Network Working Group	A. Ebalard	
Internet-Draft	EADS	
Intended status: Informational	S. Decugis	
Expires: April 3, 2011	NICT	
	September 30, 2010	

TOC

PF_KEY Extension as an Interface between Mobile IPv6 and IPsec/IKE
draft-ebalard-mext-pfkey-enhanced-migrate-01

Abstract

This document describes the need for an interface between Mobile IPv6 and IPsec/IKE and shows how the two protocols can interwork. An extension of the PF_KEY framework is proposed which allows smooth and solid operation of IPsec/IKE in a Mobile IPv6 environment.
This document is heavily based on a previous draft [MIGRATE] written by Shinta Sugimoto, Masahide Nakamura and Francis Dupont. It simply reuses the MIGRATE mechanism defined in the expired document, removes a companion extension (SADB_X_EXT_PACKET) based on implementation feedback (complexity, limitations, ...) and fills the gap by very simple changes to MIGRATE mechanism. This results in a more simple and consistent mechanism, which also proved to be easier to implement. This document is expected to serve as a continuation of [MIGRATE] work. For that reason, the name of the extension has been kept.
PF_KEY MIGRATE message serves as a carrier for updated information for both the in-kernel IPsec structures (Security Policy Database / Security Association Database) and those maintained by the key managers. This includes in-kernel Security Policy / Security Association endpoints, key manager maintained equivalents, and addresses used by IKE_SA (current and to be negotiated). The extension is helpful for assuring smooth interworking between Mobile IPv6 and IPsec/IKE for the bootstrapping of mobile nodes and their movements.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as “work in progress.”
This Internet-Draft will expire on April 3, 2011.

Copyright Notice

Table of Contents

---

1. Introduction

TOC

In Mobile IPv6 (Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6," June 2004.) [RFC3775], the Mobile Node (MN) and the Home Agent (HA) use some IPsec Security Associations (SA):

- in transport mode to protect signaling traffic (Binding Update and Binding Ack). Those SA reference the Home Address (HoA) of the MN.
- in tunnel mode to protect some mobility signaling messages, mobile prefix discovery and optionally payload traffic. Those SA reference both the Care-of Address (CoA) and the HoA of the MN.

To negotiate initial transport mode SA, the IKE daemon needs to be directed to use current CoA as source of the IKE exchanges. By default, the (currently unusable) HoA would be used.
Later, since the MN may change its attachment point to the Internet, it is necessary for it to update the tunnel endpoint address of its IPsec SA. This indicates that corresponding entries in IPsec databases (Security Policy (SPD) and Security Association (SAD) databases) should be updated when MN performs movements.
In a Mobile IPv6 environment, the key manager (KM) also needs to be notified when the SPD and SAD are updated. More generally, it needs to be provided with updated addresses for already negotiated and future IKE_SA. Because of its role and unlike common applications, a key manager has to take part to the mobility process it secures: it needs to be aware of address changes.
This document describes the need for an interface between Mobile IPv6 and IPsec/IKE and shows how the two protocols can interwork. An extension to the PF_KEY framework (McDonald, D., Metz, C., and B. Phan, "PF_KEY Key Management API, Version 2," July 1998.) [RFC2367] which allows smooth and solid operation of IKE in a Mobile IPv6 environment is defined. The extension is called PF_KEY MIGRATE and serves as a carrier for the necessary information for both the in-kernel IPsec stack and the key managers.
For the IPsec stack, this allows migrating the endpoint addresses of the IPsec SA (and associated SP). For the key managers, this allows the mirrored structures to be updated (SAD and SPD). This also allows the addresses of already negotiated and associated IKE_SA to be migrated, and to make specific addresses available for negotiations of future IKE_SA. This set of operations performed by the key manager on its internal structures is initiated by the MIPv6 process.
With the extension, the bootstrapping of the MN appears as a common operation for IKE, by having the right addresses needed for the negotiation available prior to its beginning (i.e. at the reception of the PF_KEY ACQUIRE message by the IKE daemon).

The extension is helpful for assuring smooth interworking between Mobile IPv6 and IPsec/IKE and achieving performance optimization: upon movement, both sides (MN and HA) locally notify the IPsec stack and the key manager of the new CoA, thus preventing the need to flush and renegotiate existing SA.
As stated in the abstract, this document is heavily based on the content of a previous draft MIGRATE (Sugimoto, S., Nakamura, M., and F. Dupont, "PF_KEY Extension as an Interface between Mobile IPv6 and IPsec/IKE," December 2007.) [MIGRATE]. This expired memo served as the basis for this work both from technical and editorial standpoints. Numerous technical discussions with some of its authors took place while working on this memo and associated implementations.

---

2. Terminology

TOC

In this document, the term IKE implicitly stands for both IKEv1 [RFC2409] (Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)," November 1998.) and IKEv2 [RFC5996] (Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol version 2 (IKEv2)," September 2010.). IKEv2 terminology is used preferentially when describing actions performed by the key manager but they also apply to the IKEv1 counterparts. For instance, when actions occur on IKE_SA, they also apply to ISAKMP SA for IKEv1, except otherwise specified. The terms "IKE daemon" and "Key Manager (KM)" are used interchangeably in the document.
The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] (Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels," March 1997.).

---

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

TOC

Sections 4.4 of [RFC3776] (Arkko, J., Devarapalli, V., and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents," June 2004.) and [RFC4877] (Devarapalli, V. and F. Dupont, "Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture," April 2007.) specify the rules which apply to IKE for MN and HA. The first requirement is to run IKE over the Care-of Address because the Home Address is usable only after the home registration but not yet in the bootstrapping phase, when Transport mode IPsec SA are commonly negotiated to protect BU/BA.
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. RFC 3775 (Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6," June 2004.) [RFC3775] assumes there is an API which performs the update in the SPD and SAD on both the MN and HA, and notify the IKE daemon. This memo proposes such an API based on PF_KEY framework both to document the needs and ease interoperability between components which may be provided by different vendors.
Mobile IPv6 may need to make an access to the SPD not only for updating an endpoint address but also for deleting/inserting a specific SPD entry. When the MN performs Foreign-to-Home movement, IPsec SA established between the MN and HA to protect data traffic should be deleted, and associated SPD entries should have no effect anymore. On the other hand, when the MN performs Home-to-Foreign movement, those IPsec SP should be restored. Hence security policy entries that are associated with tunnel mode SA may dynamically be added/removed (enabled/disabled) in along with MN's movements. As a side note for such a scenario, Home Link detection mechanism becomes critical security-wise [hld-sec] (Ebalard, A., "Mobile IPv6 Home Link Detection Mechanism Security considerations," April 2009.).
It should be noted that NEMO Basic Support (Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," January 2005.) [RFC3963] has similar requirements for the Mobile Router (MR) and MR's HA (MRHA). In NEMO, the MR works just like 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.

---

4. Requirements

TOC

Despite the need for an interface between Mobile IPv6 and IPsec/IKE, it should be kept simple. Following are the requirements for the interface from a software engineering point of view.

- Necessary modifications to the existing software, namely Mobile IPv6 and IPsec/IKE, in order to realize proposed mechanisms, should be kept minimum.

- Proposed mechanism should not be platform dependent. The mechanism should be based on technology which is commonly available on various platforms. This seems to be essential for achieving high portability of the implementation which supports proposed mechanisms.

5. PF_KEY Extensions for Mobile IPv6: PF_KEY MIGRATE Message

TOC

In order to fulfill the needs and requirements described in Section 3 (Needs for Interactions between Mobile IPv6 and IPsec/IKE) and Section 4 (Requirements) an extension of PF_KEY framework is proposed so that Mobile IPv6 and IPsec/IKE can interact with each other. The new message dedicated to that function is called MIGRATE. A new simple PF_KEY structure (struct sadb_x_kmaddress, see Appendix A (PF_KEY MIGRATE Message Format)) is also defined to be used by MIGRATE to serve the purpose of IKE_SA update.

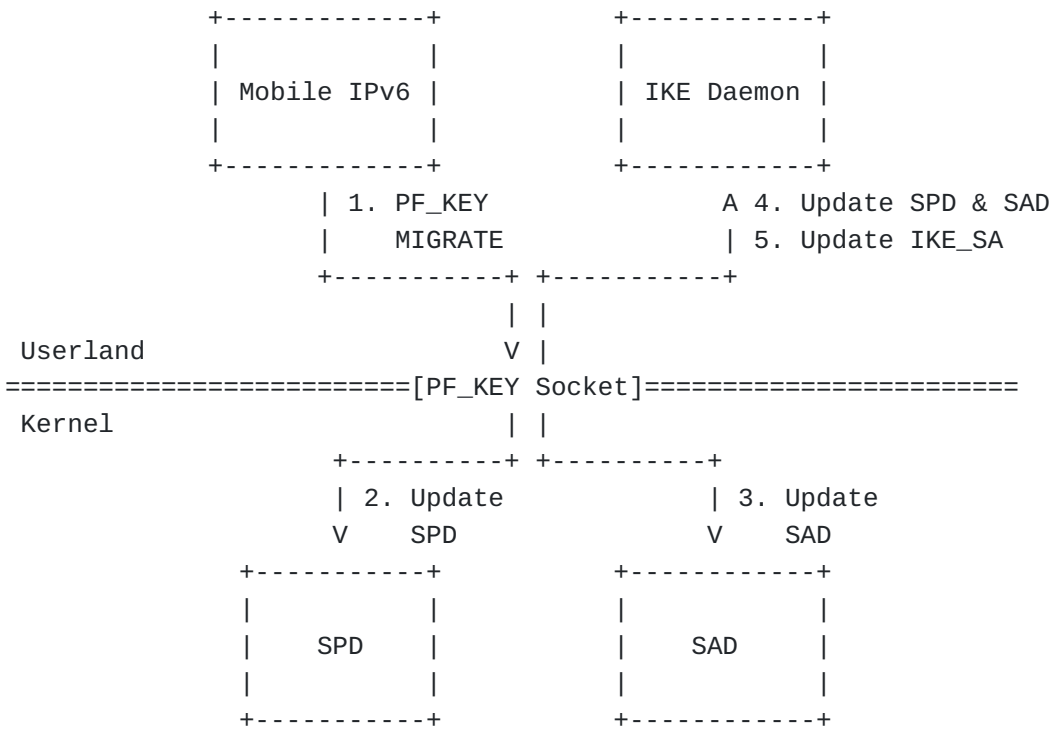
5.1. Overview

TOC

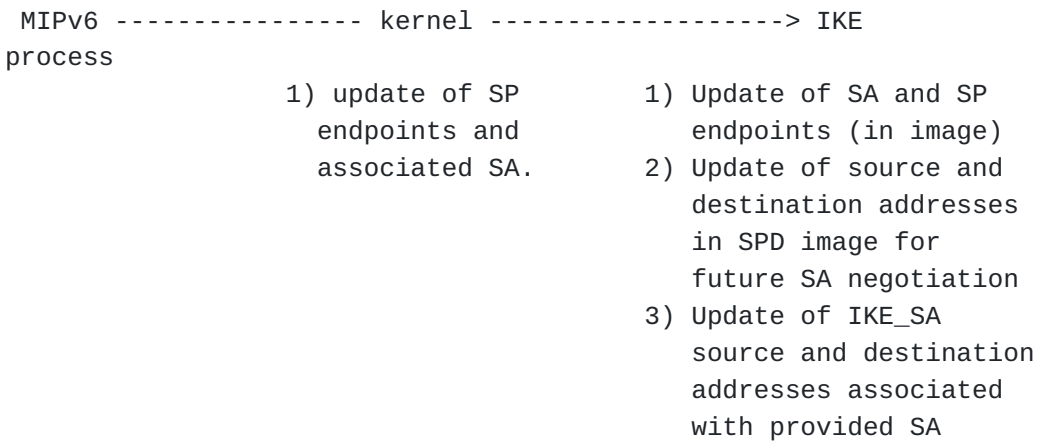
5.1.1. System Overview

TOC

The MIGRATE message is used for providing updated information to its two targets, the kernel IPsec stack and the key manager (when used). The figure below illustrates how Mobile IPv6 and IPsec/IKE components interact with each other using PF_KEY MIGRATE message in a dynamic keying scenario. On left top is a Mobile IPv6 entity (it may be possible that Mobile IPv6 component is completely implemented inside the kernel). In any case, Mobile IPv6 should be the one issuing the MIGRATE message. On right top is an IKE daemon which is responsible for establishing SA required for Mobile IPv6 operation. In a manual keying scenario, the difference is mainly that there is no IKE daemon running on the system.



In the kernel, the primary role of PF_KEY MIGRATE message is to change the endpoint addresses of SA, i.e. requesting IPsec to update its databases (SPD and SAD). Even if tunnel mode is the primary target for MIPv6, MIGRATE is not limited to that mode. Then, after proper processing by the kernel, the MIGRATE message is sent to all open PF_KEY socket. A listening key manager processes it , which results in a possible update of its internal structures. The specific actions are introduced on the following figure.



In more details, the processing of a MIGRATE message can be divided in following steps:

*Mobile IPv6 issues a PF_KEY MIGRATE message to the PF_KEY socket.

*The operating system (kernel IPsec stack) validates the message and checks if corresponding security policy entry exists in SPD.

*When the message is confirmed to be valid, the SPD entry is updated according to the MIGRATE message. If there is any target SA found that is also target of the update, it is also updated. This is detailed in Section 5.3 (Processing PF_KEY MIGRATE Message).

*After the MIGRATE has been successfully processed inside the kernel, it is sent to all open PF_KEY sockets.

*The IKE daemon receives the MIGRATE message from its PF_KEY socket and validates it.

*The key manager starts by updating the SP entries described in the message with the updated endpoint information. It also updates in its SPD image the local and remote addresses to be used for future negotiation of SA associated with those SP (addresses used by future IKE_SA). Then, it updates the SA related information: the endpoints of already negotiated SA and the local and remote values of associated IKE_SA.

Note that the way IKE maintains its local copy of SPD (the SPD image) is an implementation specific issue since there is no standard interface to access SPD. Some IKE implementations may continuously monitor the SPD inside the kernel. Some IKE implementation may expect notifications 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.

5.1.2. Bootstrapping

TOC

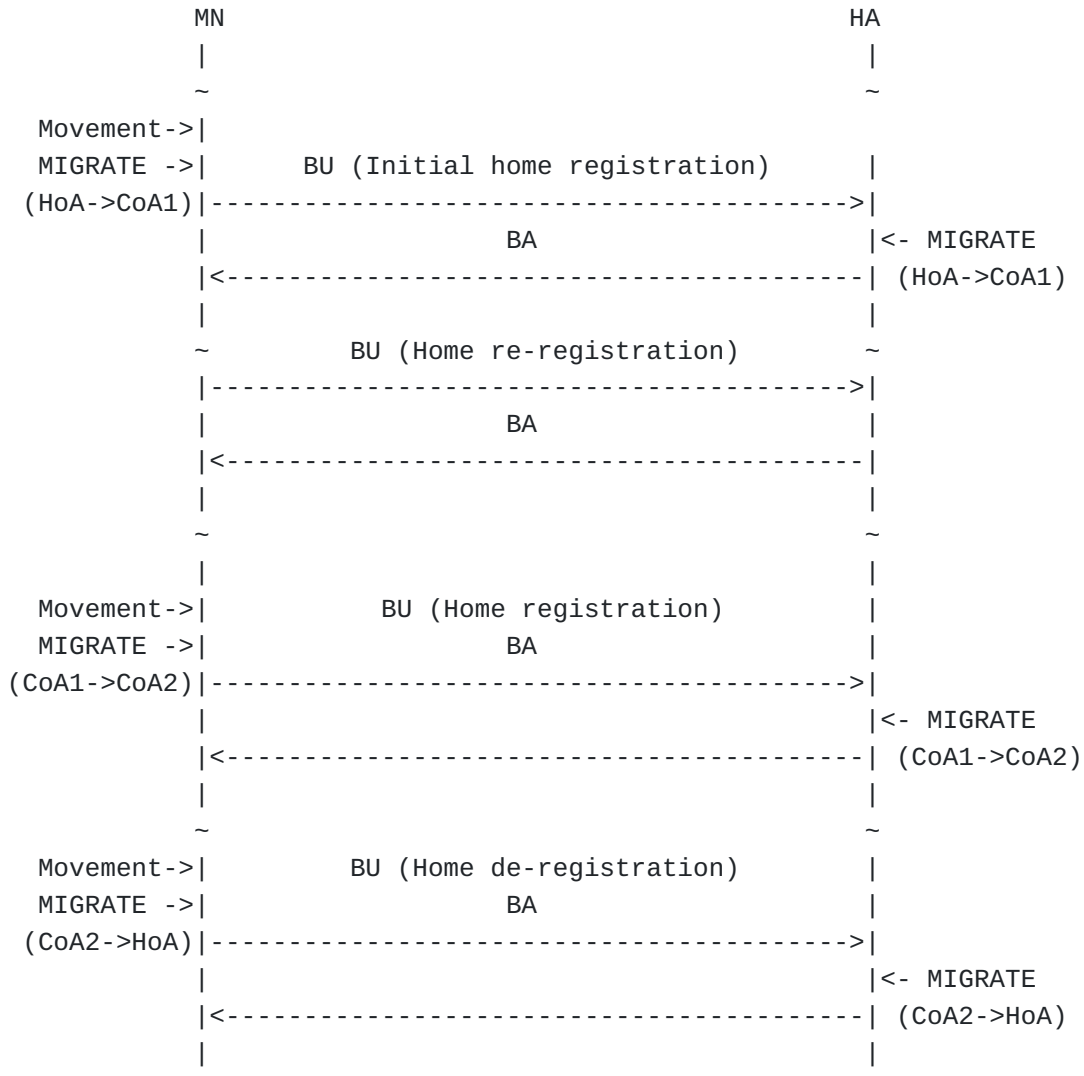
In the bootstrapping stage of Mobile IPv6, the MN and the HA need to establish IPsec SA to protect signaling messages of Mobile IPv6 such as BU and BA. When IKE is used to establish and maintain the SA pairs, the IKE negotiation is the very first transaction made between the MN and the HA.
As mentioned in [RFC3776] (Arkko, J., Devarapalli, V., and F. Dupont, “Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents,” June 2004.), some care is needed for the address management during IKE negotiations in Mobile IPv6 environments. In particular, IKE negotiation to be made to establish a transport mode IPsec SA pair is tricky because the local IKE_SA address and the SA endpoint on the MN side (the Home Address) are different. This is because the Home Address cannot be used prior to the initial home registration. SADB_X_EXT_KMADDRESS extension defined in this memo

enables the MIPv6 module to notify the IKE module about the IKE endpoints.
A simple solution to make the key manager aware that a different address must be used for the negotiation of SA is to have it record this address within its mirrored SPD entries as soon as it becomes available. With that information, the key manager is able to inflect its usual processing where it selects by default the source address of the SA for the negotiation (i.e. as local address of the IKE_SA). By having the MIGRATE message emitted by the Mobile IPv6 process before the emission of the BU, the address is already available to the key manager when the ACQUIRE message is received.
Even if the bootstrapping process initially appears differently than the usual process, having the internal structure of the key manager explicitly record the address (to be used for the negotiation of the SA for a specific SP) allows to keep things simple. The only requirement is that the MIGRATE message be emitted by the Mobile IPv6 process before it sends its Binding Update.

5.1.3. Movement

TOC

Next, we will see how migration takes place along with home registration. The figure below shows a 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 both on MN and HA for each direction. When the MN returns home, migration takes place updating the endpoint address with the MN's home address.
With regard to the timing of issuing the MIGRATE message on the MN during a handover, it must occur immediately before the emission of the binding update performing the home registration (as for bootstrapping). It is possible that ESP-protected (IPsec tunneled) user traffic be sent from the new CoA which is not known to the HA yet. As the HA processes the packets protected under IPsec, and as far as it finds a valid SA, then those packets will be authenticated regardless of their source IP address. In the end, there is no security issue in updating the IPsec SA endpoint while sending the BU and no reason not to do it.
Furthermore, this may help the MN to minimize the packet loss of its outbound traffic during the handover.



5.1.4. IKE_SA Update

TOC

The bootstrapping process described in Section 5.1.2 (Bootstrapping) allows the creation of the SA by having the right source address available to the key manager before the beginning of the negotiation. When the SA has been negotiated, some further exchanges are expected to happen during the lifetime of the SA, including rekeying related exchanges. After the first movement (and obviously further ones), the address used during the bootstrapping process becomes invalid. Even if the SPD and SAD entries are updated (as described in Section 5.1.1 (System Overview)), there is also a need for the key manager to update the addresses used by the IKE_SA.

When the key manager processes the MIGRATE message, it uses the local and remote address information provided by the sadb_x_kmaddress structure to update:

*local copy of the SP entry maintained by the IKE daemon which is specified in the MIGRATE message (as described in Section 5.1.2 (Bootstrapping)).

*the existing IKE_SA associated with the SP entry which is specified by the MIGRATE message.

5.2. Issuing PF_KEY MIGRATE Message

TOC

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:

- Key manager address information
  - source address
  - destination address

  For IKE only
- 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/Transport)

  For IKE and IPsec stack
- New SA information:
  - new source endpoint address
  - new destination endpoint address
  - IPsec protocol (ESP/AH)
  - mode (Tunnel/Transport)

Key manager address information content (source and destination address) is recorded in the associated entry of the SPD image. Those SHOULD be used from now on by the key manager for SA negotiation associated with that SP. The information SHOUD also be used by the key manager to update the local and remote addresses of the IKE_SA (used by already negotiated SA associated with the SP).
Selector information is required to specify the target SPD entry to be updated. Basically the information should contain necessary elements which characterize traffic selector as specified in the IPsec architecture ([RFC2401] (Kent, S. and R. Atkinson, “Security

Architecture for the Internet Protocol," November 1998.), [RFC4301] (Kent, S. and K. Seo, "Security Architecture for the Internet Protocol," December 2005.)). With regard to the upper layer protocol, when the Mobile IPv6 stack is not fully aware of IPsec configuration, a wildcard value can 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, along with old locator information is used to specify target SA to be updated. For tunnel mode, the endpoint addresses refer to the source and destination IP addresses that appear in the IP header, and those should be provided by the MIGRATE message. For transport mode, we require it to be present to keep a fixed message format. For all modes, the address information represents the locators of the SA. For transport mode, it must match the addresses provided in the selector. For tunnel mode, it is obviously not required.

The source and destination addresses (locators) of the target entry should be overwritten. New locator values should also be used to update SP. 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. In previous scenario, the source and destination entries of the address information for the key manager should respectively be set to the CoA and the address of the HA.

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

5.3. Processing PF_KEY MIGRATE Message

TOC

Since a PF_KEY MIGRATE message is applied to a single SPD entry, the kernel should first check validity of the message. During this process, the content of sadb_x_kmaddress structure is skipped, because its content is intended for the key manager and is simply relayed by the kernel.

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 SAD lookup/update. Note that there may be cases where a corresponding SAD 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 SAD.
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. This implies that the kernel is the only entity responsible for returning a status regarding message validation.

5.4. NAT Traversal

TOC

Dual Stack Mobile IPv6 [DSMIPv6] (Soliman, H., “Mobile IPv6 support for dual stack Hosts and Routers,” June 2009.) supports a scenario where a MN is connected to an IPv4 network behind a Network Address Translator (NAT). In such case, the MN assigns an IPv4 private address to its network interface but it is still capable of registering its care-of address to the HA, using the NAT Traversal technique [RFC3948] (Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, “UDP Encapsulation of IPsec ESP Packets,” January 2005.). The MN and HA may set up an IPsec tunnel to protect data and return routability traffic.
The PF_KEY MIGRATE mechanism described in this document does not support [DSMIPv6] (Soliman, H., “Mobile IPv6 support for dual stack Hosts and Routers,” June 2009.) operations. Even if it may be possible to extend it to support DSMIPv6, it is left for future work. The main reasons for that decision are:

- the current complexity of IPsec and IKE NAT-T implementations, including system specific differences.
- the current lack of feedback and available complete implementation of DSMIPv6 on which to implement and test extensions of MIGRATE to support DSMIPv6.

TOC

5.5. Limitations of PF_KEY MIGRATE

A Security Parameter Index (SPI) is not included in the old SA information to specify target SAD 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 used because it avoids this problem both by marking the SA 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 (even if highly unlikely) that a MIGRATE message be lost. In such case, there will be inconsistency between the binding record managed by Mobile IPv6 and IPsec database inside the kernel or the IKE daemon. 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.

---

6. Necessary Modifications to Mobile IPv6 and IPsec/IKE

TOC

In order to realize the proposed mechanism, there are some necessary modifications to Mobile IPv6 and IPsec/IKE. They are listed below for implementors of Mobile IPv6 and/or IPsec/IKE.

*Modifications to Mobile IPv6:

- The Mobile IPv6 code needs to 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 send 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 of security policy. It would also be possible that the Mobile IPv6 code is only aware of minimum IPsec configuration whether IPsec is used or not.

- With regard to the emission of the MIGRATE message during the home registration, the Mobile IPv6 code need to emit it before issuing the Binding Update.

*Modifications to IPsec stack:

  -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.

*Modifications to IKE (associated with processing of MIGRATE):

  -the IKE code needs to update its local copy of IPsec databases (SPD and SAD) in accordance with received PF_KEY MIGRATE message.

  -the IKE code needs to update its associated IKE_SA with new local and remote addresses specifically provided in PF_KEY MIGRATE messages (in sadb_x_kmaddress structure). It also needs to maintain in its SPD the addresses to be used for future negotiation of IKE_SA.

---

7. Implementation

TOC

The mechanism described in this memo has been implemented for Linux:

*Linux kernel IPsec stack: the mechanism is fully implemented since version 2.6.28 (released in December 2008) both for PF_KEY (as described in this memo) and Linux native interface (Netlink, see [RFC3549] (Salim, J., Khosravi, H., Kleen, A., and A. Kuznetsov, "Linux Netlink as an IP Services Protocol," July 2003.)) with in-kernel XFRM transformation framework (basis of the IPsec stack).

*UMIP (Linux Mobile IPv6 Daemon): the mechanism is fully supported for years. Details and documentation are available at http://umip.org. Linux native interface (Netlink) is used by UMIP to pass MIGRATE message to the kernel which passes it after processing to registered (PF_KEY and Netlink/XFRM) key managers.

*Racoon IKEv1 daemon: the mechanism is fully supported and available upstream since 2008. Racoon relies on PF_KEY for communications with the kernel IPsec stack.

*StrongSwan IKEv2 daemon for Linux: the mechanism is fully supported upstream since version 4.2.9, released in November 2008. Support has been developed by StrongSwan's main developers (Martin Willi and Andreas Steffen) based on this specification.

StrongSwan IKEv2 daemon uses Netlink for communications with the kernel.

---

8. Security Considerations

TOC

There is no specific security considerations for the mechanisms introduced by the document but as it makes 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.

---

9. IANA Considerations

TOC

This document has no actions for IANA.

---

10. Conclusion

TOC

*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 endpoint addresses IPsec SA from one to another.

  -to make the source address to be used by the key manager for SA negotiation available before it is needed.

  -to update addresses of IKE_SA after movement.

*An additional requirement associated with the solution for IKE is the addition in SPD image of additional per-SP hints to be used as addresses for negotiation of SA.

*Currently, large portion of the proposed mechanism is implementation dependent due to lack of standard interface to access the SPD (PF_POLICY?).

---

11. References

TOC

11.1. Normative References

TOC

[RFC2119]	Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels," RFC 2119, March 1997 (TXT).
[RFC2367]	McDonald, D., Metz, C., and B. Phan, "PF_KEY Key Management API, Version 2," RFC 2367, July 1998 (TXT).
[RFC2401]	Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, November 1998 (TXT).
[RFC2409]	Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)," RFC 2409, November 1998 (TXT).
[RFC3775]	Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6," RFC 3775, June 2004 (TXT).
[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 (TXT).
[RFC4301]	Kent, S. and K. Seo, "Security Architecture for the Internet Protocol," RFC 4301, December 2005 (TXT).
[RFC4877]	Devarapalli, V. and F. Dupont, "Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture," RFC 4877, April 2007 (TXT).
[RFC5996]	Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol version 2 (IKEv2)," RFC 5996, September 2010 (TXT).

11.2. Informative References

TOC

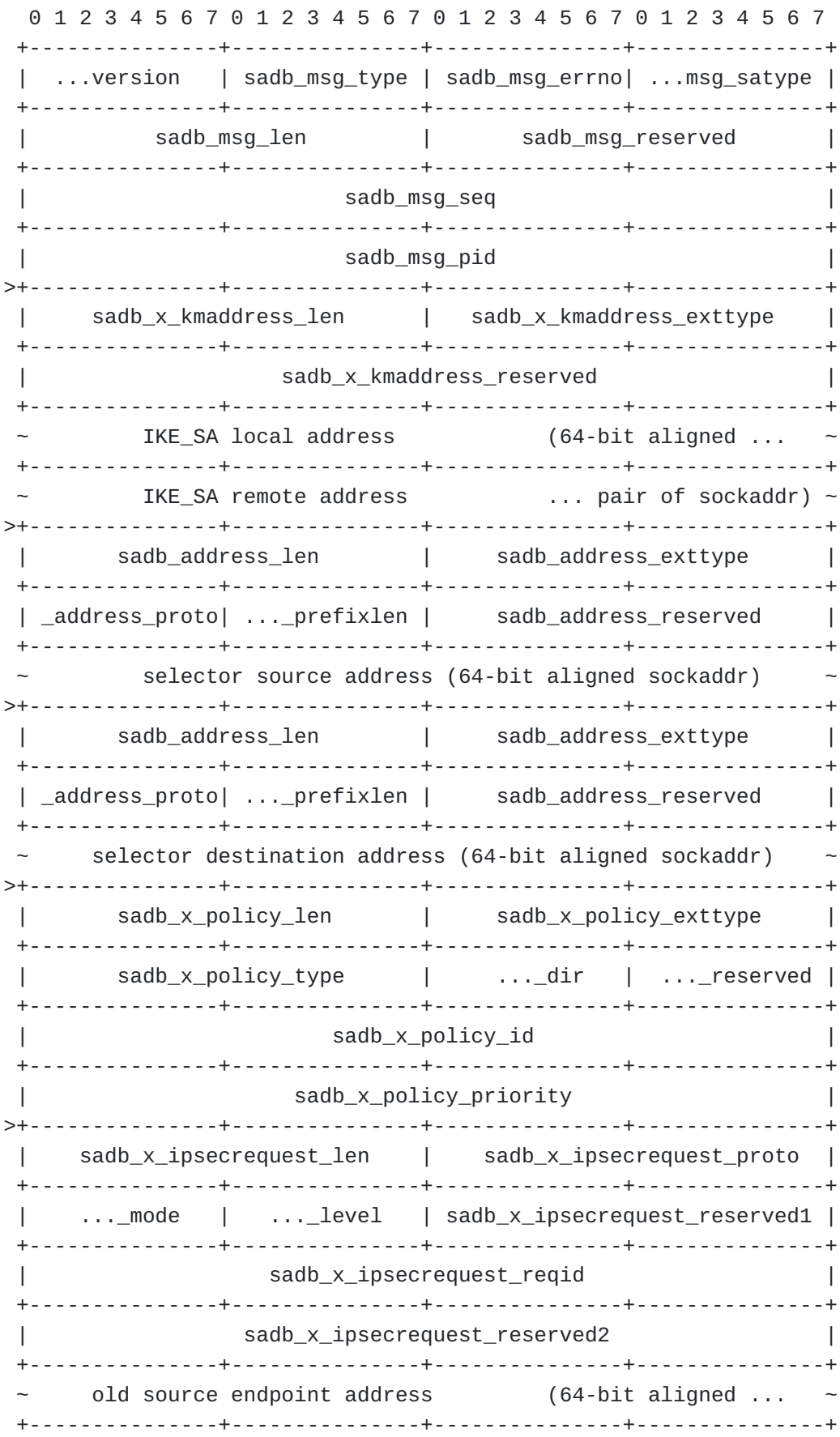
[DSMIPv6]	Soliman, H., "Mobile IPv6 support for dual stack Hosts and Routers," RFC 5555, June 2009 (TXT).
[MIGRATE]	Sugimoto, S., Nakamura, M., and F. Dupont, "PF_KEY Extension as an Interface between Mobile IPv6 and IPsec/IKE," draft-sugimoto-mip6-pfkey-migrate-04 (work in progress), December 2007 (TXT).
[RFC3549]	Salim, J., Khosravi, H., Kleen, A., and A. Kuznetsov, "Linux Netlink as an IP Services Protocol," RFC 3549, July 2003 (TXT).
[RFC3948]	Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets," RFC 3948, January 2005 (TXT).
[RFC3963]	Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," RFC 3963, January 2005 (TXT).

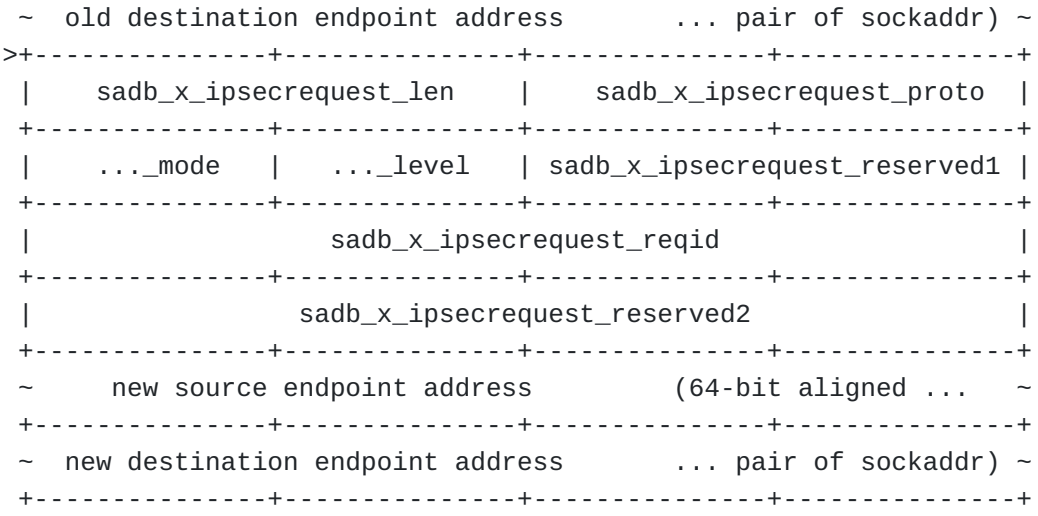
[hld-sec]	Ebalard, A., “Mobile IPv6 Home Link Detection Mechanism Security considerations,” draft-ebalard-mext-hld-security-00 (work in progress), April 2009 (TXT).

Appendix A. PF_KEY MIGRATE Message Format

TOC

The figure below shows the message format of PF_KEY MIGRATE. The message consists of 7 parts (boundary of each part is marked with ">"). The message starts with PF_KEY base message header directly followed by a sadb_x_kmaddress{} structure. The extension holds the two IKE_SA local and remote addresses as opaque data for the key manager (two 64-bit aligned sockaddr). It is then followed by two address extensions: those respectively hold source and destination addresses of the selector. The rest of the message is specific to IPsec implementations on BSD and Linux. 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_x_kmaddress_len
sadb_x_kmaddress_exttype
sadb_x_kmaddress_reserved
IKE_SA local address
(64-bit aligned ...
IKE_SA remote address
... pair of sockaddr)
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 source endpoint address
(64-bit aligned ...



Following is a structure of PF_KEY base message header specified in [RFC2367] (McDonald, D., Metz, C., and B. Phan, "PF_KEY Key Management API, Version 2," July 1998.). 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 the structure of key manager address extension header. SADB_X_EXT_KMADDRESS should be specified in sadb_x_kmaddress_exttype field. It is followed by a pair of sockaddr structures holding respectively up-to-date local and remote address to be used by IKE_SA. The pair is globally 64-bit aligned.

```
struct sadb_x_kmaddress {
        uint16_t        sadb_x_kmaddress_len;
        uint16_t        sadb_x_kmaddress_exttype;
        uint32_t        sadb_x_kmaddress_reserved;
};
/* sizeof(struct sadb_x_kmaddress) == 8 */
/* Followed by two sockaddr (local and remote) */
```

Following is a structure of address extension header specified in [RFC2367] (McDonald, D., Metz, C., and B. Phan, "PF_KEY Key Management API, Version 2," July 1998.). 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 and Linux IPsec implementations. 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 and Linux IPsec implementation. IPsec protocol (ESP or AH) and mode 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

TOC

Various people had contributed and were acknowledged in previous version of [MIGRATE] draft. Because most of the text from previous draft has been kept in this document, those acknowledgements are still

valid: Mitsuru Kanda, Kazunori Miyazawa, Tsuyoshi Momose Shoichi Sakane, Keiichi Shima, Noriaki Takamiya, and Hideaki Yoshifuji.
We would also like to acknowledge here the positive technical feedback from Shinta Sugimoto while extending his MIGRATE mechanism and also the work provided by people of USAGI (Masahide Nakamura, Shinta Sugimoto, Hideaki Yoshifuji, ...) on Linux kernel's Mobile IPv6 and IPsec stack. Additionally, Romain Kuntz and Jean-Michel Combes provided thorough reviews of the document during the publication process.
This document was generated by xml2rfc.

Authors' Addresses

TOC

	Arnaud Ebalard
	EADS Innovation Works
	12, rue Pasteur - BP76
	Suresnes 92152
	France
Email:	arno@natisbad.org
	Sebastien Decugis
	National Institute of Information and Communications Technology
	4-2-1, Nukui-Kitamachi,
	Koganei, Tokyo 184-8795
	Japan
Email:	sdecugis@nict.go.jp