Fast and Simple NEMO Authentication via Random Number

Tat Kin Tan and Azman Samsudin
School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia.
tatkin@cs.usm.my, azman@cs.usm.my

Abstract—**Network Mobility (NEMO), derived from Mobile Internet Protocol version 6 (MIPv6) technology, has been evolving and expanding in very rapidity manner across very short period of time. In MIPv6 technology, the Access Router (AR) plays important roles such as provide sessions connectivity and safe-guarding communications protocol. Whereas in NEMO, the routers in the communication chains are essentially the moving targets and hence the Mobile Router (MR). The MR too, inherits the basic functionalities of AR. In the basic MIPv6 and NEMO modus operandi, when the Mobile Network Notes (MNN) roamed away from home link and attached to some other foreign link, the MNNs are also addressable via a secondary address named as Care-of Address (CoA). The MNN essentially registered its primary CoA to the router on its own home link and thus registering the router to be its Home Agent (HA). These communication sets, and the ingredients of the communication will be managed via Binding Updates (BU) and the Binding Acknowledgement (BAck). Since the NEMO is using MIPv6 as the backbone, NEMO also inevitably inherits many design difficulties and problems that the MIPv6 seen. Such as using the Internet Protocol Security (IPSec) with end-to-end tunneling and using nonce value within the BU to serve as part of the authentication process and many more. In this paper, we illustrate the scenario of authentication problems involving NEMO movements and the flaw in security design. With the aim of succeeding IPSec we then propose a much better solution which is the use of Random Number, coupled with PKI concept.**

Index Terms—**Network Mobility (NEMO), Mobile Network Notes (MNN), Mobile Router (MR), Care-of Address (CoA), Home Agent (HA), Binding Updates (BU), Acknowledgement (BAck).**

I. Introduction

The standard specification of MIPv6 [1] defined the recommended communication protocols and the way how 2 sets of mobile notes shall be exchanging bits and bytes. Whereas the standard documentation of NEMO [2] defined the recommended communication protocols and the way how a MR shall be managed while interacting with MNNs.

Nevertheless in both MIPv6 and NEMO protocol sets, the basic expectation is that the MNN is always expected to be addressable at its home address, regardless of the current location whether at home link or at foreign link, and also being protected and secured via the implementation of IPSec [1], [10], [11]. Such security implementation is also applicable to NEMO's MR. As because the MR is simply another MNN, which is router-capable.

The term "Home Address" is the address which is being allocated to the MNN while it is first booted at its home link and is essentially a subnet IP address from the home link. "Care-of Address", is another IP address being allocated by the router, be it the AR or the MR the MNN attached to while moving away from home into a foreign subnet. Correspondent Node (CN) is any Node, regardless fix of mobile node that is communicating with the MNN.

While moving away from home, regardless of under MIPv6 or NEMO condition, the MNN will have to establish a "Binding" that will associate the HoA and CoA together so that the communications from CN to HoA will be able to reach the new CoA. The control messages that resulting the Binding will be exchanged via the BU-BAck protocol sets. The end result of the Binding will see the Home Agent being formed as the MNN requesting the router at its home link to manage the BU and thereafter communications between CN->HA->MNN(HoA-CoA).

For both MIPv6 and NEMO, the standard specification recommended the usage of IPSec to secure the communications between HA and MNN via the usage of IPSec with IPv6 Encapsulation [1],[2]. The full functionality of IPv6 Encapsulation and IP ESP can be found in [14], [15].

The objective for NEMO is to ensure session continuity for all the mobile nodes (or Mobile Host) in the mobile networks even when the MR has changed its physical attachment point [3], [4]. On top of that, the MR is also responsible in distributing subnet prefixes such as described in [7]. The standard protocol has also pointed out that in order to achieve the session connectivity, one of the basic requirements is to setup bi-directional tunnel between the MR and its HA [2]. In which, a MIPv6 compliance HA can also be a NEMO HA. The bi-directional tunnel is used as a tunnel for MR to inform HA of its attachment via BU.

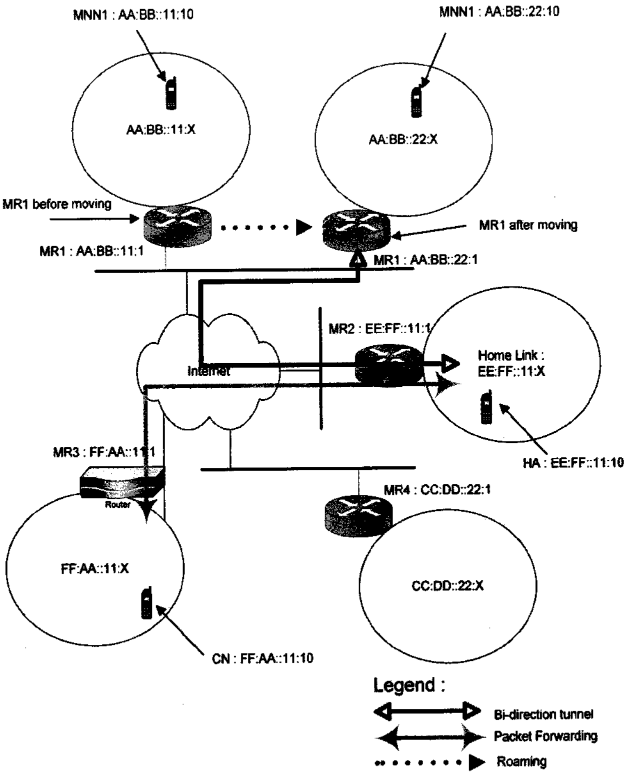

Figure 1 : Basic movement and bi-directional Tunnelling of NEMO MR supports

As shown in Figure 1, an example showed that MR1 moved from one point of attachment to another point of attachment. Let's assume that a new CoA is assigned upon the completion of roaming. MR1 will have to send BU back to it's HA, which this BU contains the CoA being given to the MR1 at new location. The HA upon receiving the BU, will echo back the BAck to MR1 and hence setting up a new the bi-directional tunnel. Until the new tunnel is setup and roaming completed, note that the old tunnel is still a valid tunnel. The MR will also aggregate the new prefixes information within the network to HA.

The HA will then setup and forward all packets destined to the nodes within MR1 via this tunnel after authentication is being done in the BU-BAck exchange [1], [2]. The MR1 will then reverse tunnel and respond to all packets' source addresses belonged to nodes within the mobile network prefixes. The HA, will then de-capsulate and forward packets to the CN. As shown in Figure 1, the scenario described is the simplest NEMO communications setup [2].

When CN sends a packet to a node that is located within the mobile network, this packet will be routed into HA, of which the HA will have the binding of the mobile router and subsequently the aggregated prefix and hence the node. This is the simplest communication model as pointing the CN directly link to same HA. The scenario will be more complicated if assuming CN has some sub layering of nested loop and hence CN need to link with its own HA (and this CN-HA is a different MR-HA) as in Figure 2.

When a HA received a packet that is meant for a node within a mobile network, the HA tunnels the packet to the MR's CoA. The MR will then de-capsulate the tunneled packet and verify that the source address on the outer IPv6 header is the HA's address. It also has to verify that the destination address on the inner IPv6 header belongs to one of its Mobile Network Prefixes before forwarding to the mobile network.

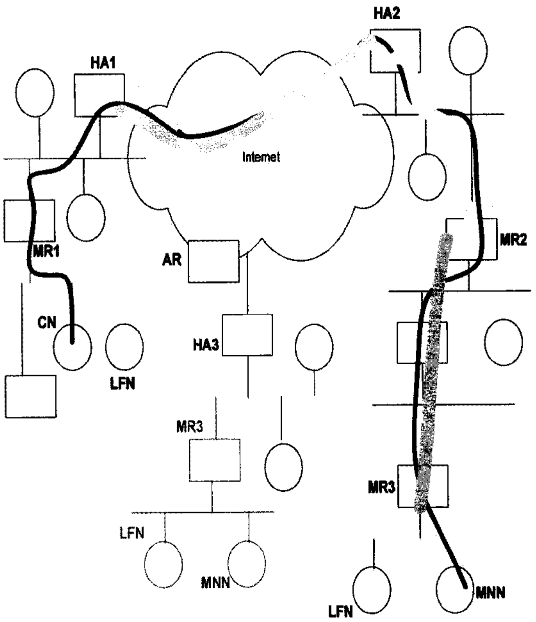

Figure 2 : Basic communication and movement of MR/CN and its operation

A rather more complicated implementation of such kind can be seen in Figure 2. The diagram has shown that how communication can be done to a typical MNN. There are 3 level of protocol encapsulation anyhow, which are HA1->HA2; HA2->MR2 and MR2->MR3. (Assuming that MR can also act as HA in the situation for instance in the event that the HA2 is malfunction.)

Also can be seen in Figure 3, the communication between CN and MNN has to be thru different level of protocol encapsulation after the MR3 has traveled to different link network. Such implementation can best be applied in a moving train, or moving bus in a city, of which the individual bus can have a router installed and hence acted like a mobile router; and the MNN will be imagine as a mobile PDA carried by passengers.

II. Problem Statement

The statement as below:

When CN sends a packet to a node that is located within the mobile network, this packet will be routed into HA, of which the HA will have the binding of the mobile router and subsequently the aggregated prefix and hence the node. <quoted : RFC3775/RFC3963>

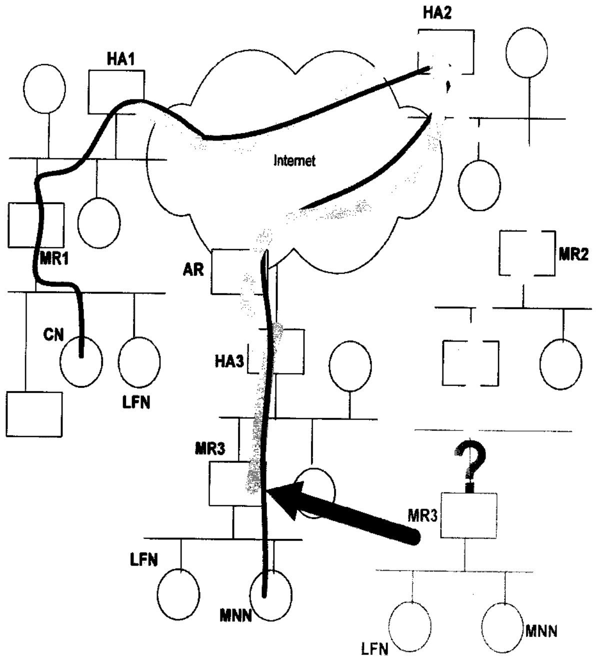

Figure 3: Basic movement of MR and the associated tunnels

When the CN is communicating to MNN/MR/HA, both parties had already agreed upon certain security rules for the encryption/decryption portion. These rules are called Security Association (SA), of which is actually a database that contains array of encryption standards and algorithms such as Advance Encryption Standard (AES), Data Encryption Standards (DES) and/or with various encryption algorithms such as Cipher Block Chaining (CBC), Electronic Coke Book (ECB) and many more variants such as Counter Mode Encryption with CBC-MAC authentication (CCM).

As an example shown in Figure 4, the first MNN may have the capability of supporting up to 6 different type of SA, while as for the second MNN which is less capable, may only support up to 3 sets of SA.

One of the ineffectiveness can be seen during the process where both MNNs will need to sync up which suitable security standards to be employed. MNN1 will typically take its first standard, as in the example shown in Figure 4 as AES, and ask MNN2 if MNN2 has AES too. The answer is obviously MNN2 does not support AES. MNN1 will then move on the second standards which is AES-ECB and the process continue until finally both found a match which is both MNNs supporting AES-CCM standards. This type of scanning and comparing processes will introduce processing overhead.

This is a typical protocol especially for IPSec. The IPSec implementation is not entirely error free and had exposed numerous security threats such as being widely discussed in technical forum, the IETF working groups as well as many research and conference papers publication. [5], [8], [9], [13]

The IPSec is the current proposed security system that is being proposed in NEMO and MIPv6 protocol. BU and BAck can be secured by using any of the security design such as the recommended IPSec or alternative security design system such as PKI [6].

AES	DES
AES-ECB	3DES
AES-CBC	AES-CCM
AES-CCM	
3DES	
DES	

Figure 4 : Sample of SA profiles between 2 MNNs

Now, imagine of what happen if a CN is still communicating to MNN/MR, and because CN does not have the visibility of the location of MNN, CN is still sending packets to MNN's Home Agent (as shown in Figure 3) with MR3's movement is transparent to the CN. What CN really cares is that, CN will send packets to CN's HA1 and it is this HA1 that then establish associations and encapsulate the packets to the MNN's HA2 (and hence reaching to MR2), which is located in another side of the world. The HA2 then relay the packets to MR3 which associated with the MR2. The HA2 itself, contains the binding of MR2-HA2 and then this MR2 decapsulate the packets to route to MR3 which contained the MNN. All communication seemed perfect except that, in Figure 3, the MR3 had just roamed away.

Just imagine, if there is multiple nested loop of mobile networks, meaning for instance another scenario of MR1 attached to another MR2 and MR2 in turns attached to another MR3 and this MR3 in turn attached to another MR4 and so on .... There will be a lot of protocol overheads just to encapsulate/decapsulate, and the each layer of HA-MRs will have verify and authenticate each other. This type of communication model will not be a problem for a 1-layer NEMO communication such as shown in Figure 1. But in real life scenario, the situation as illustrated in Figure 1 does not always happen. There will be a lot of IPSec tunnels that carries BU/BAck exchanges just for each layer and for the BU to provide authentications as scenario of Figure 3 in more reality.

Technically speaking, the industrial specifications (RFC3775 and RFC3963) suggested in using BU and BAck as a form to achieve authentication, whenever a MR/MNN roamed to another subnet and given a new CoA [1], [2]. The BU will contain old CoA and then being mapped by a new CoA, hence informing the other communication node that it has changed the location. After the exchanged of BU-BAck (because BU-BAck contains some parameter that later being used in MIPv6 protocol as a way to authenticate during each roaming), the HA can then authenticate the MR that has new CoA, is indeed the genuine MR before the roaming happened.

Since the BU is sent via an IPSec tunnel, no attacker can hack into the IPSec tunnels and possibly alter the address/content of the packets and hence redirecting the packets [11]. But this is all possible for authentication ONLY after IPSec tunnel has been setup, simply because the BU is hidden and protected behind the IPSec capacity.

As we have just described that under fast or constant roaming situation, the setting up and tearing down of bi-directional tunnel for IPSec to be applicable, thru multiple layers (nested loop), is indeed an implementation too expensive under such rapid changes. For each movement, the node will have to wait multi-layer of communication (or setting up tunnels and exchanging BU-BAck) until reaching authentication part, will be too slow and degrading overall system performance. One may argue that these inefficiencies will be insignificant when coupled with powerful hardware processor, but we can still make a different by exploring an alternative solution which is much cheaper to be implemented.

III. THE SOLUTION

The new idea we propose here to tackle the aforementioned problem (to achieve faster authentication without compromising security design as a whole), is to have each mobile device that comply with NEMO protocol to add a simple authentication mechanism protocol. This authentication mechanism is the implementation of a standalone array of Random Number (RN) in NEMO. The RNs are associated with each Correspondent Nodes that the device has exchanged with. This solution is very simple and efficient to implement.

Technically speaking this implementation is possible consider below situation:

MNN sets up bi-directional tunnels with CN as opposed to Figure2, and enjoying communications with CN. Note that this bi-directional tunnel can still use IPSec as security implementation as IPSec does not conflict with our proposed solution at this stage.

Upon the communication has established, the MR3/MR2 (that the MNN connected to) generates and exchange a truly RN value which identifying this transaction of communication, to the HA2 they associated with. The HA2 stores this standalone array of RN for future reference. The RN can be generated using either hardware or software implementation and the metrics of the MNN will look like in Figure 5. We proposed the metrics shall be kept within the each mobile host.

MR/Mobile Hosts	RN Value	Time (TTL)
MNN1	20B9CAF5ACCA177D	100
MR1	F86D11A49A5C208F	200
MR *n*	8E807ABC1B1A9E6B	300

Figure 5 : Sample metrics of RN on MNN

As seen in Figure 5 and using a 64-bits RN Value, assuming that this metrics is managed by CN, whenever the authentication process is kick-started, the CN will map the RN Value within the metrics that the CN managed and verify against the value given by the MR. The time is the time-to-live (TTL) of which the life time of the random number.

Applying RN in scenario of Figure 3, when MR3 roamed to another region with different subnet, and attached with HA3, the MR3 will just need to notify HA2 via HA3 and provide with the RN. HA2 will then take the RN, map with the metrics database it has, and find a match, and then concluded that the MR3 is a genuine MR3 that he was talking previously, and now at a different location. The entire idea is that for the MNNs which is compliant to NEMO protocol, just to safe-keep RN value and being processes at later stage.

The BU/BAck exchange is still important in the overall NEMO system protocol, but with this idea, the BU can come at the later stage and the nodes do not have to wait until the BU stage to get authenticated. This is also critical in nested layer.

One may have doubts on how can the integrity and the exchange process of the RN be secured without using IPSec? Echoing the concept of introducing PKI for security design [6], the exchange and updates of the RN can be done via PKI mechanism [12] such as shown in Figure 6.

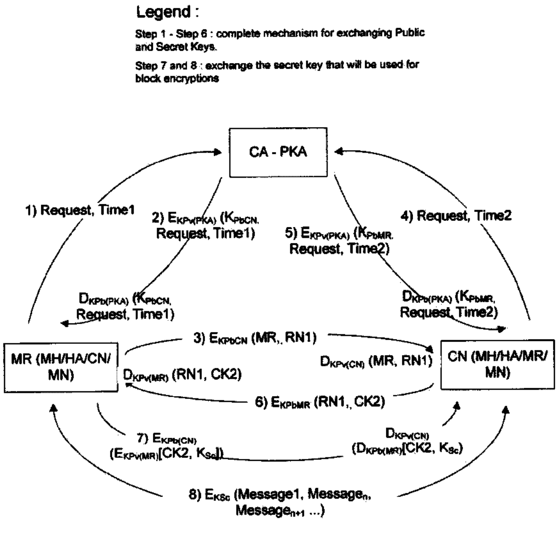
Legend :
Step 1 - Step 6 : complete mechanism for exchanging Public and Secret Keys.
Step 7 and 8 : exchange the secret key that will be used for block encryptions
CA - PKA
1) Request, Time1
2) $E_{KPv(PKA)}$ ($K_{PbCN,}$ Request, Time1)
5) $E_{KPv(PKA)}$ ($K_{PbMR,}$ Request, Time2)
4) Request, Time2
$D_{KPb(PKA)}$ ($K_{PbCN,}$ Request, Time1)
$D_{KPb(PKA)}$ ($K_{PbMR,}$ Request, Time2)
3) E_{KPbCN} (MR,, RN1)
MR (MH/HA/CN/ MN)
$D_{KPv(CN)}$ (MR, RN1)
CN (MH/HA/MR/ MN)
$D_{KPv(MR)}$ (RN1, CK2)
6) E_{KPbMR} (RN1,, CK2)
7) $E_{KPb(CN)}$ ($E_{KPv(MR)}$[CK2, K_{Sc}])
$D_{KPv(CN)}$ ($D_{KPb(MR)}$[CK2, K_{Sc})
8) E_{KSc} (Message1, $Message_n$, $Message_{n+1}$ ...)

Figure 6 : PKE mechanism

We proposed to use PKI mechanism in distribution of the RN Key. Below are the descriptions on how PKE mechanism can be applied to scenario as illustrated in Figure3:

1) The MR will request the Public Key of CN from a trusted CA. This request will have an association to Time (t1).

2) CA responded to the MR by encrypting the message using CA's private key. This way the MR can decrypt the message using CA's public key which is openly available. The message contains CN's public key and the original request from MR, and the time stamp. This is to allow MR compared the message and hence authenticate the integrity of the message.

3) MR can then encrypt message (which is the RN1 being generated) with CN's public key. This message will contain the MR's identifier as well.

4) CN decrypts the message using its own Private Key and retrieves the MR's identifier as well as the RN1. CN will send request to the trusted CA in order to obtain MR's Public Key. This step is similar to how MR obtained CN's Public Key.

5) CA will perform identical process (similar to step 2) which is to deliver MR's Public Key to CN.

6) CN can now encrypt a new message to MR, containing the RN1 as well as newly generated Secret Key. When the message arrived to MR,

7) By now the Secret Key has been securely shared between MR and CN and the further communications done between MR and CN will be via this symmetric encryption.

8) CN decrypts the message via the Secret Key and in return will start using this Secret Key for further communications to MR.

From the illustrations shown in Figure6, the concept is as simple as, to setup central and trusted Certificate Authority (mostly to be managed by the Internet Service Provider). Only simple block encryption is needed and this system's strength is maximizing the Public/Private Key Exchange mechanism to distribute the RN. Step 3 and Step 6 proof that the RN being used for authentication purposes and both MR and CN can safely verify each's identity.

In simple, our new idea is to enhance a fast authentication method, whereby the HA will firstly store RN when the MR/MNN booted up. And the HA will exchange and compare the RN of each transaction of roaming for authentication, whenever the associated MR/MNN moved or roamed away. Once the same RN is matched and the verification of genuine and accuracy of transaction can be claimed and authenticated.

From there on, Step 7 and Step 8 will only need a faster cryptography mechanism which is Symmetric Encryption, as compared to the IPSec.

IV. BENEFITS

In terms of NEMO protocol efficiency, especially in space of the security design, our proposal provided a faster and easier authentication method.

This idea can be implemented in either software or hardware level. As for software implementation, a mathematical Random Number Generator can be written and embedded into as part of the NEMO protocol standards. As in hardware's perspective, one can implement this Random Number feature and stored in flash memory of the hardware chip. Nowadays in the embedded or consumer electronic markets, hardware chips often already provided Random Number generation capabilities.

The randomness will base on the length of the key and technically speaking, a randomness of 64-bits will already sufficiently provide good randomness. And depending on implementation the Random Number (the key) can be expanded into 128bits. Currently, by following the NEMO specification the authentication can be done in IPSec's Authentication Header, or via a nonce (also known as a random number) that associated with BU. In order to keep overall BU size to be minimum, this nonce has a limited size of 16 bits and has to be used in association with cookies of BU. So as opposed to our 64-bit RN proposal, our solution will be more secure in term of the advantage in the size of RN. On top of that, our solution also make authentication achievable at early stage rather than after BU-BAck protocol exchanged. We can also argue that 128-bit RN may provide even better randomness, but we shall also consider the transactions of the communications whether 128-bit processing will defer performance over smaller scale of transactions.

In long run, our new idea can also provide a chance to eliminate the use of nonce in BU as we foresee in future the challenges of reducing usage of BU under frequent NEMO-MR-switching and intense nested looping environment.

V. CONCLUSION

We showed the current standard specification is especially under intense vulnerability under nested looping condition, of which the efficiency of the protocol as a whole is being dropped.

We also proposed a solution for faster and simpler authentication via Random Number verification method coupled with PKI mechanism to exchange the Random Number. Once, RN being used and successfully provided a channel for authentication for both MR and CN or even MNN within the MR, the rest of the communications between two parties can be via a faster Symmetric Encryption as opposed to the slower IPSec.

This mechanism is fast, simple and cheap to implement, and do not have to rely on the tedious bi-directional tunnel that can accommodate the BU-BAck exchanges and at the same time also achieve authentication objective. Most importantly, achieve authentication before reaching the BU-BAck algorithm.

While compare with the current NEMO standards of proposing to use BU-BAck's "Nonce" (also another kind of random value) together with BU's cookies as an authentication method, our idea provide a faster and earlier authentication exchange methodology, rather then slower and later authentication.

By implementing this new idea, the BU-BAck exchange time at protocol and system level will become shorter because the authentication portion is already done ahead and hence reduce overall protocol processing time.

REFERENCES

[1] Johnson, D., Perkins, C., Arkko, J., "Mobility Support in IPv6", IETF RFC3775, June 2004

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

[3] P. Thubert., R. Wakikawa., V. Devarapalli., "NEMO Home Network Models". Internet Draft, IETF. Draft-ietf-nemo-home-network-models-06.txt. February 2006.

[4] T. Ernst., H-Y. Latch., "Network Mobility Support Terminology". Internet Draft, IETF. Draft-ietf-nemo-terminology-06.txt. November 2006.

[5] T.K. Tan., A. Samsudin., "Secure Hashing of the NEMO Mobile Router Communications". MICC-ICON 2005, IEEE 05EX1235, November 2005

[6] T.K. Tan., A. Samsudin., "PKI and Secret Key Cryptography Implementation for NEMO Security". *Proceedings of the Int. Conf. on Computer and Communication Engineering, ICCCE'06* pg168, 2006.

[7] T. Kniveton., P. Thubert., "Mobile Network Prefix Delegation". Internet Draft, IETF. draft-ietf-nemo-prefix-delegation-00.txt. August 2005.

[8] Deng, R. H., Zhou, J., Bao, F., "Defending Against Redirection Attacks in Mobile IP" ACM CC @ '02 November 2002

[9] Qiu, Y,. Zhou, J., Bao, F., "Protecting All Traffics Channels in Mobile IPv6 Network" WCNC 2004 @ '02 November 2002

[10] F. Dupont., J-M. Combes. Using IPsec between Mobile Nodes and Correspondent IPv6 Nodes. Internet Draft, IETF. draft-ietf-mip6-cn-ipsec-03.txt. August 2006.

[11] J. Arkko, V. Devarapalli and F. Dupont. Using IPsec to Protect Mobile IPv6 Signaling between Mobile Nodes and Home Agents. IETF RFC3776. June 2004.

[12] Stallings, W., "Cryptography and Network Security", Third Edition, Prentice Hall

[13] Petrescu, A., Olivereau, A., Janneteau, C. and Lach H.-Y., "Threats for Basic Network Mpbility Support (NEMO threats)" Internet Draft, IETF. Draft-petrescu-nemo-threats-01.txt. January 2004.

[14] Conta, A., Deering, S., "Generic Packet Tunneling in IPv6 Specification:, RFC2473, December 1998.

[15] Kent, S., Atkinson, R., "IP Encapsulating Security Payload (ESP)", RFC2406, November 1998.