the gateway.

The problem of the star topology is all the traffic go through the gateway, then it causes high load and latency.

```
         +---------------------------------------------+
         |                IPsec Gateway                |
         |                                             |
         |    +--------------(A<->C)--------------+    |
         |    | +---(A<->B)---+   +---(B<->C)---+ |    |
         +---:|-|:-----------:|---|:-----------:|-|:---+
             :| |:           :|   |:           :| |:
             :| |:           :|   |:           :| |:
             :| |:           :|   |:           :| |:
         +---:v-v:---+       :|   |:      +---:v-v:---+
         |           |       :|   |:      |           |
         |   CPE_A   |       :|   |:      |   CPE_C   |
         |           |       :|   |:      |           |
         +-----------+    +--:v---v:--+   +-----------+
                          |           |
                          |   CPE_B   |
                          |           |
                          +-----------+
```

Figure 1

Figure.2 shows Full-mesh topology. There is no gateways. Each CPE establishes IPsec connection independently. The latency on this topology is relatively low compared to star topology.

In large system, there are huge number ((N^2-N)/2) of IPsec connections. AAA of each CPE is hard to manage in this topology. Moreover, when a CPE is added, removed or changed, reconfiguration is needed for all rest of the CPEs.

```
            +-----------+                     +-----------+
            |           |.....................|           |
            |   CPE_A   <-------(A<->C)------->  CPE_C    |
            |           |.....................|           |
            +---: ^ :---+                     +---: ^ :---+
                : | :                             : | :
                : | :        +-----------+        : | :
                : | :........|           |........: | :
                : +-(A<->B)-->  CPE_B    <--(B<->C)-+ :
                :............|           |............:
                             +-----------+
```

Figure 2

The solution in this document eliminates the problems listed above. Figure 3 shows topology of multi-point SA. Traffic between CPEs does not go through the controller, low latency, AAA of each CPE can be achieved, the number of IPsec connection is almost same as star topology, and no reconfiguration is needed for all the rest of CPEs even when a CPE is added, removed or changed. MPSA controller do not necessarily need to be router. It is possible to change MPSA controller for a software, because a communication load which spans IPsec Gateway by multi-point SA is not big.

```
          +-----------------------------------------------+
          |                MPSA Controller                |
          |                                               |
          +---: | :------------: | :------------: | :---+
              : | :            : | :            : | :
              : | :            : | :            : | :
             ----------------------------------------- SA to distribute
              : | :            : | :            : | :  Multi-point SA
              : | :            : | :            : | :
          +---: v :---+    +---: v :---+    +---: v :---+
          |           |    |           |    |           |
          |   CPE_A   |    |   CPE_B   |    |   CPE_C   |
          |           |    |           |    |           |
          +--- ^ ^ ---+    +--- ^ ^ ---+    +--- ^ ^ ---+
          .....| |..............| |..............| |.....
               | |              | |              | |     \
               | +----(A<->B)---+ +---(B<->C)----+ |   Multi-point SA
               +--------------(A<->C)--------------+   for data transfer
          ...............................................
```

Figure 3

3. Procedure

3.1. Sequence

The multi-point SA capability of the remote host is determined by an exchange of Vendor ID payloads. In the IKE_SA_INIT exchange, the Vendor ID payload for this specification is sent if the multi-point SA is used.

```
 CPE                          Controller
-----------                  -----------
 HDR, SAi1, KEi,
    Ni, V(MPSA)  -->
                         <--  HDR, SAr1, KEr,
                                 Nr, [CERTREQ,] V(MPSA)


                                 MPSA: multi-point SA
```

The initial exchange (including IKE_AUTH) is same as [IKEV2], other than Vendor ID payload included in IKE_SA_INIT.

After the initial exchange has finished successfully, a new INFORMATIONAL exchange is used to distribute multi-point SA to the CPE, with the Notify payload of MPSA_PUT that includes cryptographic algorithm, nonce, keying material, SPI and so on. Keys for multi-point SA is generated according to the contents of the Notify payload by the CPE. The response of the Notify payload has empty Encrypted payload.

```
 CPE                          Controller
-----------                  -----------
                         <--  HDR, SK {N(MPSA_PUT)}
 HDR, SK {}  -->
```

3.2. Extended format

3.2.1. Vendor ID

This document defines a new Vendor ID. The content of the payload is described below.

"multi-point SA"

3.2.2. MPSA_PUT

This document defines a new Notify Message Type MPSA_PUT. The Notify Message Type of MPSA_PUT is 40960. Notification Data of MPSA_PUT has a Proposal-substructure-like format. It consists of Transform-substructure-like structures that have following data.

Description	Trans. Type	Reference
Encryption Algorithm (ENCR)	1	RFC5996
Pseudorandom Function (PRF)	2	RFC5996
Integrity Algorithm (INTEG)	3	RFC5996
Nonce (NONCE)	241	
SK_d (SKD)	242	
Lifetime (LIFE)	243	
Rollover time 1 (ROLL1)	244	
Rollover time 2 (ROLL2)	245	

- Nonce - For Transform Type 241, the Transform ID is 1. The attribute contains actual nonce value with attribute type 16384. The size of the Nonce Data is between 16 and 256 octets.

Name	Number
NONCE_NONCE	1

Attribute Type	Value	Attribute Format
Nonce Value	16384	TLV

o SK_d - For Transform Type 242, the Transform ID is 1. The attribute contains actual SK_d value with attribute type 16385. The length of SK_d Data is the preferred key length of the PRF.

Name	Number
SKD_SK_D	1

Attribute Type	Value	Attribute Format
SK_d Value	16385	TLV

o Lifetime - For For Transform Type 243, the Transform ID is 1. The attribute contains actual lifetime value with attribute type 16386. The length of Lifetime Value is 4 octets. Lifetime is stored in seconds as effective time of the multi-point SA.

Name	Number
LIFE_LIFETIME	1

Attribute Type	Value	Attribute Format
Lifetime Value	16386	TLV

o Rollover time 1 - For Transform Type 244, the Transform ID is 1. The attribute contains actual rollover time 1 value with attribute type 16387. The length of Rollover time 1 Value is 4 octets. Rollover time 1 defines activation time delay for new outbound multi-point SA.

Name	Number
ROLL1_ROLLOVER1	1

Attribute Type	Value	Attribute Format
Rollover1 Value	16387	TLV

o Rollover time 2 - For Transform Type 245, the Transform ID is 1. The attribute contains actual rollover time 2 value with attribute type 16388. The length of Rollover time 2 Value is 4 octets. Rollover time 2 defines deactivation time delay for old inbound multi-point SA.

Name	Number
ROLL2_ROLLOVER2	1

Attribute Type	Value	Attribute Format
Rollover2 Value	16388	TLV

Therefore, the format of the MPSA_PUT of the Notify Message is described below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Protocol ID  |   SPI Size    |      Notify Message Type      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Security Parameter Index (SPI)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 (last) or 2 |   RESERVED    |         Proposal Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Proposal Num  |  Protocol ID  |    SPI Size   |Num  Transforms|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Security Parameter Index (SPI)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 (last) or 3 |   RESERVED    |        Transform Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Transform Type |   RESERVED    |          Transform ID         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                      Transform Attributes                     ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 (last) or 3 |   RESERVED    |        Transform Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Transform Type |   RESERVED    |          Transform ID         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                      Transform Attributes                     ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+


~                                                               ~


+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |   RESERVED    |        Transform Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Transform Type |   RESERVED    |          Transform ID         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                      Transform Attributes                     ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The following example shows a N(MPSA_PUT) notification message. The SPIs in the Proposal-like and Tranform-like substructure are the same value. Following values are defined by the example.

```
Protocol: ESP
ENCR:     AES-CBC (256bits)
PRF:      SHA-1
INTEG:    HAMC-SHA-1-96
NONCE:    241
SKD:      242
LIFE:     243
ROLL1:    244
ROLL2:    245
```

```
         0                   1                   2                   3
         0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       /|     0 (last)  |C|  RESERVED   |         Payload Length        |
      / +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Notify |    3 (ESP)    | SPI Size = 4  |           MPSA_PUT            |
      \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       \|                Security Parameter Index (SPI)                 |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       /|    0 (last)   |   RESERVED    |         Proposal Length       |
Pro-    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 posal-| Prop Num = 1  |    3 (ESP)    | SPI Size = 4  |Num  Transforms|
  like +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       \|                Security Parameter Index (SPI)                 |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       /|       3       |   RESERVED    |        Transform Length       |
      / +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  ENCR |   1 (ENCR)    |   RESERVED    |      12 (ENCR_AES_CBC)        |
      \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       \|1|       14 (Key Length)       |             256               |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       /|       3       |   RESERVED    |        Transform Length       |
   PRF +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       \|    2 (PRF)    |   RESERVED    |      2 (PRF_HMAC_SHA1)        |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       /|       3       |   RESERVED    |        Transform Length       |
 INTEG +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       \|   3 (INTEG)   |   RESERVED    |    2 (AUTH_HMAC_SHA1_96)      |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       /|       3       |   RESERVED    |        Transform Length       |
```

```
     / +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    /  |   241 (NONCE) |   RESERVED    |             1                 |
   /   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  NONCE |0|       16384 (Nonce)        |       Attribute Length        |
   \   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    \  |                                                               |
     \ ~                            [Nonce]                            ~
      \|                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      /|       3       |   RESERVED    |        Transform Length       |
     / +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    /  |   242 (SKD)   |   RESERVED    |             1                 |
   /   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  SKD  |0|       16385 (SK_d)          |       Attribute Length        |
   \   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    \  |                                                               |
     \ ~                            [SK_d]                             ~
      \|                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      /|       3       |   RESERVED    |        Transform Length       |
     / +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    /  |  243 (LIFE)   |   RESERVED    |             1                 |
   LIFE +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    \  |0|       16386 (Lifetime)      |     Attribute Length = 4      |
     \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      \|                          [Lifetime]                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      /|       3       |   RESERVED    |        Transform Length       |
     / +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    /  |  244 (ROLL1)  |   RESERVED    |             1                 |
  ROLL1 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    \  |0|       16386 (Lifetime)      |     Attribute Length = 4      |
     \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      \|                        [RolloverTime1]                        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      /|       3       |   RESERVED    |        Transform Length       |
     / +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    /  |  245 (ROLL2)  |   RESERVED    |             1                 |
  ROLL2 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    \  |0|       16386 (Lifetime)      |     Attribute Length = 4      |
     \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      \|                        [RolloverTime2]                        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.3. Multi-point SA Management

3.3.1. Controller

Controller generates a multi-point SA for a group before connecting to any CPEs.

After the initial exchanges have finished, controller distributes the same multi-point SA information to CPEs within the group by sending N(MPSA_PUT).

SPI and Nonce is generated similar way of [IKEv2]. SK_d is generated from random numbers similar to Nonce.

The same SPI value is stored to Notify payload and Proposal-like substructure.

The multi-point SA will not be negotiated between controller and CPE, but will be notified from controller to CPE one way.

Controller initiates rekey before Lifetime expiration. As the Lifetime, controller notifies the effective time left of the multi-point SA.

3.3.2. CPE

After the initial exchange has finished, CPE obtains multi-point SA information by receiving N(MPSA_PUT) from controller. The keys for the multi-point SA are generated in the same procedure described in [IKEv2], except Ni | Nr is replaced by Nonce.

Therefore, KEYMAT is derived by PRF listed below.

```
KEYMAT = prf+(SK_d, Nonce)
```

The multi-point SA is protected in a cryptographic manner by ENCR and

INTEG which uses the generated keys.

The SPI value for the multi-point SA is the same of its in Notify message.

CPE uses the same multi-point SA as both inbound and outbound SAs.

CPE deletes both of inbound and outbound SA when Lifetime is expired.

Rollover time 1, 2 have no meaning when no old multi-point SA exists.

3.3.3. Rekeying

Rekeying should be finished before Lifetime expiration of current multi-point SA. Rekeying of multi-point SA will be performed as follows.

- Controller generates a new multi-point SA
- Controller distributes a new multi-point SA to all CPEs within the group
- CPE replaces the current multi-point SA to new one

CPE replaces multi-point SA using rollover method like [GDOI].

3.4. Forwarding

Each CPE sends and receives encapsulated packets using the multi-point SA.

The destination address of encapsulated packet will be determined with routing information, which can achieved by static configuration or route exchange mechanism such as BGP on encapsulated environment described in [MESH].

It is applicable for any IPsec tunnels such as IPv4 over IPv4, IPv4 over IPv6, IPv6 over IPv4 and IPv6 over IPv6.

4. Peer discovery

MPSA does not provide peer discovery function by itself. However, other mechanism, such as BGP, can be employed with MPSA for automatic peer discovery. One example is a use of BGP, described in [MESH], to learn peer information as next-hops.

4.1 example of MPSA with BGP for route based VPN

Between controller and each peer, IKE_SA and CHILD_SA are established by IKEv2. On the IKE_SA, an MPSA management message (MPSA_PUT) is served from the controller to the peer.

On the CHILD_SA, the controller and the peer establish a iBGP session to exchange route information (NLRIs). Controller can act as a BGP route reflector (RR), which can reflect NLRIs among all iBGP peers of the controller. In other words, the peer can learn all NLRIs advertised by all other peers.

According to [ENCAPS], each peer can advertise ESP peer address as well as conventional NLRIs, all of those can be reflected by RR on the controller.

At this point, each peer can have all other peer addresses as well as route information. The peer can decide a peer address by mean of recursive route lookup from the destination address of a packet to be forwarded. This decision can be made by the peer itself, without any additional communication with the controller.

Instead of [ENCAPS], each peer can also do it by [RNH]. Each peer learns all other peer addresses by BGP Remote-Next-Hop attributes and decides a peer address from a packet to be forwarded, as same as using [ENCAPS].

5. Security Considerations

MPSA uses IKEv2 to protect MPSA management message, MPSA_PUT. Thus, CPEs are authenticated by IKEv2. Using a shared SA for communication between CPEs, MPSA does not provide the following features.
- Data origin authentication
- Anti-replay protection

MPSA itself does not provide access control for user datagrams, but peer discovery may be able to provide access control as well as those of route based VPN. For example, using BGP for peer discovery described in 4.1, access control could be provided by filtering exchanged routes at the controller. In this case, filtering by source address, protocol and ports can not be achieved. If you need it, you could do by other security policy rules as local setting at CPEs .

5.1. Protected by MPSA

- Authenticating CPEs and controller Authentication is provided by IKEv2 with pre-shared key or RSA signature. MPSA management messages are exchanged after IKEv2 negotiation.

- Confidentiality and integrity Packets are encapsulated by ESP, so that MPSA provides confidentiality and integrity against outside of the group, but does not them against members of the group

5.2 Security issues not to be solved by MPSA

5.2.1 Attack from outside of the group

- Anti-replay protection
MPSA does not provide anti-replay protection, because sequence number synchronization between peers needs additional mechanism. Using a closed network as a transport might be effective to mitigate this kind of attacks.

- Leaking a IKE_SA key
If an attacker could sniff packets on a IKE_SA, and key of the SA were leaked, the attacker may get a key of MPSA by decoding a sniffed MPSA_PUT message.

5.2.2 Attack from inside of the group

If there is a malicious CPE or a CPE is hijacked by an attacker, MPSA can be attacked in the following way because MPSA, including cryptograghic key, is shared by all CPEs.

- An attacker can impersonate another CPE. A closed network that prohibits source address spoofing could mitigate the impersonating.

- An attacker can decode packets between the other CPEs if the attacker could sniff packets.

5.3 Forward secrecy and backward secrecy

MPSA MAY be rekeyed when a CPE is removed from the group, for the removed CPE not to access the other CPEs communication after that, or when a CPE is added from the group, for it not to do before that. If not rekeyed, a removed/added CPE could access

5. IANA Considerations

This memo includes no request to IANA.

6. References

6.1. Normative References

[IKEv2] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

6.2. Informative References

[GDOI] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain of Interpretation", RFC 6407, October 2011.

[MESH] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, June 2009.

[ad-vpn-problem] Manral, V. and S. Hanna, "Auto-Discovery VPN Problem Statement and Requirements", RFC 7018, September 2013.

[RNH] Van de Velde, G., Patel, K., Rao, D., Raszuk, R., and Bush, R., "BGP Remote-Next-Hop", draft-vandevelde-idr-remote-next-hop-07, June 2014

[ENCAPS] L. Berger, R. White and E. Rosen, "BGP IPsec Tunnel Encapsulation Attribute", RFC 5566, June 2009.

Authors' Addresses

Arifumi Yamaya

Furukawa Network Solution Corp.
5-1-9, Higashi-Yawata, Hiratsuka
Kanagawa 254-0016, JAPAN
Email: yamaya@fnsc.co.jp

Takafumi Ohya
NTT Corporation
Nishi-shinjuku, Shinjuku-ku,
Tokyo 163-8019, JAPAN
Email: takafumi.ooya@hco.ntt.co.jp

Tomohiro Yamagata
KDDI Corporation
Garden Air Tower
Iidabashi, Chiyoda-ku,
Tokyo 102-8460, JAPAN
Email: to-yamagata@kddi.com

Satoru Matsushima
Softbank Telecom Corp.
1-9-1, Higashi-Shimbashi, Minato-Ku
Tokyo 105-7322, JAPAN
Email: satoru.matsushima@g.softbank.co.jp