Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 957690, 10 pages
doi:10.1155/2009/957690

Research Article

A Secure and Lightweight Approach for Routing Optimization in Mobile IPv6

Sehwa Song, Hyoung-Kee Choi, and Jung-Yoon Kim

School of Information and Communications Engineering, Sungkyunkwan University, Chunchun-dong 300, Suwon 440-746, South Korea

Correspondence should be addressed to Hyoung-Kee Choi, hkchoi@ece.skku.ac.kr

Received 31 January 2009; Revised 17 April 2009; Accepted 20 May 2009

Recommended by Shuhui Yang

Mobility support is an essential part of IPv6 because we have recently seen sharp increases in the number of mobile users. A security weakness in mobility support has a direct consequence on the security of users because it obscures the distinction between devices and users. Unfortunately, a malicious and unauthenticated message in mobility support may open a security hole for intruders by supplying an easy mean to launch an attack that hijacks an ongoing session to a location chosen by the intruder. In this paper, we show how to thwart such a session hijacking attack by authenticating a suspicious message. Although much research has been directed toward addressing similar problems, we contend that our proposed protocol would outperform other proposals that have been advanced. This claim is based on observations that the proposed protocol has strengths such as light computational load, backward compatibility, and dependable operation. The results of in-depth performance evaluation show that our protocol achieves strong security and at the same time requires minimal computational overhead.

1. Introduction

Mobile networking technologies, along with the proliferation of numerous portable and wireless devices, promise to change people's perceptions of the Internet. Mobility support in Internet Protocol Version 6 (IPv6) is considered particularly important because mobile devices are predicted to account for a significant portion of Internet users during the lifetime of IPv6. Mobile IPv6 (MIPv6) is an IP-layer mobility protocol for the IPv6 Internet [1]. MIPv6 allows an IPv6 mobile node (MN) to change its location on an IPv6 network and still maintain its existing connections to corresponding nodes (CNs).

In Mobile IP, an MN is addressed by two addresses, a home address (HoA) and a care-of address (CoA). An MN has its stationary HoA at its home subnet and changes its temporary CoA whenever visiting a foreign subnet. This dual address mechanism makes it possible to route packets to an MN no matter where it is attached in the Internet. Also, the complex dynamics that occur in the face of sequential handovers are absolutely transparent to transport and higher-layer protocols.

A link between an MN and a CN in Mobile IPv4 (MIPv4) is always detoured via the MN's Home Agent (HA), forming a triangular path [2]. Packets from the CN are routed to the HA and then tunneled, based on CoA, to the MN's location at the time. MIPv6 contained improvements to this rather inefficient routing. The new mechanism, called Route Optimization (RO), requires the MN to update its CoA at the CN whenever the MN changes its point of attachment to the network. The RO in MIPv6 provides an illusion to protocol layers above MIPv6 of continuing to be connected to the MN located at its HoA address. At the same time, the RO rectifies the suboptimal triangular routing by connecting the CN directly to the MN. The MN may choose to inform the CN of its new CoA by using a binding message, thereby allowing the CN to send subsequent packets directly to the MN, bypassing the HA. A binding is the association of the MN's HoA with the CoA for that MN. Unfortunately, malicious and unauthenticated binding messages may open a security hole for intruders by supplying an easy means to launch what are called redirection attacks that hijack an ongoing session to a location chosen by the intruder. IETF's approach [1] to preventing this type of attack is to authenticate the BU

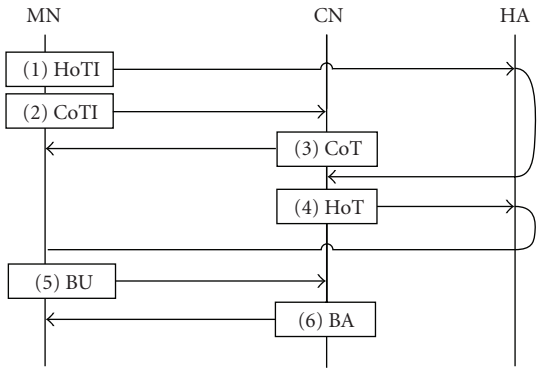

Figure 1: Illustration of secure routing optimization in MIPv6. There are six messages in total. The MN-HA path is securely protected by the IPsec tunnel.

message at the CN and to examine a return path from the CN to the claimed CoA to determine if the address is routable. These two special routines are called Binding Update (BU) and Return Routability (RR), respectively, and we refer to this series of activities as a secure RO in order to emphasize the security aspect in this RO.

In this paper we address the problem of securing the routing optimization. This is a particularly difficult problem because of the following reasons. First, we cannot expect a pre-established secure channel between communicating nodes nor an infrastructure to support secure transactions on behalf of communicating nodes [3]. In addition, the new protocol should be efficient in yielding real-time responsiveness and have a light computational load because delay in the handover greatly affects the quality of service (QoS) in mobile applications. Last but not least, the proposed protocol must be compatible with the legacy protocol to permit a smooth transition.

Our goal in this paper is to take significant steps toward a system that fulfills these criteria. In our protocol the MN creates a secret and sends this secret to the CN twice, once in the direct path to the CN and the other through an indirect path via the HA. The secret is safe from snoopers because it is wrapped in a self-encrypted message. Later, the MN discloses its secret to the public. If the CN can decrypt the MN's early messages with this secret, the CN can confirm the MN's ownership. We evaluated the proposed protocol by comparing its computational expense with five other protocols. The result showed that the proposed protocol was quite efficient and, at the same time, satisfied in a secure manner both ownership and return routability. The objective in this paper is not to explain the cause of network anomalies in the MIPv6. Instead, we seek to demonstrate the utility of new primitives and techniques a future system could exploit for efficient handover.

The paper is organized as follows. Sections 2 and 3 introduce the RO in MIPv6 and discuss related works. Section 4 presents the result of vulnerability analysis. In Section 5, we propose a new secure RO scheme. A performance analysis of the proposed scheme is given in Section 6. Section 7 contains our conclusions.

2. Route Optimization in Mobile IPv6 (MIPv6)

The secure RO in the MIPv6 is composed of six messages and is shown in Figure 1. The first four messages are dedicated to checking the RR of the CoA, and the last two messages are used to authenticate the BU message.

The MN sends the Home Test Init (HoTI) and the Care-of Test Init (CoTI) messages to initiate the binding update, that is, updating the new CoA at the CN. These two messages are sent almost simultaneously but along different paths; the CoTI is sent directly to the CN, and the HoTI is sent indirectly via the HA; (1) are the HoTI and CoTI messages, respectively,

$$\begin{aligned} \text{HoTI} &= \{\text{HoA}, \text{CN}, R_H\}, \\ \text{CoTI} &= \{\text{CoA}, \text{CN}, R_C\}, \end{aligned} \tag{1}$$

R_H and R_C are cookies to match requests with the CN's corresponding responses.

The CN sends the Home Test (HoT) and the Care-of Test (CoT) messages as responses to the previous messages. The HoT and CoT messages are sent, respectively, to the source addresses of the HoTI and CoTI, and follow the same delivery paths as the HoTI and CoTI messages. The HoT and CoT messages are shown, respectively, in

$$\text{HoT} = \{\text{CN}, \text{HoA}, R_H, \text{HT}, i\}, \tag{2}$$

$$\text{CoT} = \{\text{CN}, \text{CoA}, R_C, \text{CT}, j\}. \tag{3}$$

HT and CT are tokens generated by the CN and become a secret key after concatenating these two tokens to authenticate the BU message. HT and CT are shown, respectively, in (4). HT and CT are saved in the CN's hash under the hash indices of i and j. The MN must later return these hash indices in its BU message so that the CN can remain stateless until the BU message is received. These hash indices are included in the HoT and CoT

$$\begin{aligned} \text{HT} &= \text{First64}(H(K_{\text{CN}}, \text{HoA} \| N_i \| 0)), \\ \text{CT} &= \text{First64}\Big(H\Big(K_{\text{CN}}, \text{CoA}\Big\|N_j\Big\|1\Big)\Big). \end{aligned} \tag{4}$$

$H(\cdot)$ is a selected hash function, and First64$(\cdot)$ is a function to choose the first 64 bits in the return string of the hash function. Input to the hash function is the CN's secret key (K_{CN}) and the concatenation of MN's HoA, a nonce value (N_i) and a zero. The generation of CoT is quite similar to the HoT, and extension to the CoT should be straightforward.

The legitimate MN now possesses both tokens and generates a secret key (K_{bm}) as shown in

$$K_{\text{bm}} = H(\text{HT} \| \text{CT}). \tag{5}$$

This marks the end of the RR procedure. The MN may now generate the BU message and is ready to send

$$\text{BU} = \{\text{CoA}, \text{CN}, \text{HoA}, \text{SEQ}, \text{LT}, i, j, \text{MAC}_{\text{BU}}\}. \tag{6}$$

This BU message as shown above is sent from the MN's CoA to the CN. In addition to the CoA, HoA, CN, a sequence number (SEQ), valid lifetime (LT) for this binding update, and the two hash indices are included in the BU message. $\mathrm{MAC}_{\mathrm{BU}}$ is the sign of the BU message using K_{bm}.

On reception of the BU message, the CN recovers K_{bm} from the hash indices included in the BU message and verifies the sign. If the sign proves authentic, the CN accepts the BU message and the MN's CoA by sending an acknowledgment to the MN. The binding acknowledgement (BA) message is shown in

$$\mathrm{BA} = \{\mathrm{CN}, \mathrm{CoA}, \mathrm{SEQ}, \mathrm{LT}, \mathrm{MAC}_{\mathrm{BA}}\}. \tag{7}$$

The security of the RR and BU protocols hinges on the management of HT and CT. Note that no one except the CN can manipulate HT and CT because of the unknown K_{CN}. However, HT and CT are available to anyone in the delivery path because they are delivered in clear text. If an adversary happens to collect a pair of HT and CT in the network, the secure RO is vulnerable to a redirection attack [4].

From a security perspective, the MN's duty as defined in the RFC 3775 is twofold [1]. First, when the MN updates its temporary CoA at the CN, the MN should corroborate to the CN that the CoA is a temporary version of the HoA and that the HoA and CoA are both owned by the MN. The stationary HoA serves as an identifier for the MN. Second, from the perspective of the CN, rather than being informed by the MN that the MN's address has changed to the new CoA, it would be safer for the CN to participate actively in this binding update procedure by confirming the existence and the routability of the MN's CoA. This is very important because a dishonest MN could advertise a fake CoA. The former duty is implemented in the BU, and the latter is accomplished in the RR.

The MIPv6 is an extended version of the IPv6 implemented to support tetherless mobility to nodes but has no role in strengthening the security of the IPv6. Hence, many good security features are excluded from the MIPv6, including authentication. Indeed, authentication to the MN is excluded and furthermore is not necessary in the MIPv6. This is because, first, the security policy in the MIPv6 tries only to maintain a degree of security equal at least to the security of the IPv6 and enforces only authentication of the BU message and the RR. Second, the overhead associated with authentication is too big. Authentication necessitates establishment of a session key for the two nodes, a step that then requires a key management mechanism. Third, at the moment when the MIPv6 starts to work, authentication in the second layer has already been completed. For instance, typical authentication mechanisms in the second layer are Wi-Fi Protected Access2 (WPA2) in 802.11 [5], Privacy and Key Management v2 (PKMv2) in 802.16e [6], and Authentication and Key Agreement (AKA) in Universal Mobile Telecommunications (UMTS) [7]. Additional authentication in the MIPv6 is unnecessary for valid users in the second layer, but nevertheless, the MIPv6 monitors the behavior of these users after authentication.

3. Related Work

One popular approach for a secure RO was to establish a secure relationship between the CN and the MN. The CN first authenticated the MN so as to set up a secure channel and then exchanged useful information over this secure channel. Certificate-based Binding Update (CBU) [8], Hierarchical Certificate-based Binding Update (HCBU) [9], and Leakage-Resilient Security Architecture (LR-AKE) [10] incorporated private key cryptography to establish a secure relationship. Because the MN is authenticated, the CN can trust all messages from the MN. Such attacks as impersonation, message modification, and eavesdropping are quite difficult in the secure channel. As a result, the CN can be sure that CoA is owned by the MN and is reachable. Nonetheless, we contend that the proposed protocol has many advantages over a protocol with private key cryptography as follows.

(1) The certificate management is known to be a big overhead in the operation of asymmetric cryptography. In particular, revoking a certificate and managing the list of revoked certificate are such overheads. The proposed protocol dispenses with the certificate and its management.

(2) The MN and CN may belong to different security domains. In this case interdomain protocol for asymmetric cryptography can be quite subtle, rendering its advantages forfeit. The proposed protocol runs the same irrespective of the domains the both parties belong.

(3) The proposed protocol is quicker than the one with asymmetric cryptography in completing the bind update. This lower delay helps the MN to complete handover quicker. Furthermore, relatively light computations in the proposed protocol extend battery lifetime of mobile devices.

Greg and Michael [11] proposed another secure RO protocol, called the Child-proof Authentication for MIPv6 (CAM), using only a private/public key pair without resorting to certification of public keys. In this approach, the interface identifier of IPv6 addresses is computed from a public key and auxiliary parameters via a cryptographic one-way hash function. The MN uses the corresponding private key to assert address ownership and to sign messages sent from this address without PKI or any other security infrastructure. The binding between the public key and the address at the CN can be verified by recomputing the hash value and by comparing this hash value with the interface identifier. However, the CN cannot confirm return routability to the CoA. Further, the computation load on the MN side is heavy because every BU message requires the MN to generate a signature and the CN to verify it.

The question has been raised of whether private key cryptography is the only approach for a secure BU. Much research has been geared toward developing a secure BU that contains less expensive cryptography. Veigner and Rong [12, 13] proposed a new route optimization protocol for MIPv6

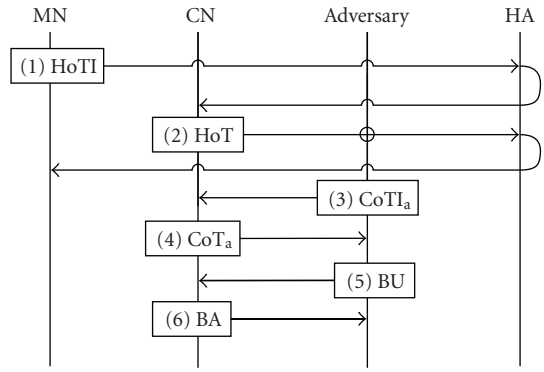

Figure 2: Illustration of a session hijacking attack. Because the (1) and (3) messages are sent independently, the sequence of messages is irrelevant.

(ROM). In their proposal, the MN uses the ROM protocol to assign a unique hash value to its currently used CN. The hash value is sent via the HA-CN path. Simultaneously the home subnet of the MN is authenticated by the CN by means of a three-way handshake. This means that now when it moves into a new subnet, the MN only has to send a BU message directly to the CN. The CN considers the BU message authentic because of the MN's knowledge of the nonce value. This nonce value included in the BU message was previously used when generating the CN's unique hash value. The MN with the paired secret (i.e., a nonce and hash value) first sends the irreversible hash value via an indirect path and has itself authenticated by the CN and then, to assert its ownership of both the HoA and CoA, discloses the nonce value through the direct path. The rather expensive private key cryptography of the approach discussed earlier is replaced by the hash operation. This protocol is similar to our proposed algorithm in its use of a paired secret. Our work complements this earlier work by providing another fully designed routing optimization protocol. However, the main differences between the two protocols include (1) the ROM protocol is not compatible backward with the legacy protocol, and (2) at the end of the ROM protocol, the MN shows the ownership of both the HoA and the CoA addresses but fails to assure the CN that the claimed CoA is routable.

4. Vulnerability of Route Optimization in MIPv6

The goal of a secure RO is to assure the CN that the MN owns the claimed CoA and that this temporary address is reachable in the Internet. Also, the design goal is motivated by the desire to achieve a security level equivalent to that of the IP network without creating major new security problems [14]. Hence, the goal is not to protect against attacks that were already possible before the introduction of IP mobility. Nonetheless, the security protocol in MIPv6 remains vulnerable to a few critical attacks. We discuss the cause and effect of the attacks in further detail in the next section.

4.1. Three Weaknesses in MIPv6. We have found at least three weaknesses in MIPv6. A brief summary of each one is as follows.

(1) The two tokens HT in (2) and CT in (3) make up the secret BU key. These tokens are delivered in clear text. Anyone can easily acquire HT and CT.

(2) Message authentication in the BU is completed after the CN receives the fifth message. Any earlier authentication for HoTI and CoTI is impossible because the MN and the CN do not share a secret key in advance. Hence, the CN must respond to all BU requests. This unconditional response involves an addition to its database, and an adversary may mount a memory overflow attack by sending meaningless BU requests.

(3) The two tokens are created independently of each other. This is because the tokens are created entirely by the CN, and the CoA is new to the CN and has never been used with the associated HoA. The CN is not able to bind the HoA with the CoA at the time of receiving HoTI and CoTI. The CN's ignorance of the association between the CoA and HoA at an early stage makes it almost impossible to generate a pair of related tokens. Because of this independence, the CN checks only to determine if a returning token is the one given by the CN but fails to determine if these two tokens come from a single source or from two different sources. An adversary needs only to manipulate the CoTI and to deceive only the CN to succeed in hijacking a session to a new CoA of the adversary's choice.

4.2. Vulnerability in MIPv6 (Session Hijacking Attack). A session hijacking attack (or redirection attack) is initiated by an adversary located between the HA and the CN. An illustration of this attack flow is depicted in Figure 2. This adversary intercepts the HoT message sent by the CN to the MN, a target victim. This message is in clear text, and the adversary can extract the token from the message (see (2)). This HT token is the first half of the session key for the BU. The adversary sends the forged CoTI message to the CN. An address chosen by the adversary appears as the source address in this message. Let us denote the forged CoTI message and the adversary's address to the CoTI$_a$ and CoA$_a$, respectively. The CN would accept the CoTI$_a$ message because of the second vulnerability described in Section 4.1. The CN generates CT$_a$ and returns this token enclosed in the CoT$_a$ message to the adversary. CoA$_a$ appears as the destination address in the CoT$_a$ message. This CoT$_a$ message is also in clear text, and the adversary acquires the second half of the token necessary to derive the session key. The adversary generates the K_{bm} according to (5) and sends the forged BU message as if it were the legitimate MN updating the new CoA.

The CN extracts the hash indices from the BU messages and reads the two tokens from its hash. Using (5) the CN recovers $K_{bm} = H(\mathrm{HT}\|\mathrm{CT}_a)$ and validates the sign in the BU message. The validation should pass, because the CN's K_{bm} is

the same as the adversary's K_{bm}. The CN accepts the forged BU and starts to communicate with the adversary located at CoA_a. The MN's session thus has been hijacked by the adversary.

This session hijacking attack exploits the third vulnerability discussed in Section 4.1; that is, the two tokens that make up the session key for the BU are created without any common factors between them. This independent key creation lays the foundation for exploitation by the adversary. From the perspective of the adversary, replacing the CoA with CoA_a is quite simple because it is the only thing required in order to send the forged $CoTI_a$ and to remember the CT_a in the CoT_a. It is such a simple attack that the adversary does not need to manipulate HT and the messages associated with HT (i.e., HoTI and HoT). If we could design the BU to have HT and CT share meaningful components known to the CN and to the MN, a session hijack attack would not be so simple. In such a case the change only to the CT is insufficient because the HT and CT_a would then share a common factor different from the one the CN recognizes. Hence, the adversary must forge HoTI and HoT_a and HT_a as well as CT_a for the attack to succeed. Forging HT and those related messages is more difficult than forging the CT. This is because (1) the adversary must be present not only in the CN-MN path but also in the CN-HA path; (2) the adversary must block the HoT_a that is destined for the HoA. The MN would be very suspicious if it found the HoT_a generated as a return of the HoTI that the MN had never sent. However, this blockage by an adversary would be almost impossible without having control of a router or a switch along the CN-HA path, which we believe it is quite difficult. Hence, our design principle for the new BU is to introduce a common factor shared only between the MN and the CN.

5. The Proposed Routing Optimization Protocol

Based upon the foregoing observations, we proposed a novel protocol for a secure RO in the MIPv6. We will discuss protocol requirements first and then the basic protocol proposed in this paper.

5.1. Protocol Requirements. Some requirements were determined in the course of designing the protocol. These requirements were selected after taking into consideration both practical implementation issues and performance issues. Five requirements summarize the most desirable attributes of the new protocol.

(i) Ownership. The MN can corroborate to the CN that the claimed CoA is owned by the MN. Also, the MN should be able to verify the CoA's binding with the MN's original HoA.

(ii) Routability. The CN should be certain that the new CoA is valid and reachable in the network.

(iii) Dependency. In the legacy protocol, the MN is given the session key (K_{bm}) and uses it to authenticate the BU message. This requirement will change how the two tokens are created. These two tokens must rely upon each other and in order to thwart any session hijacking attack and must share a factor that cannot be forged.

(iv) Compatibility and Easy Implementation. The new protocol should be easy to implement and introduce the lesser imperative amendments to the existing MIPv6 protocol so that the transition to the new protocol is smooth and transparent to end users.

(v) No Degradation of QoS. The new protocol should not degrade QoS in the MIPv6, especially the speed of handover.

The first two requirements are essential because they are the security requirements and the main purpose of the BU and of RR, respectively. We show in Section 6.1 how the new protocol satisfies these first two requirements. Satisfaction of the third requirement is discussed in the security analysis of the protocol in Section 6.2. The last two requirements are discussed in Section 6.3 in which we discuss the computational overhead of the protocol.

5.2. The Proposed Protocol. The proposed protocol inherits the strength of the legacy RO protocol in MIPv6 and eliminates the weaknesses identified by ourselves and mentioned in the related work. The advantages of the proposed protocol are concentrated in the design of the BU message. The roles and consequences of the rest of the messages are quite similar to those of the legacy protocol except for minor modification of the messages.

The MN initiates the BU by sending HoTI and CoTI shown in

$$\begin{aligned} \text{HoTI} &= \{\text{HoA}, \text{CN}, R_H, T_1\}, \\ \text{CoTI} &= \{\text{CoA}, \text{CN}, R_C, T_2\}. \end{aligned} \tag{8}$$

R_H and R_C are the random numbers to match, respectively, HoT with HoTI and CoT with CoTI. Without these parameters, mapping HoT to HoTI in the MN would be difficult in a situation such as one in which the MN might send multiple HoTI messages (or CoTI) because of retransmissions. Once the response arrives, the MN is unable to map this response to the multiple HoTI messages. The CN must return this random number in its response to avoid confusion in the MN.

T_1 and T_2 are the tokens generated by the MN in the proposed system. These tokens are shown in

$$S = H(p \| q), \tag{9}$$

$$T_1 = \text{HoA} \oplus S, \qquad T_2 = \text{CoA} \oplus q, \tag{10}$$

where p and q are the quite large random numbers and input values to the one-way hash function $H(\cdot)$. It is believed that finding input values p and q from S in a reasonable time boundary is almost impossible because of the one-wayness of the hash function which is consisted of is also impossible. Note that T_1 and T_2 share the common number q and p in S which is known only to the MN and nobody else.

HoT and CoT are the CN's responses shown in

$$\begin{aligned} \text{HoT} &= \{\text{CN}, \text{HoA}, R_H, \text{HT}_1, i\}, \\ \text{CoT} &= \{\text{CN}, \text{CoA}, R_C, \text{HT}_2, j\}. \end{aligned} \tag{11}$$

These equations are the same as (2) and (3) in the legacy protocol except that the two tokens, HT and CT, are replaced, respectively, by HT_1 and CT_1. We no longer use the session key K_{bm} to authenticate the BU message. HT_1 and CT_1 are instead referred to as cookies in our system and elaborated, respectively, in

$$\begin{aligned} \text{HT}_1 &= N_i \oplus K_{\text{cn}}, \\ \text{CT}_1 &= N_j \oplus K_{\text{cn}}. \end{aligned} \tag{12}$$

N_i and N_j are the two nonce values generated by the CN. These nonce values and two tokens, T_1 and T_2, are saved in the CN's hash under the hash indices of i and j. The indices, i and j, are included, respectively, in HoT and CoT. The CN expects to receive these indices in the next message. In this way, the CN remains stateless, dispensing with the need to remember these parameters.

The binding message is shown in

$$\text{BU} = \left\{\text{CoA}, \text{CN}, \text{HoA}, i, j, \text{LT}, \text{SEQ}, N_i \oplus N_j, p\right\}. \tag{13}$$

N_i and N_j are used with K_{CN} to verify the return routability of CoA by determining whether the MN returns $N_i \oplus N_j$ in the BU message. K_{CN} is the secret key owned by the CN and used to protect N_i and N_j, respectively, in the HoT and CoT messages. The MN should receive both the HoT and CoT messages and extract HT_1 and CT_1. By XORing HT_1 and CT_1 the MN can calculate $N_i \oplus N_j$ and include this in the BU message. Notably, the MN discloses p in this message. The BU message is authenticated with the MN's presentation of its secrets p to the CN.

The CN validates the BU message and then accepts the consequences of the return routability:

$$\text{BA} = \{\text{CN}, \text{CoA}, \text{LT}\}. \tag{14}$$

The CN confirms the BU by sending binding acknowledgment (BA) as shown in (14). CoA appears as the destination address in the BA message.

6. Performance Evaluation

We evaluated diverse aspects of the performance of the protocol. This evaluation includes an analysis to illustrate how the new protocol copes with the vulnerability of the legacy protocol and how it meets the five requirements specified earlier. A comparison of the computational cost between the five protocols is included. The delay involved in completing the secure RO is measured in terms of three popular wireless access networks, and the implications of this delay are described.

6.1. Security Analysis. By using the binding update in the proposed protocol, the MN can assure the CN that the MN is reachable (or routable) at the claimed CoA and that this MN is the owner of the HoA and CoA. The routability and ownership are the two security requirements and we intend to demonstrate that the proposed protocol is securely sound by showing that the proposed protocol satisfies these two requirements.

N_i and N_j are sent in the HoT and CoT messages by the CN and securely wrapped by the CN's secret, $K_{\text{CN}} \cdot N_i$ is directed to HoA along the indirect path, and N_j is directed to CoA along the direct path. In receiving the BU message, the CN retrieves N_i and N_j from its hash using i and j (see (13)) and calculates $N_i \oplus N_j$. The CN checks to see if the returned $N_i \oplus N_j$ is identical to the one calculated. The correct $N_i \oplus N_j$ indicates that the MN is reachable at HoA and CoA in both paths. In other words, the CN can ensure the routability of the return path to the MN.

In this scenario, an adversary impersonating the MN could have intercepted HoT and CoT and calculated $N_i \oplus N_j$ in the same way the MN did. However, the calculations required of the adversary would not be as simple as they might seem. The MN is assigned a new CoA in the foreign network, and this address has never before been associated with the MN's HoT. The adversary would not be able to couple CoT with the corresponding HoT if a fairly large number of BU messages were passing by. This coupling is also difficult for the CN. This is why CN retains K_{CN} unchanged in generating HT_1 and CT_1 and even uses a constant K_{CN} across different binding updates. However, it remains possible, even if it seems quite improbable, for adversaries to couple HT_1 and CT_1. Hence, it is not enough for the CN to assure the RR by presenting $N_i \oplus N_j$ alone. The proposed protocol compensates for this drawback by authenticating the BU message. Because the message is authentic, the content of this message is also authentic.

Using the hash indices i and j, the CN retrieves N_i and T_1 using hash index i and do the same for N_j and T_2 using hash index j. The CN XORs T_1 with the received HoA and compares the output with the hash function of p and q; that is, $\text{HoA} \oplus T_1 = \text{HoA} \oplus \text{HoA} \oplus S = H(p \| q)$. Algorithm 1 elaborates the CN's procedure to validate the BU message. Let us hypothesize that adversaries have intercepted a number of HoTI and CoTI messages in the network and also have been lucky enough to find a pair of T_1 and T_2. Even in this extreme scenario, it is almost impossible for the adversary to find p due to the one-wayness of the hash. No one except the MN that has sent HoTI and CoTI is able to present p to the CN. If the MN presents the right P-value, the CN concludes that this MN also sent HoTI and CoTI, confirming the MN's ownership of the CoA.

HoA and CoA are included in the BU message not only to compute S but also to preclude a dishonest MN from claiming a different CoA in the BU message than the CoA reported in the CoTI message.

6.2. A Suggested Solution for the Three Weaknesses. RO vulnerability is attributable to the three weaknesses discussed

```
Data: index i, j, p, N1 ⊕ N2, HoA, CoA, Hash
Result: Which Verification is confirmed
Begin
   Extract T1, N1', T2, N2' from Table of CN by i and j
   if N1 ⊕ N2 is a N1' ⊕ N2' then          /* return routability is confirmed */
      Compute q' = T2 ⊕ CoA
      X = H(p||q') and H(p||q) = T1 ⊕ HoA
      if H(p||q) is X then                 /* ownership is confirmed */
         return Verification succeeded
      else                                 /* ownership is failed */
         return Verification failed
   else                                    /* return routability is failed */
      return Verification failed
end
```

Algorithm 1: Verification procedure by CN.

in Section 4.1. A solution to any one of these three may remedy the vulnerability in the RO.

The first cause of RO vulnerability lies with delivery of the two tokens in clear text. The remedy requires a shared key to encrypt the tokens as well as authentication and a key exchange protocol for establishing the session key. This additional protocol is a heavy burden for a mobile device.

Delayed authentication causes the CN to accept all HoTI and CoTI messages that request an RO. Early authentication to the MN may be a good solution for this problem. However, following the same reasoning as discussed in the first cause, authentication necessitates a secret key, and we do not consider adding computational overhead to the existing protocol a viable option.

With the complications posed by solutions to the first and second vulnerabilities, we turn to the third of these and suggest another route to closing all three loopholes. The third vulnerability that we discussed originates in the generation by the CN of the two tokens independently of each other. Our solution to this problem is to have the two tokens share a common factor at the time of the generation. In the proposed protocol, q is this common factor. Addition of this feature complicates a session hijacking attack tremendously because an adversary must forge the two tokens and their related message simultaneously, a feat that we believe verges on impossible. In the legacy protocol, embedding a relationship into the two tokens was impossible because they are created by the CN, which has no knowledge of them at the time of their generation. In the proposed protocol, however, the MN generated the two tokens on behalf of the CN without any difficulty in pairing CoA and HoA.

6.3. Computational Comparison. The proposed protocol maintains backward compatibility with the legacy protocol. The new protocol contains six messages, and the role of each message remains the same as in the legacy protocol. The transition to the new protocol is straightforward because this requires only a software upgrade in the kernel.

We compared the computational expenses for the six protocols described in Section 3; CAM [11], the proposed protocol, the legacy protocol [1], ROM [12], CBU [8], and LR-AKE [10]. Because the number of messages to complete the RO is different from protocol to protocol, we compared them in terms of the computational expense in each message. Table 1 shows the computational expense for each message up to the thirteenth message. In order to distinguish operations in MN, CN and HA, cells in the table have different backgrounds.

The proposed protocol, which is only backward compatible with the legacy protocol, comprises the six messages. The ROM protocol is also composed of six messages, but nonetheless is incompatible with the legacy protocol. In order to form the BU message (see the fifth message in Table 1), the legacy protocol uses one 768-bit HMAC and one 128-bit SHA-1, respectively, to compute K_{bm} (see (5)) and to sign the BU message (see (6)). The MN in the proposed protocol computes the one XOR operation for the same message. In order to complete the BU (see the fifth and sixth messages in Table 1), the legacy protocol, the proposed protocol, and ROM, respectively, use five HMAC-SHA-1 operations and two SHA-1 operations, two XOR operations and one hash operation, and one hash operation. CAM is composed of two messages and the most efficient in terms of the number of messages. In contrast LR-AKE has the greatest number of messages. Operations to form each message are quite diverse from one protocol to another, ranging from simple XOR to expensive asymmetric decryption.

Figures 3 and 4 show the computational delays of the six protocols in completing the RO. The delay taken by the each operation as shown in Table 1 is modeled by its average value. The delays of operations done by the three nodes are summed together and plotted in Figures 3 and 4. (LR-AKE requires two HAs for MN and CN, resp. We did not differentiate these two HAs in the computation.) Some of the protocols show different delay measurements, depending upon whether it is the first handover or the second or later handovers. Although Figure 3 depicts the computational delay for the first handover, Figure 4 shows the delay for later handovers. In a continuing sense, the compilation in Table 1 bases RO security in terms of the first handover. CBU and LR-AKE are protocols that fit this definition, and the delay

Table 1: Computational expenses to form each message. The table shows the comparison for up to 13 messages. Although CAM needs only two messages, LR-AKE requires 13 messages to complete the RO. Note that cells in the table have different backgrounds to distinguish nodes these operations are computed. (MU: multiplication, SU: subtract, XR: XOR, MO: modulo, DV: division, EX: exponentiation, HS: one-way hash function, HM: keyed-hash for message authentication, E_S: symmetric encryption, D_S: symmetric decryption, E_{PU}: asymmetric encryption, D_{PR}: asymmetric decryption, SG: signature generation using private key, SV: signature verification using public key.)

	1	2	3	4	5
CAM	SG	HS + SV			
Our	HS + XR	XR	XR	XR	XR
Legacy	—	—	HM	HM	HM + HS
ROM	AD + HS	SU	AD	SU	—
CBU	—	HS	HS	HS + EX + SG	SV + EX + 2HS
LR-AKE	XR + HS + EX + MU + MO	2HS + DV + EX + MO	3HS + XR	E_{PU} + XR + 2HS	$2D_{PR} + E_{PU}$ + 2HS

	6	7	8	9	10	11	12	13
CAM								
Our	HS + XR							
Legacy	4HM + HS							
ROM	HS							
CBU	2HS + EX	E_S	D_S					
LR-AKE	E_S	$D_S + E_S$	$D_S + E_S$	$D_S + E_S$ + HS	$D_S + E_S$	$D_S + E_S$	E_S + HS	$D_S + E_S$

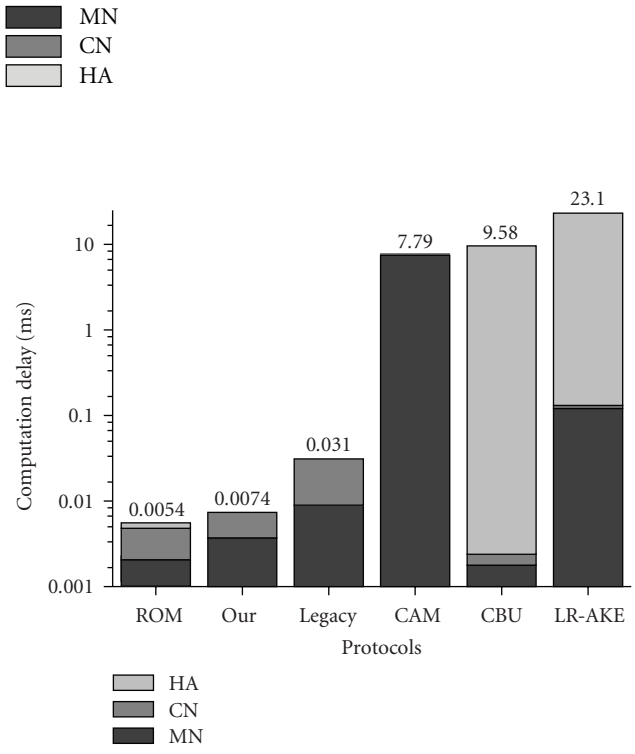

Figure 3: Computational delay for the first handover.

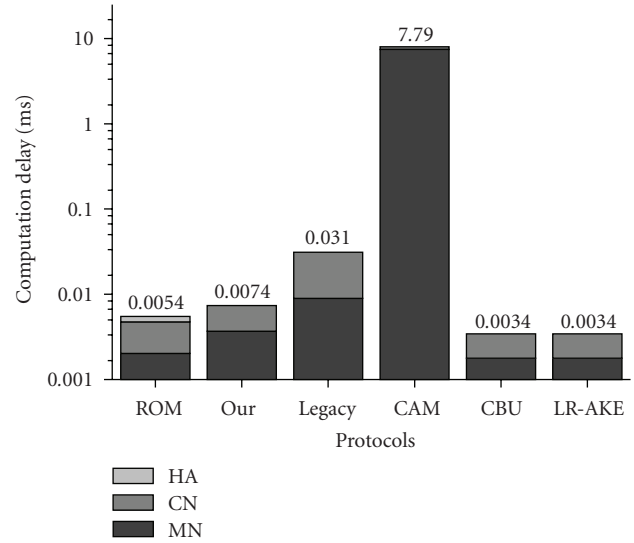

Figure 4: Computational delay for the second and later handovers.

difference between the first and later handovers is quite substantial. These two protocols use private key cryptography to establish a session key at the first handover. This approach to the session key takes considerable time, as shown in Figure 3. After the second handover, the MN and CN encrypt and decrypt messages using symmetric cryptography. The proposed protocol is the fastest in the first handover while CBU and LR-AKE are the fastest in the second and later handovers.

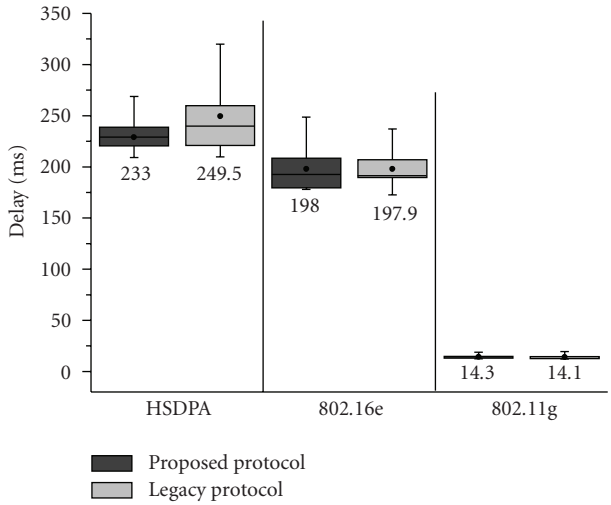

Figure 5: Delays to complete RO in three popular wireless access technologies. We repeated RO for each protocol one thousand times and plotted the outcome in a boxplot.

The delay with the legacy protocol is almost more than four times longer than with the new protocol. The speed of the new protocol is attributed to the transition from frequent hash operations in the legacy protocol to XOR and few hash operations in the new protocol. The delay of the proposed protocol outperforms the ROM protocol by 2 microseconds. Although the difference is insignificant the ROM cannot guarantee return routability to the CN. The computational delay in CAM is quite interesting. It uses an asymmetric signature for the first message in the MN and turns to a one-way hash function and signature verification for the second message in the CN. Although only two messages are used in CAM to complete a secure RO, the computational delay is quite long because of the computation load.

We have implemented the legacy and proposed protocols in three popular wireless access technologies; High Speed Downlink Packet Access (HSDPA), 802.16e [15], and 802.11g [16], illustrated in Figure 5. This is not to compare the performance of these protocols but rather to measure actual delays in order to determine whether it is appropriate to suggest deployment of these protocols in the real environment. This measurement is especially important to developers and engineers in the mobile industry because a delay in the handover greatly influences QoS in mobile applications. The handover in 802.11g completes a secure RO in 14 milliseconds, which is the shortest among the three protocols. About 10 Mbps is the measured data rate of 802.11g and is greater than the 1.3 Mbps of HSDPA and the 3.6 Mbps of 802.16e. Table 2 shows the maximum data rates of the three technologies in terms of measurement and specification. The delay in HSDPA and 802.16e takes longer than 200 milliseconds, which is not appropriate for real-time applications such as IP telephony. The RO in 802.16g is faster than the one in HSDPA because of a higher data rate. We expect Long Term Evolution (LTE) and 802.16m, which are the next versions of HSDPA and 802.16e, respectively, within the next year or so [17]. These new technologies will boost the data rate in the access network to 30 Mbps. Then, those delay-sensitive real-time applications should not have any problems running on these access technologies.

Table 2: Maximum data rates for three technologies in measurement and specification.

	Maximum data rates in measurement (DL/UL)	Maximum data rate in specification (DL/UL)
HSDPA	1.3 Mbps/66 Kbps	14.4 Mbps/2 Mbps
802.16e	3.6 Mbps/423 Kbps	46 Mbps/4 Mbps
802.11g	10.3 Mbps/9.4 Mbps	54 Mbps

7. Conclusion

The two special routines in the secure RO are BU and RR, and the purposes of these routines are to show to the CN that the claimed CoA is a temporary address of the MN and is reachable in the network.

The legacy RO in MIPv6 has a critical vulnerability that could let an adversary hijack an ongoing session to a location chosen by the adversary. This vulnerability is attributed to three weaknesses we found in the RO. The worst weakness is that the two tokens that compose the session key do not share a common factor. This weakness allows an adversary to manipulate CoTI alone, in order to initiate a session hijacking attack. We have proposed a secure RO protocol. This protocol requires only a light computational load and is compatible with the legacy protocol. Most important, this protocol provides a secure BU and RR.

To illustrate its practicality we compared the cost of establishing a secure RO with the proposed protocol with five other protocols that propose to create a secure RO. In addition, we have implemented the proposed and the legacy protocols to measure the communication delay in their use with three wireless access technologies. The evaluation results show that the proposed protocol performs well in terms of low computational cost and minimal delay.

References

[1] D. Johnson, C. Perkins, and J. Arkko, "Mobility support in IPv6," RFC 3775, June 2004.

[2] C. Perken, "IP Mobility Support," RFC 2002, October 1996.

[3] T. Aura, "Mobile IPv6 security," in *Security Protocols*, pp. 3–13, 2004.

[4] K. Elgoarany and M. Eltoweissy, "Security in mobile IPv6: a survey," *Information Security Technical Report*, vol. 12, no. 1, pp. 32–43, 2007.

[5] J.-C. Chen, M.-C. Jiang, and Y. I.-W. Liu, "Wireless LAN security and IEEE 802.11l," *IEEE Wireless Communications*, vol. 12, no. 1, pp. 27–36, 2005.

[6] D. Johnston and J. Walker, "Overview of IEEE 802.16 security," *IEEE Security and Privacy*, vol. 2, no. 3, pp. 40–48, 2004.

[7] G. M. Koien, "An introduction to access security in UMTS," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 8–18, 2004.

[8] R. H. Deng, J. Zhou, and F. Bao, "Defending against redirect attacks in mobile IP," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02)*, pp. 59–67, Washington, DC, USA, 2002.

[9] K. Ren, W. Lou, K. Zeng, F. Bao, J. Zhou, and R. H. Deng, "Routing optimization security in mobile IPv6," *Computer Networks*, vol. 50, no. 13, pp. 2401–2419, 2006.

[10] H. Fathi, S. Shin, K. Kobara, S. S. Chakraborty, H. Imai, and R. Prasad, "Leakage-resilient security architecture for mobile IPv6 in wireless overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 11, pp. 2182–2192, 2005.

[11] O. S. Greg and R. Michael, "Child-proof authentication for MIPv6 (CAM)," *ACM SIGCOMM Computer Communication Review*, vol. 31, pp. 4–8, 1984.

[12] C. Veigner and C. Rong, "A new route optimization protocol for Mobile IPv6 (ROM)," in *Proceedings of the International Computer Symposium*, Taipei, Taiwan, 2004.

[13] C. Veigner and C. Rong, "Flooding attack on the binding cache in mobile IPv6," 2007.

[14] P. Nikander, J. Arkko, T. Aura, and G. Montenegro, "Mobile IP version 6 (MIPv6) route optimization security design," in *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC '03)*, vol. 3, pp. 2004–2008, Orlando, Fla, USA, 2003.

[15] N. Johnston and H. Aghvami, "Comparing WiMAX and HSPA—a guide to the technology," *BT Technology Journal*, vol. 25, no. 2, pp. 191–199, 2007.

[16] D. Vassis, G. Kormentzas, A. Rouskas, and I. Maglogiannis, "The IEEE 802.11g standard for high data rate WLANs," *IEEE Network*, vol. 19, no. 3, pp. 21–26, 2005.

[17] S. Ortiz Jr., "4G wireless begins to take shape," *Computer*, vol. 40, no. 11, pp. 18–21, 2007.