社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2004－MBL－31 (17)
2004－ITS－19 (17)
2004／11／12

Considerations of Interactions between Mobile IPv6 and IPsec

Shinta Sugimoto, Ryoji Kato
Ericsson Research Japan, Nippon Ericsson K.K.
4-1-14, Koraku, Bunkyo-ku, Tokyo, 112-004, Japan
shinta.sugimoto@ericsson.com

Abstract

Mobile IPv6 uses IPsec to protect mobility signals and payload packets exchanged between the MN and HA. Key management by the MN and HA is complicated due to dynamic characteristics of mobile environment. In addition, endpoints of IPsec tunnels established between the MN and HA need to be updated whenever the MN performs movement. Therefore, interaction between Mobile IPv6 and IPsec is necessary. Possible key management scenarios are: (1) Manual Keying, (2) Dynamic Keying without K-bit, and (3) Dynamic Keying with K-bit. Prototype implementation proved that all of these scenarios were feasible and necessary interactions between Mobile IPv6 and IPsec became clear. It was identified that scenario of Dynamic Keying with capability of updating IKE endpoint can be realized only if indication from Mobile IPv6 about MN's binding information and K-bit information is provided.

Keywords: Mobile IPv6, IPsec, and IKE

1. Introduction

As Mobile IPv6 has now become standardized[1], there is a strong expectation for its successful deployment that could lead to the all-IP mobile systems. In order to launch commercial service that is based on Mobile IPv6 it is crucial for the mobile operators to build secure and scalable infrastructure. Therefore, the operator must fully take advantage of security design of Mobile IPv6 and relevant AAA mechanism. In order to achieve this, smooth interaction between Mobile IPv6 and IPsec[2] is necessary. In Mobile IPv6, there are several specific requirements for IPsec, which comes from dynamic characteristics of mobile environment. Therefore, details of the mechanism for Mobile IPv6 and IPsec to interwork should be examined and concerns with the interaction should be cleared.

The objective of this paper is to examine necessary interaction between Mobile IPv6 and IPsec. In order to confirm feasibility prototype implementation of Mobile IPv6 that fully takes advantage of IPsec is presented in this paper.

The rest of the paper is organized as follows. In section 2, overview of security design and mechanism of Mobile IPv6 is presented. In section 3, prototype implementation of fully functional Mobile IPv6 with IPsec is presented focusing on interactions between the two protocols. In section 4, comparative analysis of 3 key management scenarios is given. Possible enhancement and lessons learned from prototype implementation are discussed in Section 5.

2. Overview of Mobile IPv6 Security

Mobile IPv6 uses IPsec to protect signaling messages and payload packets to be exchanged between MN and HA[1][3]. False use of Mobile IPv6 signals may cause serious security problems. If one can send false BU on behalf of the MN to its HA, traffic destined to MN's home address can be hijacked. Therefore, the Mobile IPv6 mandates that the BU and BA exchanged between the MN and HA must be protected by ESP in transport mode. Contents of the BU and BA are encrypted and kept invisible on the fly. Additionally, sequence number included in the BU and BA provides protection against replay attack. It is also important to provide secure mechanism for the corresponding update. Since it is hard to assume trust relationship between the MN and CN, IPsec cannot be used to protect BU and BA of the corresponding binding. Return Routability procedure was designed to realize mutual authentication of MN and CN without security infrastructure[4]. The procedure also assures that the MN is reachable with its care-of address and home address. In Return Routability procedure, the MN and CN perform two tests called Home Test and Care-of Test by exchanging pairs of challenge and

response messages. When the two tests are completed, the MN and CN share a key to calculate authorization data to be included in BU and BA. By securing the path between the MN and HA, it becomes more difficult to eavesdrop the contents of Home Test. Thus, IPsec protection of the Home Test messages (HoTI/HoT) is strongly recommended in the specification.

Table 1 summarizes the usage of IPsec in Mobile IPv6. BU/BA in home registration must be protected by ESP in transport mode with non-null encryption. HoTI/HoT should be protected by ESP in tunnel mode. MPS/MPA which are used for Mobile Prefix Discovery should be protected by ESP in transport mode. MPS and is an ICMPv6 message sent by the MN to HA requesting for information of home prefixes. Mobile IPv6 can provide VPN-like service for the user by taking advantage of IPsec tunnel to protect payload packets. In such case, confidentiality of entire payload traffic is provided on the path between the MN and HA. The HA plays role of security gateway for the MN which is an endpoint. Even though it is not mandated in the specification, the usage would be highly needed in commercial service.

Table 1: Usage of IPsec in Mobile IPv6

Message	Proto	Mode	Support	Use
BU/BA	ESP	Transport	MUST	MUST
HoTI/HoT	ESP	Tunnel	MUST	SHOULD
MPS/MPA	ESP	Transport	MUST	SHOULD
Payload	ESP	Tunnel	MAY	MAY

In order to perform protection for the Mobile IPv6 signals and payload traffic, security policies in along with security associations must be configured for the MN and HA. With respect to the security association database (SAD), totally 4 SA pairs are necessary to be established between the MN and HA in order to fully meet the security requirements. When it comes to the security policy database (SPD), 4 security policy entries for each inbound and outbound SPD are required. It should be noted that a care should be taken for the order of the SP entries in SPD. In addition, Mobile IPv6 requires that some of the SPD entries should be associated with tunnel interface, which is called per-interface SPD entry. More specifically, traffic selector should include condition that the packet is destined to the bi-directional tunnel interface.

2.1. Manual Keying

Manual keying is a scenario where the MN and HA manage the keys statically. In this scenario, operator of the MN and HA should manually configure the security associations to protect Mobile IPv6 signals and payload traffic. Whenever the home address of the MN or HA's address is changed, the operator is required to reconfigure the security associations with manual operation. Another important remark in manual keying scenario is that MN and HA are required to update endpoint address of specific SA pairs in some cases. Such update is necessary when Security Policy configured at the MN and HA mandates that Return Routability signals and/or payload packets must be protected by IPsec (ESP tunnel mode).

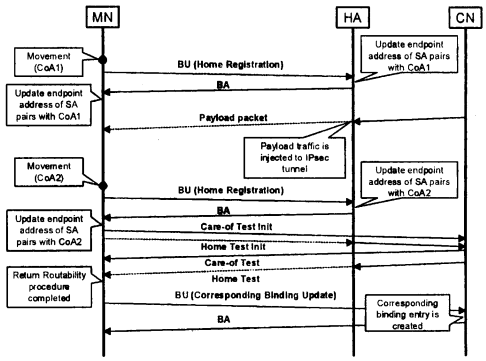

Figure 1: Manual Keying

Figure 1 shows sequence of the MN which uses manual keying performs movement from one network to another. In this scenario, the MN attempts to update corresponding binding stored at the CN after the second movement. As described in the figure, both MN and HA updates endpoint address of specific SA pair when primary CoA of the MN is changed. Once the SA pairs for protecting Return Routability signals and payload packets are successfully updated, IPsec tunnel between the HA and MN become effective. Note that the outer address of the IPsec tunnel should be MN's CoA and HA's address. Packets that are protected by IPsec tunnel mode are depicted with dashed line in the figure. Accordingly, Return Routability procedure takes place and corresponding binding update will be performed. When it comes to the signaling messages exchanged between the MN and HA, there is no additional control packets introduced.

2.2. Dynamic Keying

It is also possible for the MN and HA to run dynamic key management protocol to exchange keys. In dynamic keying scenario, security associations are

automatically managed by Internet Key Exchange (IKE) protocol[5].

From Mobile IPv6 perspective, there are two scenarios to run dynamic keying depending on capability of MN and HA to dynamically update IKE endpoint. Note that IKE endpoint means logical connection of phase 1 negotiation. The capability is represented by K-bit flag set in BU/BA message. K-bit flag is set when the node has capability of moving IKE endpoint from one to another.

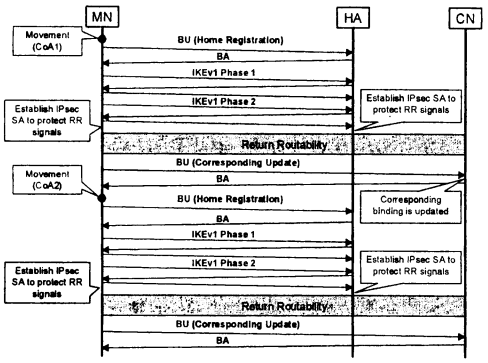

Figure 2: Dynamic Keying K-bit=0

2.2.1. Dynamic Keying without K-bit. The first scenario of dynamic keying can be realized by normal use of IKE. Figure 2 shows sequence of the MN which uses dynamic keying performs movements. When the primary CoA of the MN is changed, the MN initiates IKE phase 1 negotiation and tries to establish ISAKMP SA for protecting further phase 2 messages to be exchanged. Phase 1 negotiation should be followed by phase 2 negotiation. Note that the number of SA pairs depends on the security policy settings. For example, if the MN and HA require that Return Routability messages and payload packet be protected by IPsec tunnel, phase 2 negotiation must be run twice.

2.2.2. Dynamic Keying with K-bit. If both the MN and HA have capability to update its IKE endpoint, K-bit flag in BU and BA is set when the MN performs home registration. In this scenario, the MN should update its local IKE endpoint with its new CoA. Meanwhile, the HA should update remote address of the phase 1 connection when it receives BU indicating that MN's primary CoA is changed.

Once the home registration is done, the MN and HA should first check if there is existing phase 1 connection for the IPsec SA needs to be established. If there is no existing phase 1 connection, new connection should be established. If there has already been a phase 1 connection established between the MN, remote address of the connection should be updated with newly registered CoA. Accordingly, phase 2 negotiation will run being protected by the existing ISAKMP SA. Note that number of phase 2 negotiations is dependent on the security policy settings. After necessary IPsec SA is established, Return Routability procedure is ready to be run.

3. Prototype Implementation

Next, we will see how these key management scenarios actually work in the real implementation. This section gives an introduction to the prototype implementation of fully functional Mobile IPv6. Primary focus is placed on interactions between Mobile IPv6 and IPsec in dynamic keying scenarios.

3.1 Overview

Our original prototype implementation of Mobile IPv6 consists of core and application parts. The core part includes main functionality of Mobile IPv6 including binding cache management, movement detection, handling of Mobile IPv6 specific extension headers etc. The core part is completely integrated into generic IPv6 stack of NetBSD inside the kernel. The application part is implemented in userland as a daemon program (mip6d). The daemon program is responsible for Dynamic Home Agent Address Discovery, Mobile Prefix Discovery, and Security Policy management. All the system is based on NetBSD 1.6.2 Release[6]. With regard to the IPsec components, we used software developed by the KAME Project[7]. IPsec functionality inside the kernel is available by default in NetBSD. As for the key management daemon, we used an user application called `racoon(8)`. The daemon program runs IKE protocol to establish security associations dynamically. In order to make it work on top of Mobile IPv6, we have made several modifications to the IKE daemon. Those modifications are described later in this section.

3.1 Static Keying

In our case, both Mobile IPv6 and IPsec core are implemented inside the kernel. In addition, the entire IPv6 protocol stack including IPsec is open source, we could easily modify Mobile IPv6 and IPsec to interact each other. In order to realize static keying scenario, Mobile IPv6 should have a capability to access IPsec,

especially the SAD. Update should be made in a way that endpoint addresses of SA pairs for protecting Return Routability messages and payload packets must be updated with MN's new CoA. In order to realize this a routine named `mip6_update_sa()` was newly introduced, which allows the Mobile IPv6 to directly access SAD. The routine takes arguments that are new CoA and identifier of specific SA entry that needs to be updated. It should be noted that such direct access is only possible when the IPsec protocol stack is not a black box.

3.2 Dynamic Keying

Next, a detailed mechanism for realizing dynamic keying scenario with/without capability of updating IKE endpoint is presented. Interactions between Mobile IPv6 and IPsec especially the key management daemon are focused.

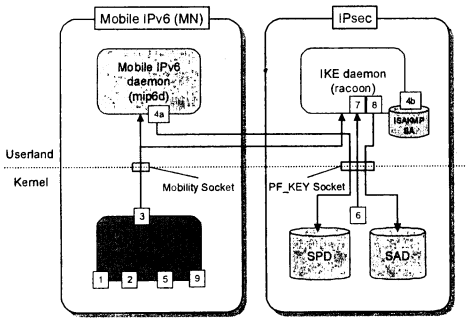

Figure 3: Mobile IPv6-IPsec Interaction on MN

Figure 3 is a block diagram which illustrates how the Mobile IPv6 and IPsec components interwork on MN which runs dynamic keying with/without K-bit support. Since the vertical separation of the components is relatively implementation specific, horizontal interactions rather than vertical interactions have more significance in this figure. Arrows depicted in the figure shows certain kind of notification. Small rectangle represents an action, which should be taken in order of number marked. Followings are the sequence of MN performing movements in Dynamic Keying scenarios. Note that each numbered item corresponds to each rectangle depicted in the figure above:

1) Mobile IPv6 core detects movement and performs home registration
2) Mobile IPv6 core receives BA with successful status code
3) Mobile IPv6 core announce movement in along with K-bit information.
4) Once the movement is announced, following actions should be taken simultaneously:
   a) Mobile IPv6 updates Security Policy.
   b) If K-bit=1, IKE daemon updates local address of associated phase 1 connection.
5) Mobile IPv6 initiates Return Routability procedure (HoTI is sent by the MN).
6) SADB_ACQUIRE message sent to PF_KEY socket indicating that IPsec SA needs to be negotiated.
7) If K-bit=0, phase 1 negotiation is first initiated. Accordingly, phase 2 negotiation takes place.
8) After the IPsec SA is established, SADB_ADD message is sent by the IKE daemon to the PF_KEY socket. Accordingly, SAD is updated and IPsec tunnel is established.
9) IPsec protected HoTI is sent by the MN

Similarly, Figure 4 illustrates how components of the two protocols interwork on HA. Since all of software components are commonly used for HA and MN, overall structure of the HA seems quite similar to that of MN.

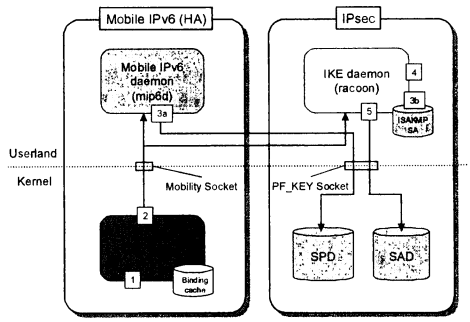

Figure 4: Mobile IPv6-IPsec Interaction on HA

Following actions must be taken when the HA receives home registration BU indicating that the MN's primary CoA is changed:

1) Mobile IPv6 core receives home registration BU from the MN.
2) Mobile IPv6 core announces movement in along with K-bit information.

3) Once the movement is announced, following actions should be taken simultaneously:
   a) Mobile IPv6 updates Security Policy.
   b) If K-bit=1, IKE daemon updates remote address of associated phase 1 connection.
4) IKE daemon responds to the IKE negotiation initiated by the MN.
5) When IPsec SA is negotiated, SADB_ADD is sent to PF_KEY socket. Accordingly, SAD is updated and IPsec tunnel is established.

After all the above procedures are completed, the HA should be able to forward Home Test messages (HoTI/HoT) via IPsec tunnel between the MN.

3.2. Method to Announce Movement

In order to realize K-bit support, Mobile IPv6 should be able to notify key management daemon of the incident in which primary CoA of the MN is changed. It should be noted that the update is necessary for both MN and HA. In our implementation, we have chosen routing socket[9] as a method to pass message from kernel to userland. Mobile IPv6 specific routing message called RTM_MOVEMENT has been newly defined to realize the notification. When the message is issued inside the kernel, required information of the MN's movement are set in specific data structure called `rt_movement_msghdr{}`. Format of the message is presented in Figure 5.

```
struct rt_movement_msghdr {
        u_short rtmv_msglen;
        u_char rtmv_version;
        u_char rtmv_type;
        u_int  rtmv_key_mgmt;          /* K-bit */
        struct in6_addr rtmv_hoa;      /* HoA */
        struct in6_addr rtmv_coa_new;  /* new CoA */
        struct in6_addr rtmv_coa_old;  /* old CoA */
        struct in6_addr rtmv_haa;      /* HA address */
};
```

Figure 5: Message for Movement Announcement

The message includes {home address, new care-of address, and old care-of address} of the MN and HA's address. If the Mobile IPv6 core initially sends the message and there is no old CoA available, the address should be unspecified. At the MN, the RTM_MOVEMENT message must be advertised to the routing socket when its primary CoA is changed. However it does not necessarily mean that message is advertised whenever the MN performs home registration. For instance, the message should not be announced when the MN makes re-registration. It is remarkable that the message can be used by the HA as well. When the HA receives BU from the MN and is noticed that MN's primary CoA is changed, Mobile IPv6 core should advertise the message to the routing socket.

3.3. Modifications to IKE Daemon

First modification is required for the MN regarding the source address selection when running IKE. As mentioned in the specification[3], IKE daemon must use CoA rather than home address to run IKE. In order to make sure that CoA is selected as a source address during the IKE negotiations, additional routine called `getlocalcoa()` was newly introduced.

The next modification is necessary for both MN and HA to dynamically update IKE endpoints, which makes the K-bit support possible. In our implementation, both the MN and HA are triggered by receiving RTM_MOVEMENT message to update IKE endpoint. In MN, the IKE daemon is supposed to hear the message when the MN itself performs movement. In HA, the IKE daemon will hear the message when the HA receives home registration BU and determines that primary CoA for the MN is changed. Two functions, `updateph1_remote()` and `updateph1_local()` are newly introduced to update IKE endpoint based on the information included in RTM_MOVEMENT message. IKE endpoint is an information stored in data structure which holds soft state of phase 1 negotiation. `updateph1_remote()` is used by the HA to update remote address. Meanwhile, `updateph1_local()` is called by the MN to update local address. Old CoA (`rtmv_coa_old`) and HA address (`rtmv_haa`) included in the RTM_MOVEMENT message is used to locate specific entry of phase 1 negotiation to be updated by the routines. Old CoA should be the only clue for the HA to identify specific phase 1 entry for the MN. Thus, not only new CoA but also old CoA is essential information for the IKE daemon to realize K-bit support.

4. Comparative Analysis

Prototype implementation of Mobile IPv6 proved that all of key management scenarios were feasible. Next, we will compare the three scenarios from various perspectives: interaction between Mobile IPv6 and IPsec, security, and traffic cost.

4.1. Interactions between Mobile IPv6 and IPsec

Each key management scenario gives different requirements for the interactions between Mobile IPv6 and IPsec. Since the two protocols are originally independent each other, it is necessary to figure out exact interactions needed. I may be possible that Mobile IPv6 and IPsec are developed by different vendor and combined together. Thus, necessary interactions between the two protocols must be clearly identified.

In Manual Keying scenario, Mobile IPv6 requires that endpoint addresses of specific SA pairs be updated when the MN changes its primary CoA. However, it cannot be realized in a straightforward manner due to the rules specified in PF_KEY Key Management API specification[8]. It is not allowed to update endpoint address of the specific SA entry by the SADB_UPDATE message. Such request will end up in receiving EINVAL error message from the kernel in normal IPsec implementation. Therefore, Mobile IPv6 should issue a request for recreating SA pairs that has newly acquired CoA as a tunnel endpoint address. Both the MN and HA should trigger the update when the primary CoA of the MN is changed. It is important to note that there should be a consistency of SPI assigned for newly created SA pairs on the MN and HA. In order to assure this, the MN and HA should have an agreement on SPI in advance. It may be possible for the MN and HA to agree on use of specific APIs one after the other.

In Dynamic Keying scenarios, there is a requirement that should be satisfied regardless of K-bit capability. Mobile IPv6 should proactively delete the old SA pairs so that IKE negotiation can be successfully invoked. In other words, Mobile IPv6 (both MN and HA) should explicitly requests IPsec for deletion of old SA pairs which is no more in use prior to initiation of the new phase 1 or phase 2 negotiations. Issuing the request should be easily done by PF_KEY socket API without any problem.

In Dynamic Keying K-bit=1 scenario, key management daemon is required to update endpoint address of specific phase 1 connection. In order to realize this, key management daemon should be informed of local and remote address to identify the phase 1 connection to be updated. Besides, an address with which remote/local address is updated is also mandatory information. Additionally, Boolean parameter of K-bit is also required. The MN may dynamically change its HA and capability of updating IKE endpoint may be changed. In such case, key management daemon needs the information whether if it should dynamically update IKE endpoint or not.

4.2. Security

Dynamic keying can provide better security than manual keying, since protection against replay attack is not provided when manual keying is used. Although sequence number included in BU/BA can realize anti-replay attack protection, it is not applicable when the HA lost its state.

Comparing the two scenarios that use dynamic keying, there seems no threat newly introduced by the capability of updating IKE endpoint. From HA perspective, update of IKE endpoint is only performed when the BU is confirmed to be valid. On the MN, movement is announced only when the home registration is successfully completed. Hence, there is little chance to make false update.

4.3. Signaling Cost

Next, we compare signaling cost of each scenario in respect of signaling volume and number of messages to be exchanged on the path between the MN and HA.

Volume of signaling messages may have serious impact on scalability of the HA. As the number of mobile nodes increases total volume of signaling messages that HA should handle will become huge. Therefore, it is important to figure out precise traffic volume of signaling messages required in each key management scenario. Figure 6 shows the comparison of signaling volume in each case. Each bar represents the total volume of signaling messages in units of byte. The results were measured by capturing tool that monitors the network interface of the MN. Targets of packet capturing are BU/BA and IKE signaling messages, which are exchanged between the MN and HA. It is assumed that BU/BA is protected by ESP transport mode and pre-shared secret is used for authentication method in phase 1 negotiation. Note that MN plays role of initiator in every IKE negotiations and aggressive mode is used in phase 1 negotiation. Another important assumption is that security policy requires that only Return Routability messages are protected by IPsec, which means that 1 SA pair should be created upon MN's movement.

Needless to say, Manual Keying is the best in performance since it does not require any IKE signaling. In Dynamic Keying K-bit=0 scenario, total volume of signaling traffic is about 1,800 bytes which is nearly 2 times of the case where K-bit is on. It should be noted that overhead of phase 1 negotiation

depends on authentication mechanism and mode of operation. The graph shows that overhead of IKE negotiations in terms of signaling volume is considerably large when compared to that of Mobile IPv6 generic signals.

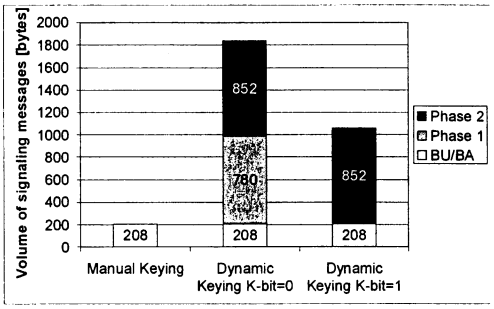

Figure 6: Comparison of volume of signaling messages per movement

Next, we will focus on the number of signaling messages which may put impact on delay. In a case where the CN already has binding of the MN prior to its movement, there will be a disruption of connectivity. More specifically, IP reachability to the MN's home address will be lost until the corresponding registration is successfully updated by the CN. Therefore, it is important to figure out the number of signaling messages needs to be exchanged between the MN and HA before the corresponding binding is updated. In case of manual keying, the MN should wait for at least 1 round-trip between the MN and HA for the Home Test to be completed. In case of Dynamic Keying K-bit=0 scenario, the MN should wait for at least 4 round-trip to receive HoT message sent back from the CN. In a case where K-bit is on, the MN will save 1.5 round-trip required for the phase 1 negotiation. From these observations, use of dynamic keying seems to be very expensive in terms of network delay. For instance, if the round-trip time between the MN and HA is 100ms, the Dynamic Keying without K-bit support should take at least 400ms to update the corresponding binding.

4.4. Summary

To outline comparisons given in the section, Table 2 summarizes the characteristics of 3 scenarios of key management in Mobile IPv6. From operational expense and security perspective, manual keying does not seem to be a recommended solution for the commercial service of Mobile IPv6. Comparing the two scenarios that facilitate dynamic keying, it's a tradeoff of complexity and performance. Dynamic key management may put serious impact on performance of the corresponding update without support of K-bit. The MN should wait for a number of round-trip delays until the Home Test is done. K-bit capability can greatly help MN save round-trip delay which is required for IKE phase 1 negotiation. However, in order to realize K-bit support, tight interaction between Mobile IPv6 and IPsec is necessary.

Table 2: Comparison of key management scenarios

Scenario	MIPv6 - IPsec Interaction	Security	Signaling Cost
Manual Keying	Normal	**Low**	Low
Dynamic Keying K-bit=0	Loose	High	**High**
Dynamic Keying K-bit=1	**Tight**	High	Normal

5. Discussions

In this section, we discuss technical issues that are identified throughout the prototyping activities, which might be commonly raised in other systems. In addition, possible enhancements are also discussed, which could achieve better performance.

5.1. API for Movement Announcement

In order to realize K-bit support, Mobile IPv6 and IPsec should interwork. We have reached to a conclusion that notification should be sent from Mobile IPv6 to IPsec. To assure interoperability between Mobile IPv6 and IPsec protocol stack, standard API for the interaction seems to be needed. Such information may also be useful for user applications other than key management daemon.

It should be also noted that Mobile IPv6 extension to advanced socket API is now being prepared to become standard[10]. The extension makes it possible for application programmer to access Mobile IPv6 specific extension headers and mobility signals by giving special option for socket interface. Therefore, it may be technically possible for user application including key management daemon to monitor status of Mobile IPv6 in a stateful manner. However, it seems that the solution is too expensive for the key management daemon to make such extension and specific API for movement announcement seems more suitable.

5.2. Trigger for Return Routability procedure

Home Test would probably be the most expensive process to be completed before the MN gets ready for corresponding update. Since Home Test must wait for completion of IKE negotiation, it is desirable that transmission of HoTI message is synchronized with the creation of SA pair. Otherwise, transmission of the message is only taken cared by retransmission engine of Mobile IPv6, which might end up in generating huge delay for the completion of Home Test. In order to optimize the performance, Mobile IPv6 should be triggered by IPsec or proactively monitor the status of specific SA pairs.

5.3. Security Policy Management

Throughout the prototyping activity, we have also identified that there are issues around management of Security Policy in Mobile IPv6 operation. Security Policy is essentially defined by selector which is a set of source address, destination address, protocol, and the destination port of the traffic. As described in the specification[3], Mobile IPv6 requires that per-interface SPD entry specially be configured for protection of Return Routability messages and payload packets. This requirement raises a question whether if such policy description is widely feasible or not. For instance, we have identified that per-interface SPD cannot be specified in IPsec architecture on BSD. One of the solutions for this problem is to dynamically update Security Policy settings. Whenever the MN's primary CoA is changed, security policy for protecting Return Routability messages and payload packet can be updated so that the policy entry is associated with the updated/created SA entry. Besides, dynamic update of Security Policy entry may also be necessary in advanced scenario where the MN's home address and HA's address can be dynamically allocated or updated. In such environment, Mobile IPv6 requires that Security Policy settings be dynamically updated.

Nevertheless, it should be noted there has been no standardized API to manage Security Policy Database. In IP Security Policy (IPSP) WG of IETF, there have been discussions on a need to specify PF_POLICY so that the SPD can be accessed via common interface. Such API should be quite useful for Mobile IPv6.

6. Conclusions

Comprehensive scenarios of key management in Mobile IPv6 were compared and analyzed. Prototype implementation of fully functional Mobile IPv6 has shown feasibility of all the scenarios.

Throughout the prototyping activity, necessary interactions between Mobile IPv6 and IPsec are examined in detail. According to the comparative analysis made, dynamic keying scenario without capability of updating IKE endpoint requires less interaction between the Mobile IPv6 and IPsec. However, there is a serious concern with regard to delay due to a number of round-trip delays required for IKE negotiations. Although, capability of updating IKE endpoint can save a number of round-trip delays by omitting phase 1 negotiation, it requires tight interaction between Mobile IPv6 and IPsec in both MN and HA. The interaction is essentially a notification from Mobile IPv6 to IPsec which includes binding information of the MN in along with capability of updating IKE endpoint. Prototype of the notification message was shown and confirmed that it can commonly be used by both MN and HA.

7. Acknowledgements

Authors are grateful to members of USAGI Project for giving us valuable comments and participating in the technical discussions.

8. References

[1] D. Johnson, C. Perkins, J. Arkko, "Mobility Support for IPv6", RFC 3775, June 2004.

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

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

[4] P. Nikander, J. Arkko, T. Aura, G. Montenegro, E. Nordmark, "Mobile IP version 6 Route Optimization Security Design Background", draft-ietf-mip6-ro-sec-01, July 19, 2004.

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

[6] NetBSD, http://www.netbsd.org/

[7] KAME Project, http://www.kame.net/

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

[9] K. Sklower, "A Tree-based Packet Routing Table for Berkeley UNIX", Proceedings of Winter 1991 USENIX Conference, pp. 93-103, 1991.

[10] S. Chakrabarti, E. Nordmark, "Extension to Sockets API for Mobile IPv6", draft-ietf-mip6-mipext-advapi-03.txt, September, 2004.