This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 workshop proceedings.

A Novel Interfacing Solution to Make IKEv2 Work in MIPv6 Environment

Ke Xu, Minpeng Qi, Haitao Li
Department of Computer Science and Technology
Tsinghua University
Beijing China

Peng Yang
Hitachi (China) R&D Corporation
Beijing China

Hui Deng
China Mobile Corporation
Beijing China

***Abstract*—In IETF, Mobile IPv6 signaling and data must be protected by IP security association, while IKEv2 is recommended to make dynamical IPSec security association maintenance. But when mobile node bootstrapped in foreign net work or handover to a new network, IKEv2 cannot get information about MIPv6 to negotiate right IP security associations, which will incur bootstrap failure and long handover delay. In this paper, a novel solution is proposed to interface IKEv2 and MIPv6 for information sharing and mobility detection. In according with our evaluation result, this solution could solve bootstrap problem and reduce the handover delay greatly.**

Keywords- Mobile IPv6; IKEv2; IPSec; PF_KEY; Fast handover

I. INTRODUCTION

Mobile IPv6 (MIPv6) is the standard protocol used to provide mobility support in IPv6 which is designed by Internet Engineering Task Force (IETF) [1]. The protocol defined three kinds of communication nodes: Mobile Node (MN) as the mobile communication entity, Home Agent (HA) as the intelligent router to manage each MN and to forward its packets, Correspondent Node (CN) as the peer node which is communicating with MN. As MIPv6 protocol designed, each MN has an identifier named Home Address (HoA), which can be used for as constant IP address at the layer that is upper than IP layer. When MN roams from its home, it will get another temporary IP address for real communication as Care of Address (CoA). MIPv6 relies on IPSec [2] to protect its management message and the user data message [3].

IPSec is the security protocol at IP layer. A Security Association (SA) is a simplex "connection" that affords security services to the traffic carried by it [2] that stored in Security Association Database (SAD). Internet Key Exchange version 2 (IKEv2) protocol is a component of IPSec used for performing mutual authentication and establishing and maintaining SA [4] according to the policy in Security Policy Database (SPD). The relative SA in IKE should be created. The protocol uses Security Parameter Index (SPI), source address and destination address to get the proper SA.

Reference [3] shows how to negotiate dynamic security association by IKEv2. It defined the pair of SAs protecting registration signaling using addresses of HoA and HA's address. So SA can survive the switch of CoA. It also indicates that SA should be updated after MN's movement, or the SA should be made. But there are two problems to fulfill its need. The first problem is at the stage of MN's bootstrapping, that MN is configured with static Home Address instead of dynamic configuration. It is defined that the pair of address in IPSec SAs should be HoA and HA. But at that time, IKE can only negotiate between addresses of CoA and HA. The other problem is that no perfect solution is made for updating tunnel mode SA after MN's handover. So new SA should be initiated, and it will cause long delay of handover.

There is a proposal being made by Shinta Sugimoto, et al, that proposed a simple PF_KEY [5] extension to make IKEv2 and MIPv6 work together [6]. In their proposal, an extension named PF_KEY MIGRATE was used as interface between MIPv6 and IKEv2, while a series of signaling were designed to update SPD and SADB when MN's handover was made. In their proposal, the MIGRATE message contains information of both old SA and new SA, and was used for notifying MN's movement from MIPv6 to kernel and IKEv2 daemon. The security policy and association should be updated by kernel, then IKEv2 daemon copy SPD and SADB from kernel to update its image. In my opinion, some extra consideration should be made. On the one hand, the proposal makes no contribution to bootstrapping problems mentioned above. One the other hand, MIPv6 daemon, as an application consumer of SAs, should not care for the details of SAs. It would have little knowledge of SA. Moreover, IKE is used for negotiating and managing IPSec SAs. It should have the latest IPSec SA and make a copy into kernel instead of copying an image from kernel.

MOBIKE [7] [8] made another proposal for integration between MIPv6 and IKEv2. It is based on the assumption that both of peers knows IP addresses they would use. It is difficult to fulfill this demand in real world. It also makes no consideration of how to fetch CoA at the beginning of bootstrapping.

In our design mentioned later, we made a data sharing mechanism. What MIPv6 daemon should do is to notify MN's movement and update the database. At the other side, IKEv2 daemon can update and create IPSec SA according to the message from database. Each module can focus on the key points they should do.

In this article, the problems are presented in detail at section 2. The design and analysis of modification is discussed in section 3. The potential attack is discussed in section 4.

II. ANALYSIS OF PROBLEMS

In this section, we discuss the problems that mentioned above in detail. We focus on when and why the problems will happen and the result caused by the problems.

Before we start to analyze the problems, there is a basic requirement of system to make MIPv6 and IKEv2 interwork. It is necessary of updating SPD when MN moves from one network to another in both MN and HA peers. Policies in SPD determine whether the packet should be protected and way of protection. When MN roams, it should update related SPD entries to fit for the need with new CoA.

The first problem will occur at the early stage of MN's roaming. In the scenario of MN's traveling, it would send a request to home agent (HA) to register its home address (HoA). At that time, MN can communicate with HA only by care of address (CoA). If HA accepts the request, it would reply a message to MN. We call the process bootstrapping. The request message is binding update (BU), and the reply is binding acknowledgment (BA). There must be a pair of SAs to protect the signaling message of BU and BA. The SAs addresses pair should be MN's HoA and HA's address. But the original IKE can make IPSec SAs by using the same source and destination addresses.

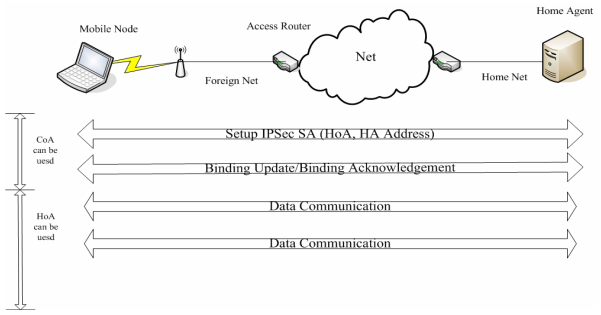

Figure 1. Process of Mobile Node's bootstrapping

As shown in Fig. 1, at the MN's first travel, the creation of this pair SAs should base on CoA to communicate because the HoA is unreachable before registration. The HoA can be used in data communication after the signaling of BU/BA. Because of this problem, IKE cannot sets up these IPSec SAs to protect BU/BA. As a result, the pair of IPSec SAs should be configured manually. It will reduce the security level and cause a series of problems related.

Since MN can work well at foreign net, the second problem will happen when MN is making handover from one foreign net to another.

As shown in Fig. 2, MN has communication with CN in traverse tunnel in the left foreign net at first. The data they communicated and some of control message between MN and HA would be protected by IPSec SAs in tunnel mode. The IPSec SAs should be addressed of CoA and HA's address. Then MN moves from left foreign net to right foreign net. While it's handover, the communication between MN and CN should be maintained. As we can see, in the environment of Mobile IPv6, movement of MN will change its CoA. At that time, some related SA should be updated because of the IP address changes. In the figure, the IPSec SAs with address of CoA1 and HA's address should be updated to IPSec SAs with address of CoA2 and HA's address. But the normal IKE is not able to make it. So new SAs will be created and the old SAs will be died. This process will consume much system resource and time to deal with, and some of the processes even need manual intervention.

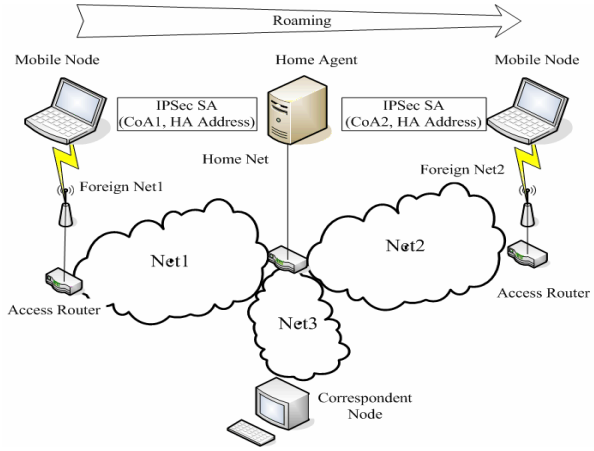

Figure 2. Process of Mobile Node's handover

III. DESIGN AND ANALYSIS

A. Design Goals

The root of two problems described above is that IKEv2 daemon has no ability to detect and use related movement information about MIPv6. Two or more addresses are associated with in MIPv6 module, but IKEv2 doesn't know. IKEv2 take the addresses as separately. If IKEv2 daemon can get enough information about MIPv6, the little modification of IKEv2 can make it update and create SA by avoiding large resource consuming. So the key point is how to make IKEv2 is aware of the information of MIPv6. We designed a data sharing mechanism between MIPv6 and IKEv2. The mechanism should fulfill the goals as following:

- It can take enough information of sharing.
- It can access the information fast and exactly.
- It should assure the information latest and valid.
- It should consider the threat to the information.

After that, IKEv2 is aware of MIPv6, and its daemon should make little modification to update and create the proper SA according to the information.

Now we present the design of the mechanism in detail. The data sharing mechanism is a database named Mobile Security Reference Database (MSRD) [9] and a simple extension of

interface between IKEv2 daemon and IPSec module in kernel: the PF_KEY framework [5].

B. *Mobile Security Reference Database*

MSRD is a database that stores mobile security reference. MIPv6 daemon stores security related information in it. To meet the first need mentioned above, related information in database should group together as following: MN's HoA, CoA, HA's address, binding lifetime and any other information. In peer named MN, it would have only one mobile status of itself. The database contains one entry of the information as a result. In peer named HA, it will store many MNs' status. Each should be stored in database as an entry. Both in MN and in HA, the entry/entries should be indexed and queried by mobile parameter index (MPI) for the request of accessing information fast and exactly. The structure of MSRD is shown as following Fig. 3:

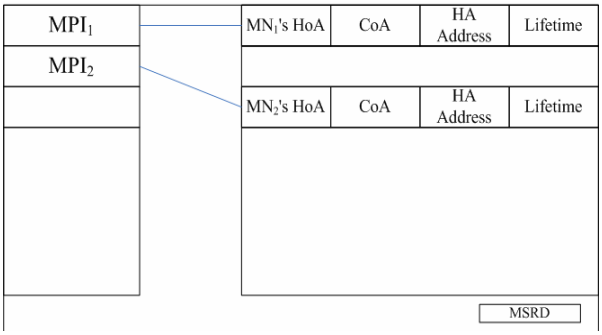

Figure 3. Structure of Mobile Security Reference Database

To make the information latest and valid, it is asked that MIPv6 daemon should update the information as soon as possible when MN's status changes. The proper architecture of database can prevent from the potential attacks. It will discuss in the next section in detail.

C. *Extension of PF_KEY framework*

MSRD is the data sharing basis for MIPv6 and IKEv2 integration, but there must be a path or a pipe to make IKEv2 sensitive about information stored in this database. So we design a set of application program interface (API) in MSRD. IKEv2 daemon can query the database via APIs according to MPI. But there is another problem that where can the IKEv2 daemon get exact MPI. As we know, IKEv2 daemon gets the proper parameter to negotiate IPSec SA from kernel via PF_KEY API normally. So it is acceptable to make a little extension on the PF_KEY framework to transfer MPI. PF_KEY is a new socket protocol family used by trusted privileged key management applications which communicate with an operating system's key management internals [5]. One message named SADB_ACQUIRE is used for creating new IPSec SA. It can be sent to IKEv2 daemon by kernel and application SA consumers. The extension of this message can fit our need and didn't act against its normal behavior because IKEv2 will create new SA without MIPv6 information. The format of this message is as following:

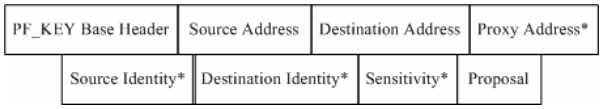

Figure 4. Header format of SADB_ACQUIRE message

As shown in Fig. 4, source address and destination address are addresses with upper layer protocol's port; Proposal payload contains parameter to negotiate SA. Symbol * means the payload is optional.

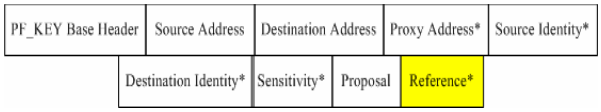

Figure 5. Header format of extension SADB_ACQUIRE message

As shown in Fig. 5, the extension is an optional payload named Reference. MPI can be inserted into this payload. As MN roams, either MIPv6 daemon or kernel module can send SADB_ACQUIRE with this payload to IKEv2 daemon. Then IKEv2 daemon can get information via MSRD's API according to the MPI extracted from reference payload.

D. *Necessary Modification*

To fulfill the demand of our design, there are some modifications in MIPv6, kernel and IKEv2.

First, MSRD is database stores mobility information. It should be a sub-system in MIPv6. It is acceptable that MSRD is create and managed by MIPv6. MIPv6 will have the highest privilege for accessing the database. In addition, MIPv6 daemon should be able to send PF_KEY SADB_ACQUIRE message with reference extension when MN roams.

Second, reference extension should be accepted as an optional payload of SADB_ACQUIRE message. The modification of kernel should fulfill this demand. In addition, Kernel should be able to get exact MPI to organize extension payload independently.

Third, when an SADB_ACQUIRE message with reference extension sends to IKEv2 daemon, it should be able to parse the payload and get MPI. It should get the related secure information by using API of MSRD. Then the daemon should be able to use additional information about MIPv6 in mobile environment to negotiate proper SAs according to the procedure for initiation of bootstrapping mentioned in RFC 4877[3], and to update or create SAs without full negotiation.

E. *Performance Analysis*

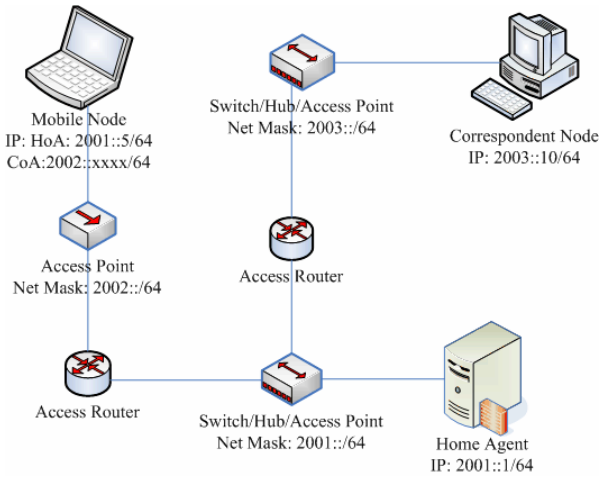

Figure 6. Topology of Test Bed

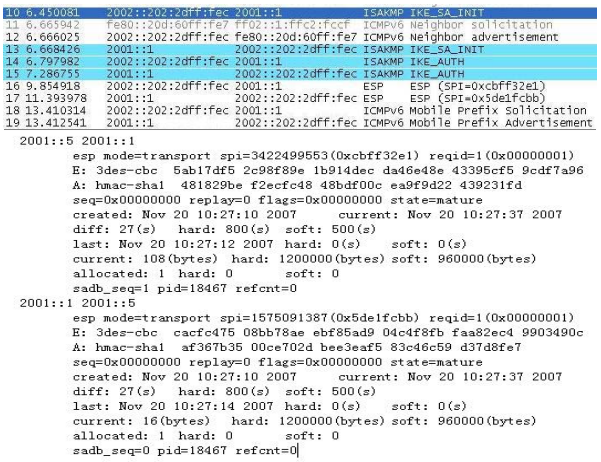

Figure 7. Process of IKEv2 negotiation for Bootstrapping

As shown in Fig. 6, the test bed contains 3 nodes, 2 access routers with three nets. The 3 nodes are represented with MN, HA and CN. The nets are divided into two parts: one for MN's home net and the other for foreign nets. CN is in one of foreign nets. The two access routers are responsible for router advertisements. The system kernel of nodes is Linux v2.6.16 with Mobile IPv6 for Linux (MIPL) v2.0.2 kernel patch. The IKE daemon uses OpenIKEv2 v0.92. The Mobile IPv6 daemon is MIPL v2.0.2. They are all altered with necessary modification to make the system running. After modification implemented, we present that how the new mechanism is used to solve the problems.

At MN's bootstrapping scenario, MN starts at foreign net or moves from home net to foreign net at the first time. As soon as MIPv6 daemon detects the node is out of home net, it can acquire CoA through some auto-configuration mechanism, i.e. router advertisement mechanism. Then MIPv6 daemon writes the binding message into MSRD as well as sending out BU message. The message will trigger SADB_ACQUIRE in kernel. At that time, kernel should be able to get right MPI from MSRD and inserts it into reference payload. After that, the message is sent to IKEv2 daemon. So IKEv2 daemon can get MN's CoA from MSRD like the scenario above and communicate with the other end with CoA. After the connection is set up, IKEv2 daemon can take HoA into Traffic Selector (TS) payload in IKE_AUTH exchange [3]. At the end, two ends can make SAs with MN's HoA. Now the signaling of BU/BA can be protected with dynamic IPSec SAs. The implementation can be seen as shown in Fig.7.

The upper half of Fig. 7 shows the packets analysis captured by Ethereal in Linux. The lower half is the real IPSec SAs used for protection of signaling BU/BA. HoA of MN is 2001::5 with 64 bits net prefix. HA's address is 2001::1. Now the MN is bootstrapped at 2002::/64 foreign net. According to this figure, we can see that IKEv2 daemon in both peer communicating with address CoA and HA's address at first. After the four exchanges are made, a pair of encrypted data is sent. The pair of IPSec SAs is found by their SPIs and made with addresses HoA and HA's address as shown.

In the scenario of a MN's handover from one place to another, its CoA will change from CoA1 to CoA2, and tunnel mode IPSec SAs should be updated with new address as shown in Fig. 2 above. In the system with our proposal fixed, MIPv6 daemon will write the information into MSRD and get the MPI, then it will create SADB_ACQUIRE message with reference payload inserting the MPI and send it to kernel as PF_KEY API defined. Kernel receives this message and checks it, then forwards the message to IKEv2 daemon. As IKEv2 daemon receives this message, it can extract MPI from reference payload. By querying MSRD according MPI, IKEv2 daemon can get enough mobility information to update the related SAs. At the end, IKEv2 daemon can send a notification message to the other end for advanced secure consideration. The following figure will show data transferring information when MN is handover.

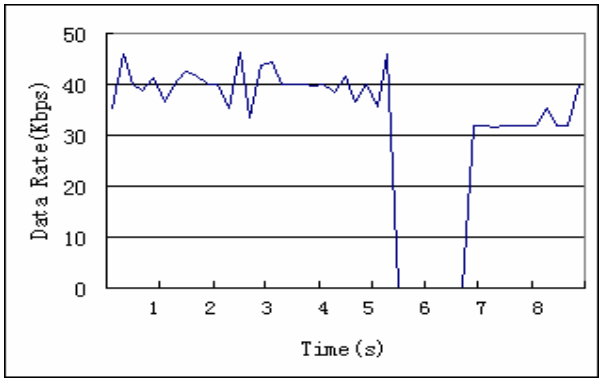

Figure 8. Application Data Rate in MN's Handover

Fig. 8 indicates how the application data rate changes with time when MN's handover. It is clear that data is transferring before MN's movement. When MN does handover, it is required to re-configure care of address and local routing table, update binding with HA, and re-establish IPSec SAs. During the handover, the network side will drop the packets to/from

the MN. As the IPSec SA re-establishment finished, the communication continues again. The handover delay contains time consumed for movement detection, router discovery, handover decision and SAs updating. In Fig.8, the handover delay caused by router discovery is removed to simplify the analysis. In our system, this router discovery procedure may take 6~9 seconds, which highly depends on the network infrastructure. The procedure of Duplicate Address Detection (DAD) is avoided to bypass the effects of long time delay. From Fig.8, it is clear that application communication was recovered quickly after MN's handover and data rate recovered close to it was before MN's handover.

TABLE I. CONTRAST OF TIME SPENDING TO RECOVER COMMUNITATION

Data Transferring Bit Rate (Kbps)	Direct Update with Proposal (s)	Normal Re-generation of IKE to get new pair of IPSec SAs (s)
40	1.62	2.99
100	1.66	2.71
1000	1.65	2.62

The data in Table 1 shows the performance compared of handover delay with same network infrastructure. The first column is application's data rates. It indicates different kinds of rate-limited ftp application scenarios traffic features. The second column shows recovery time of handover that it takes when our solution is applied in the MIPv6 system. The third column shows recovery time of the system without our solution. The handover delay can be divided into two parts: the time for binding registration and the time for IPSec SA re-establishment. Nearly 20%-30% time is used for registration and the test for IKE/IPSec procedure. As the table shows, simply updating IPSec SAs instead of IKEv2 re-negotiation can definitely reduce the handover latency greatly while MN is doing handover.

IV. POTANTIAL ATTACK

In this section, the security of our design will be discussed in detail. As the MSRD is added, it is important to consider the security of data organization and interface. In our design, the data in database is organized as a table, modified only by MIPv6, and queried by address in kernel or by MPI in application except MIPv6. MSRD should have ability to check whether the query is valid. This provides security from attacking to MSRD.

Second, as a data sharing basis, one or more query requests and changing request may be sent to the database at the same time. The reply sequence for these requests is very important. If the query requests are dealt with firstly, the mistake will happen. So in our design, the query request is always sent later than changing request.

Third, the security of PF_KEY extension should be thought about. Someone will be afraid of the payload's misuse. Someone will worry about its malicious usage. Their worrying was included into consideration when we designed this extension. The extension has a small and fixed size to take message: MPI. The extension is an optional payload in the message and can be ignored by applications that don't use it.

At last, in the network environment, there is no additional potential attack to our design because we add no more communication between the two ends except a pair of IKE notification exchange. The security of IKE exchange is secure enough and out of our scope.

V. CONCLUSION

In this paper, A novel proposal is made for interfacing between MIPv6 and IKEv2 to solve the compatibility problem between IKEv2 and Mobile IPv6. The proposal can deal with the problem of MIPv6 bootstrapping with dynamic IPSec SA negotiation as indicated in the paper, and avoid the inefficiency IKEv2 re-negotiation during handover. A data sharing mechanism named MSRD is brought out as a basis of this proposal, and a small extension of PF_KEY framework is made for interworking between MIPv6 and IKEv2. This extension can transfer the index of shared mobility information from MIPv6 to IKEv2. With this data sharing mechanism and simple PF_KEY extension, IKEv2 can set up or update the right pair of IPSec SAs for the need of MIPv6.

ACKNOWLEDGMENT

The authors would like to thank Yuanchen Ma, and Jiping Lv from Hitachi (China) Research and Development Corporation for their help and valuable discussions on this paper. The authors also thank Yan Li, Xiaobo Qu, Yan Zhou and Yan Xu for reviewing and proofreading this paper.

REFERENCES

[1] D. Johnson, C. Perkins and J. Arkko, "Mobility support for IPv6," RFC 3775, Internet Engineering Task Force, June 2004

[2] S. Kent and K. Seo, "Security architecture for the Internet protocol," RFC 4301, Internet Engineering Task Force, December 2005

[3] V. Devarapalli and F. Dupont, "Mobile IPv6 operation with IKEv2 and the revised IPSec architecture," RFC 4877, Internet Engineering Task Force, April 2007

[4] C. Kaufman, "Internet key exchange (IKEv2) protocol," RFC 4306, Internet Engineering Task Force, December 2005

[5] D. McDonald, C. Metz and B. Phan, "PF_KEY key management API, version 2," RFC 2367, Internet Engineering Task Force, July 1998.

[6] S. Sugimoto, F. Dupont and R. Kato, "Interaction between mobile IPv6 and IPSec/IKE," in IPSJ Digital Courier, November 2006.

[7] P. Eronen, "IKEv2 mobility and multihoming protocol (MOBIKE)," RFC 4555, Internet Engineering Task Force, June 2006

[8] T. Kivinen, "Design of the IKEv2 mobility and multihoming (MOBIKE) protocol," RFC 4621, Internet Engineering Task Force, August 2006

[9] M. Qi, H. Li, P. Yang, H. Deng, Y. Ma and K. Xu, "Interfacing between IKEv2/IPsec & MIPv6 by simple PF_KEY extensions," unpublished.