International Journal of Computing and Network Technology

Functional Scheme for IPv6 Mobile Handoff

Hengky Susanto and Byung Guk Kim

Department of Computer Science, University of Massachusetts at Lowell, USA

Email Address: {hsusanto, kim}@cs.uml.edu

Received: 28 Oct. 2013, Revised 2 Dec. 2013, Accepted 30 Dec. 2013, Published 1 Jan. 2014

Abstract: As a mobile node switches from one subnet to another, Mobile IPv6 requires that it authenticate the new Care of Address of the new subnet with the corresponding node. The authentication process, called Return Routability (RR) test, is required in every handoff and can delay the handoff process by a round trip delay between mobile and corresponding nodes. In this paper, a mechanism is proposed to reduce the latency involved in RR tests and binding association. It eliminates triangular routing delay in RR tests by using multiple authentication tokens or easily regenerative tokens, so that the round trip delay in performing RR tests can be eliminated.

Keywords: Wireless Network, Wireless Handoff, IPv6, QoS.

I. Introduction

In a wireless network, a mobile node (MN) acquires a new *care of address* (CoA) from the new access point. MN needs to inform the new CoA both to the Home Agent and (HA) the corresponding node (CN) [20]. In order to prevent an imposter of MN from generating a spurious CoA, a new CoA is protected by an authentication procedure, called the Return Routability (RR) test. In mobile IPv6 [1], MN in required to send two messages carrying the new CoA to CN: one via the home agent and the other directly to CN. In response, CN generates tokens and send them in different paths: one via the home agent and the other directly to MN. When two response messages are received, MN combines both tokens to formulate a new message encryption key. Since both tokens are known to CN, it is able to formulate the identical message encryption key.

It can easily be seen that an RR test incurs at least a round trip delay (a longer delay between the one via the home agent and the other between MN and CN) each time a handoff takes place. In order to reduce a handoff delay, the previous access point is proposed to act as the home agent until the CoA is authenticated [2]. Another proposal in [3] allows an access point to forward information to its neighbor as soon as it learns that MN has detached itself from its network. In [4], a fast handoff scheme in Session Initiation Protocol (SIP) is suggested. However, SIP is an application-layer process and is not adequate to handle handoff at the network layer. Using higher priorities for handoff messages is considered in [5]. In order to expedite the authentication process of MN, MN is proposed to perform authentication with neighboring access points prior the handoff operation [6]. This is valid when all access points share information with each other. In [7], RR test is performed in the old access point prior to handoff so that data can be continued to be forwarded to CN. However, MN and CN will perform another RR test with the new access point after the handoff. Another study in [8] suggests that RR test can be initialized via old and new access points instead of MN, such that the delay involved in the authentication process can be reduced. However, MN still needs to be involved in a later stage of authentication and gains in handoff delays may not be substantial. There are other research works that focuses on combating man-in-the-middle attack in the RR test. Authors of [13] proposed a scheme where attacker is residing at the same location as MN. The scheme requires the involvement of NAR (new access point) and PAR (previous access point) to assist CN in authenticating MN and the procedure involves exchanging information between MN, CN, HA, NAR, and PAR, which is more complicate and caused longer delay compare to the original RR test in [1]. Furthermore, the author of [14] and [15] proposed the used of digital signature and Diffie Hellman scheme respectably. However, [14] and [15] offer stronger

security scheme but the scheme does not reduce the handoff delay.

Additionally, [1] also discusses the security consideration and features in Mobile handoff in IPv6. Prior to performing RR test, HA and MN must establish a secure tunnel and perform IP security (IPsec) protocol. Furthermore, [1] proposes HA, MN, and CN to employ message exchange verification, cryptography to secure binding update (BU), symmetric exchanges to avoid reflection attack, and the employment of IPSec Encapsulating Security Payload (ESP) between HA and MN to limit the possibility of attacker seeing nonce in HoT. Moreover, MNs are often found itself on an insecure link such as public access wireless LAN. Thus, attacker could send packets that appear to come from MN without attacking tunnel itself. The attacker may simply send packets with the source address set to the MN's home address. Thus, the author of [1] recommended employing end-to-end security or additional protection when MN is away from home network. Another of security threat discussed in [1] is that a routing header could be used in reflection attack which is an attack that is designed to bypass firewall. The used of routing header is to allow a node to get around IP-address based ruled in firewalls but it also allows refection of traffic to other nodes. This requires another protection although the threat exists with routing headers in general even if the usage that mobile IPv6 requires is safe. These extra security measures have to be initiated, set up, and performed in every handoff thus caused a delay in every handoff.

In this paper, we propose functional token scheme, where handoff-related information is exchanged at the beginning of a connection between MN and CN. RR test is not required to be performed in subsequent handoffs, thus round trip delay is eliminated. An elaborate scheme is to generate a chain of authentication tokens locally at MN so that MN is not required to send a token to CN in each handoff. Additionally, as proposed in [1,18] that binding between MN and CN is only performed after HA and MN re-establish a secure tunnel between them but the functional token scheme allows MN quickly binds to CN, as soon as MN obtains a new CoA from the new access point, without waiting upon the completion of MN and HA secure tunnel establishment. By reducing delays in handoffs, the proposed schemes are expected to be particularly beneficial to real-time connections. Furthermore, we also proposed the security measure taken to combat the man-in-the-middle attack, reflection attack, and any attack which involved traffic redirection.

The remainder of the paper is organized as follows. Details of mobile IPv6 RR test are described in the next section. In section 3, functional token scheme is described. A brief remark on performance implication is included. Finally, in Section 4, concluding remarks are presented.

II. Per-handoff RR-test in Mobile IPV6

After a successful handoff, MN informs the new CoA to the home agent and CN in a Bind Update (BU) message, as illustrated in figure 1. In order to prevent an impersonator to generate a spurious BU message, an authentication process, called Return Routability (RR) test, is performed in mobile IPv6 [1]. Prior to sending a BU, MN launches two messages to CN to initiate the RR test as shown in figure 1. Home Test Init (HoTI) is sent via Home Agent (HA) whereas Care of Test Init (COTI) is sent directly to CN. CN responds to both messages by sending Home Test (HoT) and Care of Test (CoT) in response to HoTI and CoTI, respectively. HoT is sent via HA and CoT is sent directly to MN. The critical feature of authentication is that HoT and CoT contain *home keygen* (HK) and *care of keygen* (CK) tokens, respectively. BU message is encrypted based on a *binding management key* K_{bm}, which is derived from CK and HK tokens.

HK and CK tokens are generated as follows. Let K_{CN} denote the secret key of CN. Let 'nonce' denote a unique number, which is guaranteed not to be repeated.

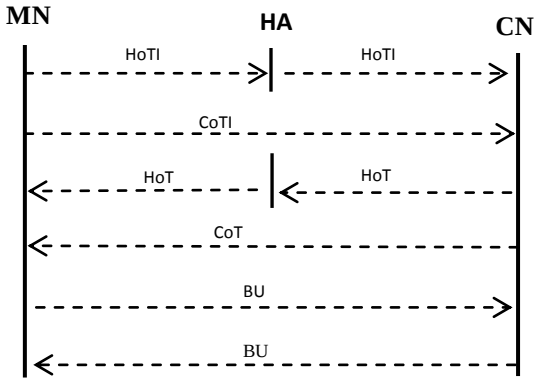

Fig. 1. Return Routatbility Test.

HK is generated first by forming a concatenated string of home address of MN (Haddr), 'nonce' and '0', and then hashing it by the secrete key of CN. Namely

$$\mathrm{HK} = \mathrm{First}\left(64, \mathrm{HMAC\_SHA3}\left(K_{CN}(\mathrm{Haddr}|\mathrm{nonce}|0)\right)\right),$$

where HMAC_SHA3 is a cryptographic hash function and function 'First' truncates a message to the first 64 bits. HMAC implants the secret key into the data and then compute a hash value from it. HMAC_SHA3 is a particular function which uses SHA-3 cryptographic hash function in the calculation of an HMAC function [1,9,12].

HMAC_SHA3 is a particular function which uses SHA-3 cryptographic hash function in the calculation of an HMAC [1,9,12]. As this paper is written, SHA-3 is certified to be the standard for cryptographic hash function [16]. Although, the output from the hash function is 96 bits, the reduced token size of 64 is sufficient to protect it from spoof [1]. The author claims that the attacker has to send a large number of messages before attacker can successfully launches the blindly spoof if the attacker is able to intersect the connection and recognize which packets carry the Binding Update (BU). However, the security strength of BU is beyond the scoop of this paper.

Similarly, CK token is generated by

$$\mathrm{CK} = \mathrm{First}\left(64, \mathrm{HMAC\_SHA3}\left(K_{CN}(\mathrm{CoA}|\mathrm{nonce}|1)\right)\right).$$

HK and CK tokens are sent to MN in HoT can CoT messages, respectively. Like HoTI and CoTI messages, HoT and CoT messages are delivered in different paths. HoT is sent via HA. HA forwards the HoT message to MN though a secure tunnel, ensuring a secure delivery. CoT message, on the other hand, is sent directly to MN. Upon receiving HK and CK tokens, MN hashes both keys to form a binding management key K_{bm} as follows.

$$K_{bm} = \mathrm{SHA3}(\mathrm{HK} \mid \mathrm{CK}).$$

The binding management key, K_{bm}, is then used to encrypt the BU message to CN. Since CN has copies of HK and CK, it is also able to generate K_{bm} and can authenticate that BU indeed was generated from the intended MN.

III. Per-Connection RR-test Schemes

Two schemes presented in this section perform RR test at the beginning of a connection between MN and CN, and eliminate per-handoff RR test in subsequent handoffs. By reducing the delay involved in RR test, each handoff performance can be improved.

A. Functional Token Scheme

In the authentication pool scheme, MN generates a set of tokens, which is included in the BU message. When the size of pool is large, the size of BU message may be increase, resulting in slightly longer transmission time of the initial BU message. Moreover, the initial BU message containing the pool of tokens may be intercepted. Although BU message is encrypted, there may be concerns that it may be compromised while tokens in the pool are being used. Functional token scheme addresses the concerns by adopting the one-time password mechanism in [10]. Let function $F(x)$ be a computationally inexpensive linear function. However, it is computationally difficult to invert $F(x)$. Some of hash functions have such property [10]. Consider repeatedly applying $F(x)$ recursively. Let $h_1(x) = F(x)$ and

$$h_i(x) = F\left(h_{i-1}(x)\right) = F^{(i)}(x),$$

where $F^{(i)}(x)$ denotes i successive applications of function $F(x)$.

From the property of $F(x)$, it is relatively straightforward to obtain h_i from h_{i-1}, but not the other way around. The one-time password system first generates $\{h_i(x) \mid 1 \le i \le n\}$, and uses passwords in the backward sequence of $\{h_i\}$. Namely, the k-th password, pw_k , is given by

$$pw_k = \{h_{n-k}\}, \quad for\ 1 \le k \le n.$$

When n=1000, for example, the sequence of one-time passwords becomes $F^{(999)}(x)$, $F^{(998)}(x)$, ..., $F^{(2)}(x), F^{(1)}(x)$. Thus, when a user logs in with the i-th password pw_i, the system applies the function $F()$ to pw_i and checks if it matches with the previous password pw_{i-1}. Namely, if $pw_{i-1} = F(pw_i)$, the new password is verified. In order to verify a new password, the system is required to record only the last password used, which is substantially efficient than maintaining a table of passwords.

We adopt the one-time password system to creating a hash chain of authentication tokens. After a connection is established, both MN and CN obtain the identical K_{bm} as in section 2. Now, MN and CN independently compute

$$h_i(K_{bm}) = F\left(h_{i-1}(K_{bm})\right) = F^{(i)}(K_{bm}),$$

for $1 \le i \le n$. MN keeps the entire table of

$$\{h_i(K_{bm}) \mid 1 \le i \le n\},$$

whereas CN only keeps the last token, $h_n(K_{bm})$. At the first handoff, MN includes the 2^{nd} from the last token,

$h_{n-1}(K_{bm})$ in the BU message (which is encrypted by K_{bm}). When CN receives the BU message, it decrypts it with K_{bm} and applies $F(.)$ to the authentication token in the BU message to check if

$$F(h_{n-1}(K_{bm})) = F.$$

If they check out, the authentication token in the BU message is indeed valid. CN then keeps the authentication token, $h_{n-1}(K_{bm})$ for the next round of handoff. This method allows the authentication to be performed without exchanging or distributing information over network. Also, the amount of information required to be maintained at CN is kept to its minimum.

When MN and CN run out tokens, they can be replenished by generating $\{h_i(x) \mid 1 \le i \le n\}$ with a different seed for x. Generating a new seed value can be implemented in many different ways. One simple example is to combine the token kept in MN or CN with K_{bm}, such that

$$x = h_1 \oplus K_{bm},$$

assuming that the regeneration of tokens takes place when MN or CN has the token of h_1. As in the initial connection establishment, MN and CN generate new $\{h_i(x) \mid 1 \le i \le n\}$ with the new seed, and MN keeps the entire sequence whereas CN keeps the last computed value. The approach eliminates needs for synchronizing when to replenish tokens and for distributing a new pool of tokens.

Next, we illustrate a practical implementation of functional scheme. In this example, SHA-3 is selected to illustrate hash chain because it has the computation complexity of 2^{512} to break the hash collisions [16]. The token generation can be done as follow

$$t = \text{SHA-3}(K_{bm}).$$

Secondly, both CN and MN generate a hash chain using SHA-3, let F be SHA-3. Thus, we have

$$t^n = F\left(F\left(F(\ldots(F(K_{bm}))\ldots)\right)\right),$$

where t^n is the n^{th} token generated by the hash chain.

B. Securing Functional Token

One possible weakness of the functional scheme is, when a functional token is embedded in BU message, Man-in-the-middle attack may intercept BU message and exposes the token. To avoid such attack, we extend the functional scheme with encryption scheme to guard the functional scheme with different unique secret-shared-key in each handoff. The objective is to encrypt BU packet with one-time shared key that is only known to both MN and CN and these shared keys can be generated with the hash chain [10] for each handoff. The implementation one-time shared key as follow. Let s be the seed, F be the hash function, t be the functional token, and y^k be the k^{th} encryption key used in the k^{th} handoff. We have

$$y^1 = F(s),$$

$$y^k = F^{(k)}(y^{k-1}),$$

$$t' = E_y^k(t),$$

Where $E_y^k(z)$ is a function that encrypts z using one-time shared key y^k. This approach allows MN and CN continuously generate unique shared keys for as long as needed. The recipient retrieves the functional token with $t = D_y^k(t')$.

The seed will be discarded once the connection is terminated. Furthermore, the seed of the one-time shared keys s can be generated using the existing management key K_{bm} which is already known to both MN and CN. One simple example is to combine a token that is kept in MN and CN with K_{bm} and manipulate K_{bm} such that

$$s = h_1 \oplus \text{inverse}(K_{bm}),$$

where $\text{inverse}(x)$ is function that inverse the binary number of value x assuming that the regeneration of tokens takes place when MN or CN has the token of h_1. Furthermore, for $j = 0,1,2,\ldots,\infty$, secret shared key k_{j+1} is generated and key k_j is discarded only after the k^{th} handoff is successfully performed.

The other possible weakness is when the-man-in-the-middle attacker modifies the content of BU message. In RR test scheme uses K_{bm} and cryptographic value "Authenticator" to authenticate the authenticity of the message whether the message has been tempered. RR test scheme's message authentication [1] is done as follow:

Data = care-of address | CN's IP address| MH data.

Authenticator = First (96, HMAC_SHA3 (K_{bm}, Data).

MH data is the content of mobility header which is an extension packet header used by MN, CN, and HA in all messaging related to the creation and management of bindings. Thus, to assure the authenticity of BU message, the CN computes its own authenticator value and compares it with the authenticator sent by MN. If the authenticator values are different then clearly the message has been tempered. The functional scheme also adopts message validation authentication using the functional token instead of using K_{bm}.

C. Initializing Functional Scheme

MN-CN connection establishment can be vulnerable to the-man-in-the-middle-attack, where the attacker may illegally acquire information to break the functional scheme and impersonate MN or CN, because MN and CN may not familiar with each other. In this section we propose an approach to assure that the connected party is indeed the perpetuated party.

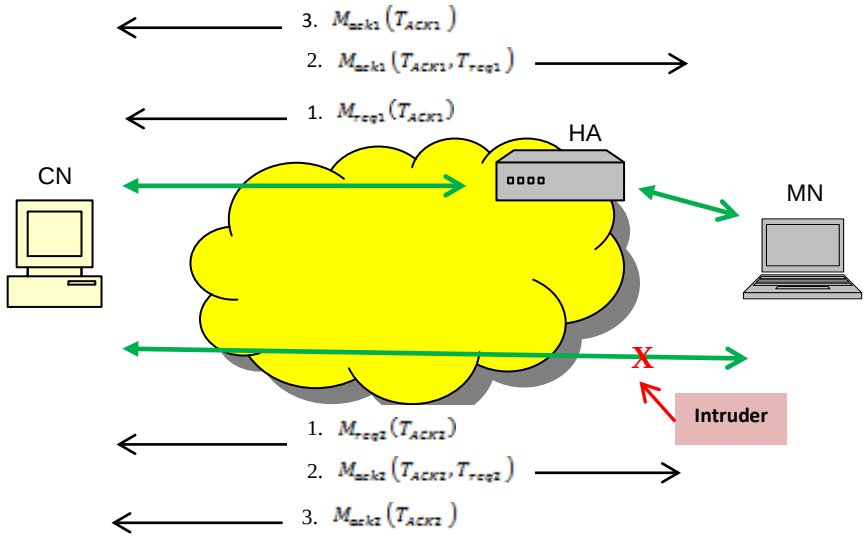

Fig. 2. Three ways handshakes.

Here, we consider one of the two links between MN and CN is compromised meaning the attacker randomly taps one of the two links, the direct or indirectly link through HA. To combat such attacker, First, MN and CN must be able to establish a secure connection with the intended party. Second, MN and CN must deliver the information safely across the network. The attacker may be able to acquire incomplete information but may not sufficient to break the functional scheme. To overcome such challenge, we introduce a three ways handshake using modified RR test [1] to provide more secure connection establishment, as depicted in figure 2. Three ways handshake is done as follow: First, MN must send two request messages M_{req1} and M_{req2} to CN where M_{req1} is sent over the direct link to CN and M_{req2} is sent trough HA. In addition, M_{req1} is embedded with token T_{req1} and M_{req2} with T_{req2}. Second, CN responses both messages with ACK message M_{ack1} and M_{ack2}; each message is embedded with token T_{ack1} or T_{ack2}. M_{ack1} is sent directly to MN and M_{ack2} through HA. Also, token T_{req1} and T_{req2} are also included in M_{req1} or M_{req2} depending upon which link they came from, such that token T_{req1} is included in M_{req1} and T_{req2} is in M_{req2}. Third, MN responds CN's M_{ack} with similar fashion by sending accepting message M_{acpt1} and M_{acpt2}; token T_{ack1} is included in M_{acpt1} and T_{ack2} is in M_{acpt2}. This allows MN and CN to verify each other. MN and CN can prove their legitimacy by showing that they receive both tokens. By dividing information and sending it over two links, the attacker may only able to acquire partial information.

Once MN-CN verification is completed, MN and CN can generate two function tokens of

$$F(T_{CN-MN} | T_{CN-HA-MN})$$

and

$$F(T_{MN-CN} | T_{MN-HA-CN}),$$

where F is a one way hash function, and the selection process of deciding, which two tokens is used , can done by

$$\max\left(F(T_{req1}|T_{req2}), F(T_{ack1}|T_{ack2})\right).$$

IV. Performance Analysis

In Mobile IPv6, to establish secure a channel between HA-MN delay and two round-trip delays for RR test are required at each handoff: a round-trip delay for exchanges of HOTI/HOT and COTI/COT messages, and another round-trip delay for BU message and its acknowledgement. Since HOTI and HOT messages have to be delivered via the home agent, it may take slightly longer than the other message exchanges. Establishing to secure a channel between HA-MN and RR test schemes

require two round-trip delays at the beginning of a connection between MN and CN, as in Mobile IPv6. However, all subsequent handoffs are performed with exchanges of BU messages only, thereby reducing handoff delay to a single round-trip delay.

In order to provide better comparison between RR test and functional scheme, we present a numerical example by analyzing the handoff delay, where HA is located in these cities: Boston, San Francisco, Hong Kong, Singapore, and Jakarta. Furthermore, both MN and CN are located in the same location in a city near Boston, Lowell, and the RTT delay between MN and CN is about 0.1 milliseconds (*ms*). The simulation setup is there are nodes acting as HA residing in those cities mentioned above, a mobile node acting as MN, and a wired server acting CN. Furthermore, to make the case clearer, MN and CN are connected to the same router that has the functionality of both wired and wireless connection.

Table 1. Handoff delay with RR Test with HA is located in various cities.

City	RR-test Delay
Boston	109 *ms*
San Francisco	121 *ms*
Hong Kong	265 *ms*
Singapore	279 *ms*
Jakarta	285 *ms*

Our experiment shows that, depending upon the location of HA, the handoff using RR Test may take as long as 285 milliseconds and as short as 109 milliseconds. However, the handoff delay using functional scheme is generally only about 0.1 milliseconds. It is because the HA servers are located far from MN-CN and RR test requires to MN-CN to forward their information through HA. Clearly, by removing HA from the handoff protocol, functional scheme allows a direct communication between MN and CN, which resulting in a much shorter delay. Hence, handoff delay with function scheme is only determined by the delay between MN and CN. This experiment shows that, in comparison to the functional scheme, the handoff delay using RR test increases as the distance between HA and CN-MN increases. Thus, employing functional test reduces the handoff delay. However, when HA, CN, and MN are located very close to each other, the performance difference between the two schemes becomes less evident.

A. Security Analysis

In this section, we discuss functional scheme's security measure in mobile IPv6. Most of the potential threats are concerned with false binding of MN with an attacker which usually resulting in Denial of Service (DoS) attack, Man-in-the-Middle, hijacking, confidentiality, starvation, and impersonating attacks [1].

The security strength of function scheme may be determent by the strength of the hash function used in the functional scheme. Since today's Secure Hash Standard is typically very difficult to break, breaking functional scheme may require enormous effort. For example, Assume AES or SHA-512 [12] is adopted into functional scheme, AES requires computation complexity of 2^{48} and a memory complexity of 2^{32} and SHA-512 requires computation complexity of 2^{128} [12]. Then deciphering the secret key and the functional token may involve many resources, consume a lot of time, and require multiple attempts to bind with CN. Conversely, multiple binding attempts can be easily detected by CN by counting the number of attempts made and later take an appropriate action.

Another potential attack is the reflection attack, where the attacker poses as a valid MN, sends a challenge to CN for verification, and CN responds to the attacker with some value of x. Next, the attacker opens a new connection to CN, CN sends a challenge to the attacker, and the attacker responds to MN with value x that is received from the previous connection with CN. However, functional token scheme prevents reflection attack between MN and CN because the binding request requires a unique token for each binding and CN may ignore a binding request if the request is embedded with invalid token which include token from previous handoff.

Another case to consider is when attacker initially poses as legitimate MN but redirects the traffic to another node to launch a DoS or bombing attack. To prevent such attack, CN sends a re-binding request to MN's new address in y unit times after handoff procedure is completed. If MN responds with the proper token within y than MN continues with the connection otherwise terminate the connection. This approach is possible because MN and CN can regenerate tokens. One candidate for deciding y is

$$y = 2 * RTT.$$

In addition to that, if attacker is capable of intercepting and responding to the rebind request, then it means it is possible that the attacker is located at the same location as MN. Thus, if the attacker redirects traffic and launches bombing attack to its own location then the

traffic will clog the access point of where the attacker is located. In consequence, attacker also suffers form its own attack. Another case of this scenario is that attacker is residing at CN's location and this will require different approach, which is beyond the scope of this discussion.

In order to strengthen the security, the authors of [1] propose a tunnel between MN and HA should be protected to ensure the used of proper source address and optimal cryptographic protection. HA ensures that MN's outer IP address (CoA) corresponds to the actual current location of MN to prevent spoofed packets which appears from MN. This checking methodology is not sufficient if the attacker is residing at the ingress router and capable of modifying the packet header. Thus, MN and HA requires stronger end-to-end security and additional tunnel protection. However, the functional token scheme bypasses this step and CN can quickly validate MN's authenticity by checking whether token $x^k = h_{n-k}(K_{bm})$ without relying on the MN and HA end-to-end protection.

B. Limitation

Functional token scheme is not designed to solve problem when attacker changes the content of the BU packet. For example, an attacker intercepts BU packet and alter one of the bit in the packet. As the result, CN does not receive the correct payload or unable to properly encrypt the packet. The functional token scheme is also not designed to handle this scenario but fortunately there are many proposed solutions to this problem using digital finger printing to assure the authenticity of the content sent over network [12]. Furthermore, the functional scheme requires stronger scheme to secure the connection initialization and the scheme is useless once the attacker acquire the seed value to generate token chain and one-time-secret - shared-key.

However, providing better and stronger security may incur higher cost, such as requirement for more memory and higher computational complexity [19], which may result in poorer performance. This defied the purpose for which it is set up for. Thus, we must consider the tradeoff between security and performance.

V. Concluding Remarks

In mobile IPv6, an RR test is performed in each handover in order to authenticate the new CoA of mobile node. In this paper, functional token scheme is proposed to reduce the time required to authenticate the new CoA of MN which relies on generating a pool of authentication tokens. In each subsequent handoff, both MN and CN agree on a linear function with difficult inverse operation so that a pool of keys may be generated dynamically. The scheme is also expended with one-time-shared-secret-key scheme to assure the integrity of the information in the BU packet. Thus, this scheme may require CN possesses some computational capability. In the future work, we will investigate how the functional scheme may impact the lower layer protocol in the wireless network.

REFERENCE

[1] D. Johnson, C. Perkins, and J Arkko, "Mobility Support in IPV6", *RFC 3775*, June 2004.

[2] E. Koodli, "Fast Handovers for Mobile IPv6", *RFC 4068*, July 2005.

[3] S. Balasubramaniam and J. Indulska, "Vertical Handover Supporting Pervasive Computing in Future Wireless Networks" *Computer Comm. Journal, Special Issue on 4G/Future Wireless networks*, 2004.

[4] A. Dutta, S. Madhani, and W. Chen, "Fast-handoff Schemes for Application Layer Mobility Management", *Personal, Indoor and Mobile Radio Comm*., 2004**.**

[5] D. Tandjaoui, N. Badache, and I Romdhani, "A New Prioritized Fast Handoff Protocol for Mobile IP**",** *13th European Wireless Conf.*, 2007**.**

[6] P.S. Kim and J.S. Han, "New Authorizing Binding to Reduce Binding Latency during Mobile UPv6 Handover Procedure", *Int'l Journal of Computer Science and Network Security*, 2006.

[7] H. Nguyen et al, "A Temporary Binding Update in Fast Handover for Mobile IPv6", *Int'l Conf. on Convergence Information Technology*, 2007.

[8] J.Zhang and D.A. J. Pearce, "Proactive care-of address test for route optimization in FMIPv6", *Wireless Mobile Applications and Services on WLAN*, 2005.

[9] E. Rescorla, "Diffie-Hellman Key Agreement Method", *RFC 2631*, 1999.

[10] L. Lamport. "Password Authentication with Insecure Communication", *Comm. of the ACM*, 1981.

[11] H. Krawczyk, M. Bellare, R. Canetti, "HMAC*:* Keyed-Hashing for Message Authentication*", RFC 2104*, 1997.

[12] Jie wang, "Computer Network Security, Theory and Practice", *Springer, Berlin*, 2009.

[13] Y. Mun, K. Lee, S. Ryu, and T. Shin, "Using Return Routability for Authentication of Fast Handovers in Mobile IPv6". *Conference on Computational Science and Applications*. 2007.

[14] H. Zhu, F. Bao, R. Deng, "securing Return Routability Protocol Against Active Attack", *Vehicular Technology Conference*, 2004.

[15] Y. C. Chen and F. C. Yang "an efficient MIPv6 Return ROutability Scheme Based on Geometrix Computing", *world Academy of Science, Engineering, and Technology*, 2009.

[16] G. Bertoni, J. Daemen, M. Peeters, and G. Assche," The Keccak SHA3", *keccak.noekeon.org*, 2011.

[17] H. Gilbert and T. Peyrin, "Super-Sbox Cryptanalysis: Improved Attacks for AES-like Permutations", Fast Software Encryption, 2009.

[18] Hengky Susanto and B. G. Kim, "Pre-Connection Return Routability Test in Mobile UPv6", *IEEE Conference on Network-Based Information Systems*, 2009.

[19] A. Couch, N. Wu, and H. Susanto, "Toward a cost model for system administration," *Proc. Large Installation System Administration*, 2005.
[20] B. G. Kim, H. Susanto, Hwang S. Lee, "Survey of Vertical Handoff Models in Heterogeneous Wireless Networks", *International Conference on Wireless Network* , 2008.

Hengky Susanto received the BS degree in computer science from University of Massachusetts Amherst in 1999, the MS degree in computer science from University of Massachusetts Lowell in 2004. He also did his post MS degree at Tufts University. He is currently PhD candidate in computer science at University of Massachusetts Lowell. His main interest includes wired and wireless network, network multimedia, protocol design, and network pricing. He has also received several awards including the best USENIX LISA paper award 2005, EMC Achievement award 2000, and StorageNetwork Shine Award 2002.

Byung Guk kim received the BS degree in electrical engineering from Seoul National University, South Korea, in 1975, the MS in computer science from University of Massachusetts Amherst in 1978, the PhD Degree in computer science from University of Massachusetts Amherst degree in computer science from University of Massachusetts Amherst in 1980. He is an Associate professor and undergraduate coordinator in the Department of Computer Science, University of Massachusetts Lowell. His research interests include network performance, wireless network, network multimedia, network traffic congestion and management, and Bioinformatics.