Network Working Group
Internet-Draft
Expires: February 2, 2006

S. Sugimoto
Ericsson
F. Dupont
Point6
M. Nakamura
Hitachi

August 2005

PF_KEY Extension as an Interface between Mobile IPv6 and IPsec/IKE draft-sugimoto-mip6-pfkey-migrate-01

Status of this Memo

Copyright Notice

Abstract

This document describes the need for an interface between Mobile IPv6 and IPsec/IKE and show how the two protocols can interwork. We propose a set of extensions to the PF_KEY framework which allows smooth and solid operation of IKE in a Mobile IPv6 environment. The first extension is called PF_KEY MIGRATE and is for migrating the

endpoint addresses of a IPsec Security Association pair in tunnel mode. The second extension is named SADB_X_EXT_PACKET and allows IKE to make the right choice for address selection in bootstrapping process. Both extensions are helpful for assuring smooth interworking between Mobile IPv6 and IPsec/IKE and achieving performance optimization.

Table of Contents

1. Introduction

In Mobile IPv6 [RFC3775], the Mobile Node (MN) and the Home Agent (HA) use some IPsec Security Associations (SAs) in tunnel mode to protect some mobility signaling messages, mobile prefix discovery and optionally payload traffic. Since the MN may change its attachment point to the Internet, it is necessary to update its endpoint address of the IPsec SAs. This indicates that corresponding entry in IPsec databases (Security Policy (SPD) and SA (SADB) databases) should be updated when MN performs movements. In addition, IKE requires treatment to keep its IKE session alive in a Mobile IPv6 environment.

This document describes the need for an interface between Mobile IPv6 and IPsec/IKE and shows how the two protocols can interwork. We propose a set of extensions to the PF_KEY framework [RFC2367] which allows smooth and solid operation of IKE in an Mobile IPv6 environment. The first extension is called PF_KEY MIGRATE and is for migrating the endpoint addresses of the IPsec SAs in tunnel mode. The second extension is named SADB_X_EXT_PACKET and allows IKE to make the right choice in address selection in the bootstrapping process. Both extensions are helpful for assuring smooth interworking between Mobile IPv6 and IPsec/IKE and achieving performance optimization.

2. Needs for Interactions between Mobile IPv6 and IPsec/IKE

The section 4.4 of RFC 3776 [RFC3776] specifies the rules which applies to IKEv1 for MNs and HAs. The first requirement is to run IKE over the Care-of Address (CoA) because the Home Address (HoA) is usable only after the home registration so not yet in the bootstrapping phase.

A tunnel IPsec SA pair protects some signaling messages and optionally all the traffic between the MN and HA. The initial SPD entry uses the HoA for the MN endpoint address and updates this

address to the new CoA at each movement. A tunnel SA pair is created on demand and is updated too. The RFC 3775 [RFC3775] assumes there is an API which performs the update in the SPD and SADB on both the MN and HA. This document is mainly about this API.

Mobile IPv6 specifies a flag named Key Management Mobility Capability bit (K-bit) in Binding Update (BU) and Binding Acknowledgement (BA) messages (section 10.3.1 of [RFC3775]), which indicates the ability of IKE sessions to survive movement. When both the MN and HA agree to use this functionality, the IKE daemons dynamically update the IKE session when the MN moves. In order to realize this, IKE daemons should be notified by Mobile IPv6, and necessary information to

migrate the IKE session should be provided.

Mobile IPv6 may need to make an access to the SPD not only for updating an endpoint address but also for the deletion/insertion of a specific SPD entry. When the MN performs Foreign-to-Home movement, IPsec SAs established between the MN and HA should be deleted, which means that the SPD entry should have no effect any more. On the other hand, when the MN performs Home-to-Foreign movement, the IPsec SAs should be restored. Hence security policy entries that are associated with tunnel mode SAs may dynamically be added/removed (enabled/disabled) in along with MN's movements.

It should be noted that NEMO Basic Support [RFC3963] has similar requirements for the Mobile Router (MR) and MR's HA (MRHA). In NEMO, the MR works just as same as a MN registering its location information to the MRHA and establishes a tunnel (IP-in-IP or IPsec tunnel). When an IPsec tunnel is established between MR and MRHA, the MR serves as a Security Gateway for the nodes connected to the mobile network. The MR is responsible for handling its tunnel endpoint properly.

3. Requirements

Given the need for an interface between Mobile IPv6 and IPsec/IKE, there should be a minimum interface between the two protocols. Followings are the requirements for the interface from a software engineering point of view.

- o Necessary modifications to the existing software, namely Mobile IPv6 and IPsec/IKE, in order to realize proposed mechanisms, should be kept minimum.
- o Proposed mechanism should not be platform dependent. The mechanism should be based on technology which is commonly available on various platform. This seems to be essential for achieving high portability of the implementation which supports proposed mechanisms.

4. PF_KEY Extensions for Mobile IPv6

In order to fulfil the needs and requirements described in Section 2 and Section 3 we propose to extend the PF_KEY framework so that Mobile IPv6 and IPsec/IKE could interact with each other.

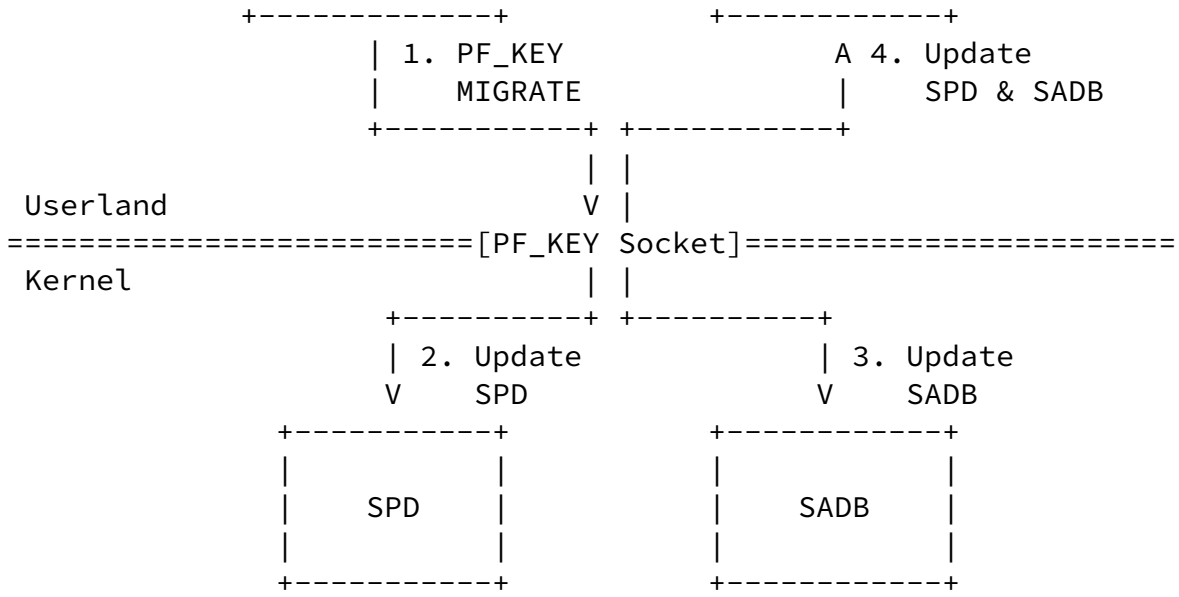
4.1. PF_KEY MIGRATE Message

The first extension is primarily for migrating an endpoint address of

an IPsec SA pair in tunnel mode from one to another, which results in updating IPsec databases. A new PF_KEY message named MIGRATE is introduced for the mechanism.

4.1.1. Overview

The figure below illustrates how Mobile IPv6 and IPsec/IKE components interact with each other using PF_KEY MIGRATE messages in a dynamic keying scenario. On left top, there is a Mobile IPv6 entity. It may be possible that Mobile IPv6 component is completely implemented inside the kernel (this is the case for our implementations because it makes some facilities and extensions far easier at the cost of maintaining a SPD image in daemons). In any case, Mobile IPv6 should be the one which issues the MIGRATE messages. On right top, there is an IKE daemon which is responsible for establishing SAs required for Mobile IPv6 operation. In a manual keying scenario, the difference is only that there is no IKE daemon running on the system.



The primary role of PF_KEY MIGRATE messages is to migrate endpoint addresses of tunnel mode SA pairs requesting IPsec to update its databases (SPD and SADB). In addition, the new message can be used by IKE to enhance its mobility capability. When a PF_KEY MIGRATE message is properly processed by the kernel, it is sent to all open sockets as normal PF_KEY messages. The processing of a sequence of MIGRATE messages is done in following steps:

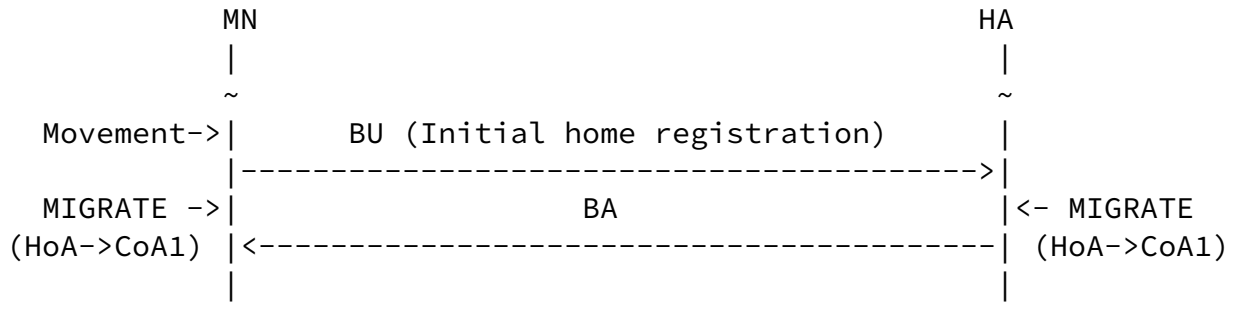
- o Mobile IPv6 issues a PF_KEY MIGRATE message to the PF_KEY socket.
- o The operating system (kernel) validates the message and checks if corresponding security policy entry exists in SPD.
- o When the message is confirmed to be valid, the target SPD entry is updated according to the MIGRATE message. If there is any target SA found that are also target of the update, those should also be updated.
- o After the MIGRATE message is successfully processed inside the kernel, it will be sent to all open PF_KEY sockets.
- o The IKE daemon receives the MIGRATE message from its PF_KEY socket and updates its SPD and SADB images. The IKE daemon may also update its state to keep the IKE session alive.

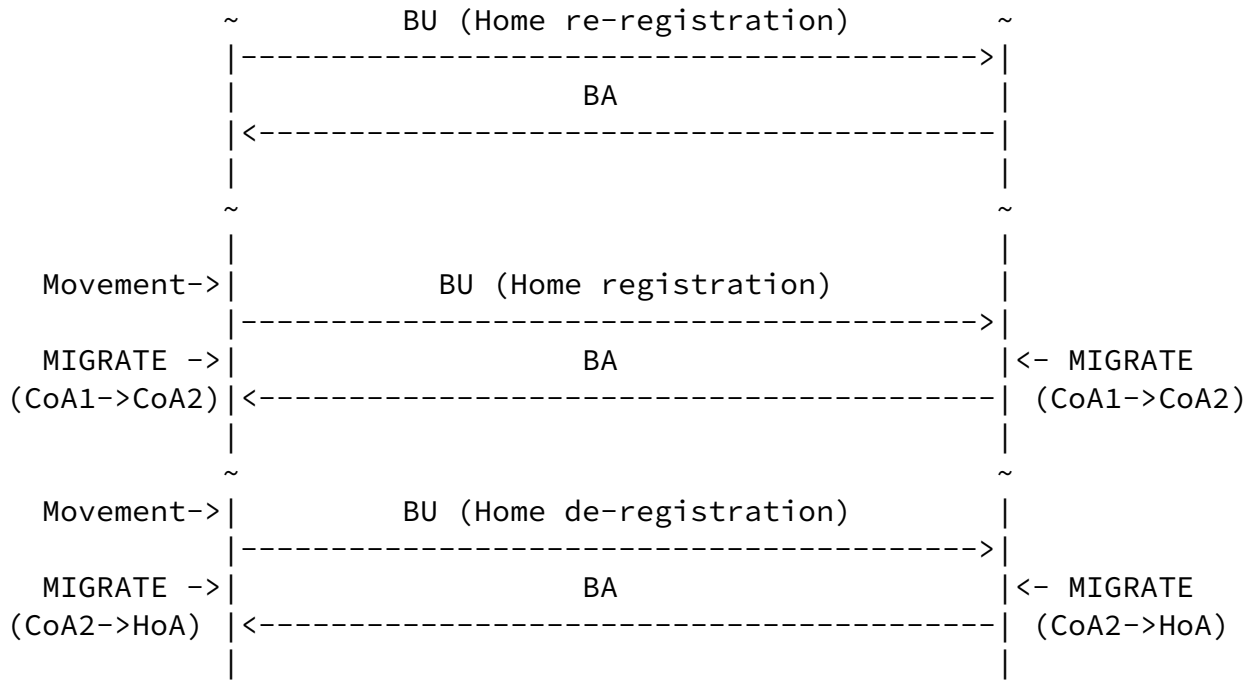
Note that the way IKE maintains its local copy of SPD (the SPD image) is implementation specific issue since there is no standard interface

to access SPD. Some IKE implementation may continuously monitor the SPD inside the kernel. Some IKE implementation may expect notification from the kernel when the SPD is modified. In either way, the proposed mechanism gives a chance for IKE to keep its SPD image up-to-date which is significant in Mobile IPv6 operation.

4.1.2. Message Sequence

Next, we will see how migration takes place in along with home registration. The figure below shows sequence of mobility signaling and PF_KEY MIGRATE messages while the MN roams around links. It is assumed that in the initial state the tunnel endpoint address for a given MN is set as its home address. In the initial home registration, the MN and HA migrate the tunnel endpoint address from the HoA to CoA1. It should be noted that no migration takes place when the MN performs re-registration since the care-of address remains the same. Accordingly, the MN performs movement and changes its primary care-of address from CoA1 to CoA2. A PF_KEY MIGRATE message is issued on both MN and HA for each direction. When the MN returns to home, migration takes place updating the endpoint address with the MN's home address.





4.1.3. Issuing PF_KEY MIGRATE Message

The Mobile IPv6 entity (MN or HA code) triggers the migration by sending a PF_KEY MIGRATE message to its PF_KEY socket. Conceptually, the PF_KEY MIGRATE message should contain following information:

- Selector information:
  - source address/port
  - destination address/port
  - upper layer protocol (i.e., Mobility Header)
  - direction (inbound/outbound)
- Old SA information:
  - old source endpoint address
  - old destination endpoint address
  - IPsec protocol (ESP/AH)
  - mode (Tunnel)
- New SA information:
  - new source endpoint address
  - new destination endpoint address

  - IPsec protocol (ESP/AH)

* mode (Tunnel)

Selector information is required for specifying the target SPD entry to be updated. Basically the information should contain necessary elements which characterize traffic selector as specified in [RFC2401]. With regard to the upper layer protocol, when the Mobile IPv6 stack is not fully aware of IPsec configuration, an wild-card value could be given. In such case, an upper layer protocol information should not be taken into account for searching SPD entry. Plus, the direction of the security policy (inbound/outbound) should be provided. The old SA information is used to specify target security association to be updated. The source and destination addresses of the target entry should be overwritten with the ones included in the new SA information. Note that the IPsec protocol and mode fields should not be updated by a PF_KEY MIGRATE message.

A PF_KEY MIGRATE message should be formed based on security policy configuration and binding record. The selector information and some parts of the SA information (IPsec protocol and mode) should be taken from the policy configuration. The rest of the information should be taken from the sequential binding information. For example, in the case where the MN updates its inbound security policy and corresponding tunnel mode SA pair, the old source address should be set as its previous CoA, and the new source address should be set as its current CoA. Hence, the MN should sequentially keep track of its CoA record. Such information shall be stored in binding update list entry. For the same reason, the HA should keep track of previous CoAs of MNs. Such information shall be stored in binding cache entry.

Additionally, a piece of information which indicates a mobility capability of IKE (K-bit) should be provided by any means. This makes it possible for IKE to see if there is a need to update its state (IKE endpoint addresses) in accordance with PF_KEY MIGRATE messages.

A detailed message format of PF_KEY MIGRATE is provided in Appendix A.

4.1.4. Processing PF_KEY MIGRATE Message

Since a PF_KEY MIGRATE message is applied to a single SPD entry, the kernel should first check validity of the message. If the message is invalid, an EINVAL error MUST be returned as a return value for the write() operation made to the PF_KEY socket. After the validation, the kernel checks if the target SPD entry really exists. If no entry is found, an ENOENT error MUST be returned. If a SPD entry is found

and successfully updated, a success (0) MUST be returned regardless of subsequent result of SADB lookup/update. Note that there may be a case where a corresponding SADB entry does not exist even if a SPD entry is successfully updated. In any error case, a PF_KEY MIGRATE message MUST NOT have any effect on the SPD and SADB.

With respect to the behavior of a normal process (including the IKE daemon) which receives a PF_KEY MIGRATE message from a PF_KEY socket, it SHOULD first check if the message does not include erroneous information. When there is any error indicated, the process MUST silently discard the PF_KEY MIGRATE message. Otherwise, the processing of the message may continue.

4.1.5. Applicability of PF_KEY MIGRATE to Other Systems

It should be noted that the PF_KEY MIGRATE extension is also applicable to other systems than Mobile IPv6 and/or IKEv1. For example, it can be used in a scenario where an IPsec/IKE enabled node establishes tunnel mode SAs association with its Security Gateway while it roams around the network (aka "road warrior"). The security policy is set as such that all traffic should bi-directionally go through the tunnel IPsec SAs. In such case, the migration of a tunnel endpoint address can be handled by PF_KEY MIGRATE messages. Each time the node changes its attachment point to the Internet, PF_KEY MIGRATE messages should be issued to the system. Consequently, the IPsec databases (SPD and SADB) shall be properly updated.

It is also essential to keep design of the mechanism protocol independent. More specifically, the PF_KEY MIGRATE extension should be able to work on both IPv4 and IPv6. In order to achieve this, the IP addresses to be stored in selector and SA information should be handled in a protocol-independent manner.

4.1.6. Limitation of PF_KEY MIGRATE

Currently, a Security Parameter Index (SPI) is not included in the old SA information to specify target SADB entry. This helps to lessen operational burden of Mobile IPv6. However, this simplification can produce ambiguity in searching for the target security association entry. When the unique SPD level is available, it should be use because it avoids this problem both by marking the SAs to update and by limiting SA sharing.

It should be noted that delivery of PF_KEY MIGRATE messages cannot be guaranteed, which is common to other PF_KEY messages. It may be

possible that a MIGRATE message is lost. In such case, there will be inconsistency between the binding record managed by Mobile IPv6 and

IPsec database inside the kernel. Once a PF_KEY MIGRATE message is lost, it would not be possible for the receiver to process some subsequent MIGRATE messages properly. Reinitialization of the Mobile IPv6 stack and IPsec databases may be needed for recovery.

4.2. PF_KEY Packet Extension

In the initial stage of MIPv6 operation known as the bootstrapping process, the MN and HA probably need to establish SAs from scratch in order to start the MIPv6 operation. If IKE is used to maintain the SAs, the MN and HA are required to establish a transport mode SA pair so that the MN could make the initial protected home registration to the HA. As mentioned in RFC 3776 [RFC3776], the IKE negotiation should be done carefully in terms of handling the identity of the MN. More specifically, IKE must be run over the MN's primary CoA while the SA pair should be based on the MN's HoA. Note that the HoA cannot be used prior to the initial home registration. This is an exceptional case of IKE negotiation in a sense that the peer address (the address on which IKE runs) and the IP address to be used as selector for the SAs are different. Since IKE should not be required to maintain mobility state, there is a need to guide IKE to make the right choice for address/identity.

A simple solution for this explicit notification can be provided by extending PF_KEY framework by including information of the triggering packet into SADB_ACQUIRE messages. This extension allows receiver of a SADB_ACQUIRE message to determine which address to use for what purpose, i.e., to recognize the exceptional case as all the needed informations are already in the home registration binding update. As shown below, a SADB_ACQUIRE message MAY contain an extension which contains the triggering packet (the whole packet, information extracted from it by the kernel or as we recommend enough of the beginning of it).

```
<base, address(SD), address(P)*, identity(SD)*,
                        sensitivity*, proposal, packet*>
```

4.2.1. Inserting Packet Extension to SADB_ACQUIRE Message

The IPsec subsystem MAY include a Packet Extension to a SADB_ACQUIRE message when it is triggered by an output of data packet. The Packet Extension simply contains the information of the triggering packet. Like any other extension headers specified in RFC 2367 [RFC2367], a Packet Extension (SADB_X_EXT_PACKET) MUST follow the basic rules and be formulated in the type-length-value format. A redundant part of the original IP packet (i.e., payload/tralier) MAY be eliminated. More than one Packet Extension header MUST NOT be appended to the message. A sadb_x_packet extension header is followed by an IP

packet which has triggered the SADB_ACQUIRE message. Note that the Packet Extension is protocol independent, which means that the triggering packet included in the extension header could be either IPv4 or IPv6. The address family of the triggering packet can be recognized by the first 4 bits of the IP packet.

```
struct sadb_x_packet {
        uint16_t sadb_packet_len;
        uint16_t sadb_packet_exttype;
};
/* sizeof(struct sadb_x_packet) == 4 */
/* followed by an IP packet header which triggered
                                the SADB_ACQUIRE message */
```

4.2.2. Processing SADB_ACQUIRE Message with Packet Extension

A receiver of a SADB_ACQUIRE message with a Packet Extension MAY extract and process the extension header. A MIPv6-aware IKE daemon should be able to process a Packet Extension which includes the IPv6 packet which carries an initial home registration BU message. Such packet includes a home address destination option which contains the primary CoA of the MN and the source address field of the IPv6 header contains the HoA of the MN (note the exact layout depends on the place of the IPsec acquiring code, we assume here its place follows the section 11.3.2 of [RFC3775]). The destination address field of the IPv6 header contains the address of the HA, the mobility header a BU (type 5) for home registration (H flag set to one).

Receipt of SADB_ACQUIRE Message with Packet Extension containing BU message implies that IKE is required to establish SAs for the MIPv6 home registration. Accordingly, the IKE should be able to make a right choice of address selection. The CoA must be used as a peer

address in the IKE negotiation and the HoA should be used as selector of transport mode SAs and as endpoint address of tunnel mode SAs.

4.2.3. Relation of Packet Extension to IKEv2

In IKEv2 [I-D.ietf-ipsec-ikev2], when the initiator has requested to establish SAs triggered by a data packet, the first traffic selector of TSi and TSr should reflect the triggering packet. Therefore, IKEv2 could take advantage of Packet Extensions when some information from triggering packets are needed for a traffic selector negotiation.

5. Necessary Modifications to Mobile IPv6 and IPsec/IKE

In order to realize the proposed mechanism, there are some necessary

modifications to Mobile IPv6 and IPsec/IKE. Following are the summary of necessary modifications, which could be of interest to implementors of Mobile IPv6 and/or IPsec/IKE.

- Modifications to Mobile IPv6:
  - The Mobile IPv6 code can make an access to PF_KEY socket. In particular, the Mobile IPv6 code should have privilege to write messages into a PF_KEY socket.
  - Issuing PF_KEY MIGRATE messages: in order to form MIGRATE messages, it is required that the Mobile IPv6 code has some knowledge of its IPsec configuration and precise binding record. The Mobile IPv6 code may be aware of exact IPsec configuration in form or security policy. It would also be possible that the Mobile IPv6 code is only aware of minimum IPsec configuration whether if IPsec is utilized or not.
- Modifications to IPsec:
  - Processing PF_KEY MIGRATE messages: the kernel should be able to process PF_KEY MIGRATE messages sent by the Mobile IPv6 code. Unless the message is invalid, it should be sent to all open PF_KEY sockets.
  - Enabling Packet Extensions (SADB_X_EXT_PACKET): the kernel should be able to append a SADB_X_EXT_PACKET extension to SADB_MIGRATE messages when they are triggered by an output of a data packet.
- Modifications to IKE:

* Processing PF_KEY MIGRATE messages: the IKE code may update its local copy of IPsec databases (SPD and SADB) in accordance with received PF_KEY MIGRATE messages. In addition, it may update its state / IKE session with new endpoint addresses indicated by PF_KEY MIGRATE messages.
* Processing of Packet Extensions (SADB_X_EXT_PACKET): the IKE code may process SADB_X_EXT_PACKET extensions and extract necessary information from triggering packets. In order for the IKE code to be MIPv6-aware, it should properly extract the home address, care-of address, and HA address from IP packets which carry home registration BU messages.

6. Security Considerations

There is no specific security considerations for the mechanisms introduced by the document but as it should make deployment of dynamic keying in Mobile IPv6 environments easier it should improve the security of such environments. Note that dynamic keying is known to be more secure (it provides anti-replay for instance) and far more scalable.

7. Conclusion

- There is a need for Mobile IPv6 and IPsec/IKE to interact with each other to provide full support of IPsec security functions.
- An extension to the PF_KEY framework (PF_KEY MIGRATE message) is proposed, which makes it possible for the IPsec/IKE to migrate an endpoint address of tunnel IPsec SAs from one to another.
- PF_KEY MIGRATE messages also make it possible for IKE to survive movements by updating its IKE session.
- In order for the IKE to perform key negotiations and rekeying, effort should be made to keep its SPD image up-to-date.
- The proposed mechanism was implemented on both Linux and BSD platforms and confirmed to be working well.
- Currently, large portion of the proposed mechanism is implementation dependent due to lack of standard interface to access the SPD (PF_POLICY?).

8. References

[I-D.ietf-ipsec-ikev2]
Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", draft-ietf-ipsec-ikev2-17 (work in progress), October 2004.

[I-D.ietf-ipsec-rfc2401bis]
Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", draft-ietf-ipsec-rfc2401bis-06 (work in progress), April 2005.

[RFC2367] McDonald, D., Metz, C., and B. Phan, "PF_KEY Key Management API, Version 2", RFC 2367, July 1998.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

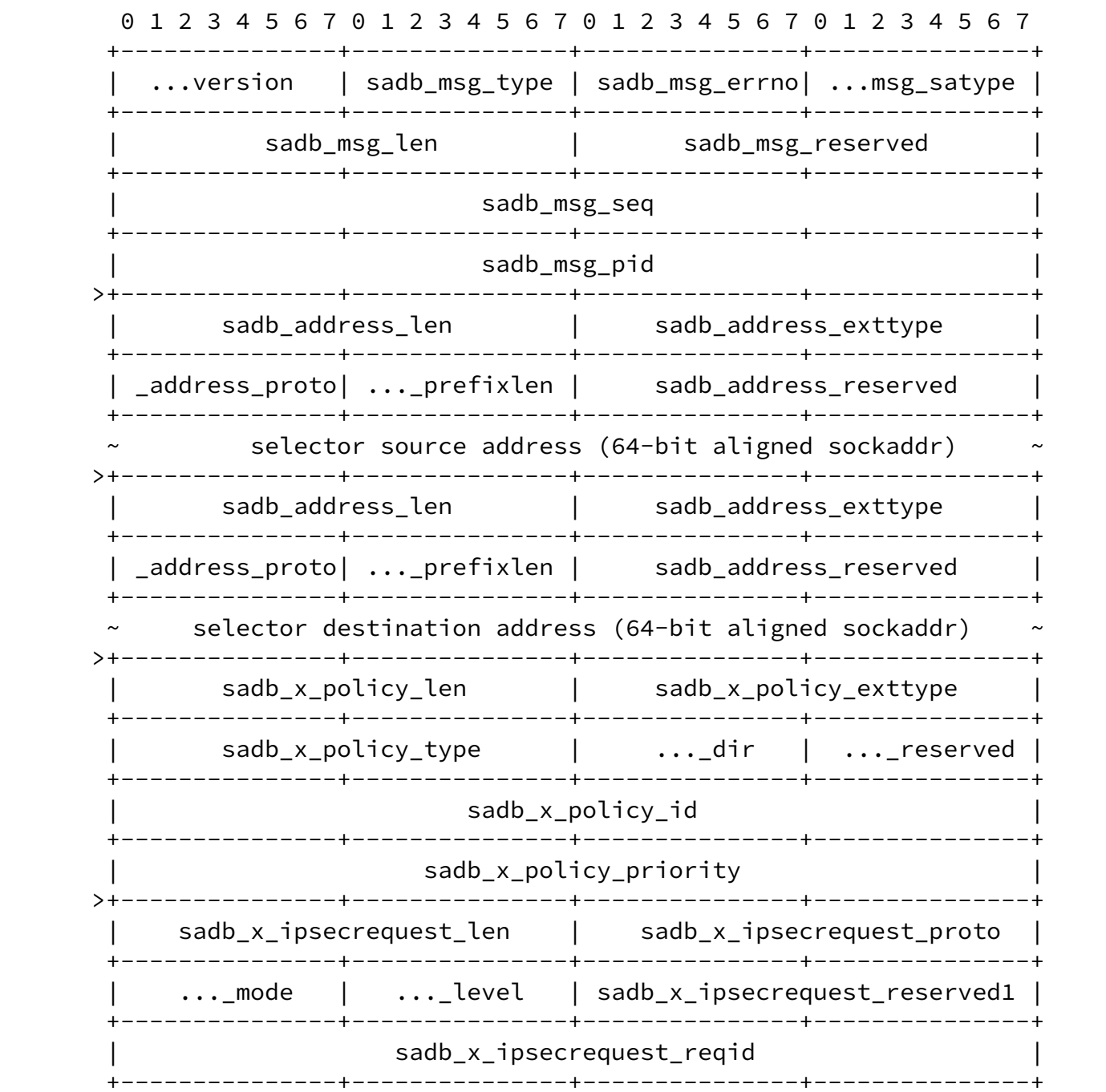
[RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

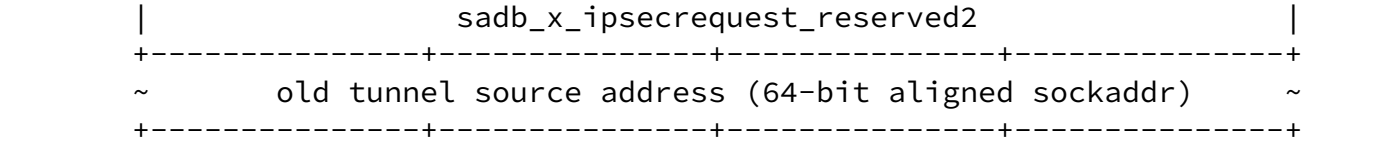
[RFC3776] Arkko, J., Devarapalli, V., and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", RFC 3776, June 2004.

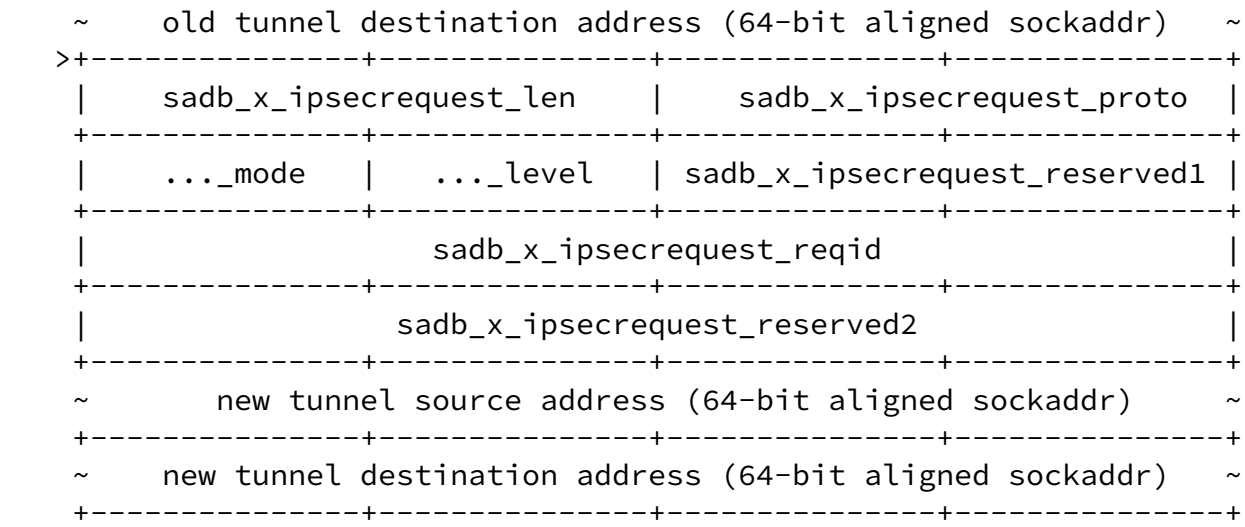
[RFC3963] Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", RFC 3963, January 2005.

Appendix A. PF_KEY MIGRATE Message Format

The figure below shows the message format of PF_KEY MIGRATE. The message consists of 6 parts (boundary of each part is marked with ">"). The message starts with PF_KEY base message header followed by two address extensions. A pair of address extensions hold source and destination address of the selector. Rest of the message are specific to IPsec implementation on BSD. sadb_x_policy{} structure holds additional information of security policy. The last part of the message is a pair of sadb_x_ipsecrequest{} structures that hold old and new SA information.


0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
...version
sadb_msg_type
sadb_msg_errno
...msg_satype
sadb_msg_len
sadb_msg_reserved
sadb_msg_seq
sadb_msg_pid
sadb_address_len
sadb_address_exttype
_address_proto
..._prefixlen
sadb_address_reserved
selector source address (64-bit aligned sockaddr)
sadb_address_len
sadb_address_exttype
_address_proto
..._prefixlen
sadb_address_reserved
selector destination address (64-bit aligned sockaddr)
sadb_x_policy_len
sadb_x_policy_exttype
sadb_x_policy_type
..._dir
..._reserved
sadb_x_policy_id
sadb_x_policy_priority
sadb_x_ipsecrequest_len
sadb_x_ipsecrequest_proto
..._mode
..._level
sadb_x_ipsecrequest_reserved1
sadb_x_ipsecrequest_reqid


sadb_x_ipsecrequest_reserved2
old tunnel source address (64-bit aligned sockaddr)



Following is a structure of PF_KEY base message header specified in [RFC2367]. A new message type for PF_KEY MIGRATE (i.e., SADB_X_MIGRATE) should be specified in member sadb_msg_type.

```
struct sadb_msg {
        uint8_t         sadb_msg_version;
        uint8_t         sadb_msg_type;
        uint8_t         sadb_msg_errno;
        uint8_t         sadb_msg_satype;
        uint16_t        sadb_msg_len;
        uint16_t        sadb_msg_reserved;
        uint32_t        sadb_msg_seq;
        uint32_t        sadb_msg_pid;
};
```

Following is a structure of address extension header specified in [RFC2367]. Upper layer protocol should be specified in member sadb_address_proto.

```
struct sadb_address {
        uint16_t        sadb_address_len;
        uint16_t        sadb_address_exttype;
        uint8_t         sadb_address_proto;
        uint8_t         sadb_address_prefixlen;
        uint16_t        sadb_address_reserved;
};
```

Following is a structure for holding attributes that are relevant to security policy, which is available on BSD IPsec implementation.

Direction of the target security policy should be specified in member sadb_x_policy_dir.

```
struct sadb_x_policy {
        uint16_t        sadb_x_policy_len;
        uint16_t        sadb_x_policy_exttype;
        uint16_t        sadb_x_policy_type;
        uint8_t         sadb_x_policy_dir;
        uint8_t         sadb_x_policy_reserved;
        uint32_t        sadb_x_policy_id;
        uint32_t        sadb_x_policy_priority;
};
```

Following is a structure for holding attributes that are relevant to security association, which is available on BSD IPsec implementation. IPsec protocol (ESP or AH) and mode (Tunnel) of the target security association should be provided in member sadb_x_ipsecrequest_proto and sadb_x_ipsecrequest_mode, respectively.

```
struct sadb_x_ipsecrequest {
        uint16_t        sadb_x_ipsecrequest_len;
        uint16_t        sadb_x_ipsecrequest_proto;
        uint8_t         sadb_x_ipsecrequest_mode;
        uint8_t         sadb_x_ipsecrequest_level;
        uint16_t        sadb_x_ipsecrequest_reserved1;
        uint32_t        sadb_x_ipsecrequest_reqid;
        uint32_t        sadb_x_ipsecrequest_reserved2;
};
```

Appendix B. Acknowledgements

The authors gratefully acknowledge the contribution of: Kazunori Miyazawa, Noriaki Takamiya, Shoichi Sakane, Mitsuru Kanda, Keiichi Shima, Tsuyoshi Momose and other members from USAGI Project and KAME Project.

Authors' Addresses

Shinta Sugimoto
Nippon Ericsson K.K.
Koraku Mori Building
1-4-14, Koraku, Bunkyo-ku
Tokyo 112-0004
Japan

Phone: +81 3 3830 2241
Email: shinta.sugimoto@ericsson.com

Francis Dupont
Point6
c/o GET/ENST Bretagne
2 rue de la Chataigneraie
CS 17607
35576 Cesson-Sevigne Cedex
France

Fax: +33 2 99 12 70 30
Email: Francis.Dupont@enst-bretagne.fr

Masahide Nakamura
Hitachi Communication Technologies, Ltd.
216 Totsuka-cho, Totsuka-ku
Yokohama 244-8567
Japan

Phone: +81 45 865 7003
Email: masahide_nakamura@hitachi-com.co.jp

Intellectual Property Statement

Disclaimer of Validity

WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.