INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2010; **23**:1382–1404
Published online 31 March 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.1113

caTBUA: Context-aware ticket-based binding update authentication protocol for trust-enabled mobile networks

Ilsun You[1], Jong-Hyouk Lee[2] and Bonam Kim[3,*,†]

[1]*School of Information Science, Korean Bible University, Seoul, Korea*
[2]*IMARA Team, INRIA, Rocquencourt, France*
[3]*School of Electrical and Computer Engineering, Chungbuk National University, Cheongju, Korea*

SUMMARY

Existing binding update (BU) authentication protocols do not consider context information, such as trust, location, and current time, when verifying a mobile node's care-of address (CoA). Instead, the correspondent node executes its own CoA validation in spite of facing a highly trusted situation or simply bypasses the CoA validation, making it difficult to maintain a reasonable trade-off between security and efficiency. This paper applies the context-aware concept to the BU process and proposes a new context-aware ticket-based binding update authentication (caTBUA) protocol. The proposed protocol dynamically performs an appropriate CoA validation based on the context information to achieve a good balance between security and efficiency. Utilizing numerical analysis to compare the performance of the proposed protocol to that of existing authentication protocols in terms of authentication cost and authentication message transmission latency confirmed that the proposed caTBUA protocol yields a better performance than the existing BU authentication protocols. Copyright © 2010 John Wiley & Sons, Ltd.

Received 31 March 2009; Revised 9 October 2009; Accepted 8 December 2009

KEY WORDS: MIPv6; binding update; context-aware; CoA validation; security

1. INTRODUCTION

With the advent of mobile computing environments, mobile devices have become a routine part of our everyday lives. Today's mobile network environments are designed to satisfy the various requirements of the mobile devices connected to the network seamlessly and effortlessly at all times wherever the user happens to be. Mobile IPv6 (MIPv6), specified by the IETF, is one of the protocols that makes this possible [1].

MIPv6 is a mobility support protocol that enables nodes to stay reachable regardless of their movements and locations in IPv6-based networks. In order to achieve mobility and reachability, this

[*] Correspondence to: Bonam Kim, School of Electrical and Computer Engineering, Chungbuk National University, Cheongju, Korea.
[†] E-mail: kimbona@chungbuk.ac.kr

protocol assigns two addresses to each mobile node (MN): its home address (HoA) and its care-of address (CoA). Each MN belongs to a home network where it is always identified by its HoA, the permanent address allocated by its home network. When the MN visits a foreign network, it is issued with a CoA temporary address in the foreign network. The relationship between the MN's HoA and CoA is referred to as its binding; whenever the MN changes its location, it must not only notify its home agent (HA), a router in the MN's home network, but also the correspondent node (CN) in the new network, the MN's peer node, and provide both with its new binding information via the binding update (BU) process.

MIPv6 provides two possible modes for communications between the MN and the CN. In the first, known as bidirectional tunneling, an HA is deployed as a proxy for the MN to relay packets between the MN and the CN. However, this mode requires an inefficient triangle routing path, which involves a BU process between the MN and the CN. In the second mode, route optimization (RO), the packets received from the CN can be routed directly to the MN's CoA, thus eliminating the overhead that results from tunneling via the HA. Before initiating this mode, the MN must register its current binding for both the HA and the CN by performing the BU processes.

Unlike the MN-HA path, which is protected by IPsec, the MN-CN path is insecure when dealing with the BU process between the MN and the CN, exposing the involved nodes to various security threats. In order to protect the BU process, the IETF provides the return routability (RR) method [1], where the CN verifies the MN's HoA and CoA while sharing secret information with the MN. In spite of its advantages, however, this method results in both performance and security problems [2–4]. Numerous alternative approaches to the RR method have been proposed based on the use of public key cryptography [2–13]. These use their own public key method to enable the MN and the CN to share a strong secret, the lifetime of which is sufficiently long to minimize the amount of signaling messages and handover latency.

These methods have all tried to secure the BU process between two previously unknown nodes on the assumption that no global security infrastructure is available. However, this assumption may not be true for trust-enabled network environments, which can establish trust among their own nodes. Thus, a more efficient method would be helpful for this kind of network environment. Recently, a ticket-based binding update authentication (TBUA) protocol has been proposed for trust-enabled network environments where involved nodes can have faith in each other [14]. The TBUA protocol employs an HA as a ticket issue server that issues tickets based on the pre-established levels of trust. Such an employment requires the CN to maintain a trust relationship with the HA instead of the MN, thus reducing the management cost of the CN. In addition, the TBUA protocol adopts the early binding update (EBU) and credit-based authorization (CBA) techniques to optimize the CoA test. Consequently, the protocol is efficient in terms of both its management cost and security.

However, the TBUA protocol, like previous BU security protocols, always verifies the MN's CoA in a fixed way that does not take into account the MN's context, such as its trust, location, or the current time. In such fixed approaches, the CN executes its own CoA validation, for example the CoA test or the Parallel CoA test, independent of whether it is in a highly trusted situation where it would be safe to bypass the CoA validation or when it is an unsafe situation where it would not be safe to do so. Thus, it is difficult to maintain a reasonable trade-off between security and efficiency. To address this issue, this study developed a new method that balances security and efficiency by incorporating context-aware concepts into the BU process in MIPv6-based networks.

In this paper, we propose a context-aware ticket-based binding update authentication (caTBUA) protocol. The new protocol extends the TBUA and includes considerations of the MN's context when conducting the BU process. The proposed caTBUA protocol dynamically performs the CoA validation based on the MN's context while enjoying every benefit of the original TBUA protocol. The CN only bypasses the CoA validation if the MN is able to prove its trust status to the CN during the BU process. Consequently, the new protocol offers a significant improvement in both security and efficiency compared with other protocols that do not consider the MN's context. A rigorous mathematical analysis comparing the new protocol with existing models reveals that security and efficiency for the BU process can be optimized concurrently if the CN selects the appropriate CoA validation depending on the MN's context.

The remainder of this paper is organized as follows. In Section 2, the existing BU authentication protocols for trust-enabled mobile network environments are reviewed. Section 3 presents the proposed caTBUA protocol, including its motivation, algorithm, and procedures. The analytical models for the performance analysis are described in Section 4, and Section 5 examines the results of the mathematical analysis and comparison. The paper concludes by summarizing the study in Section 6.

2. RELATED WORKS

Before beginning the RO process in MIPv6, an MN performs a BU process by sending a BU message to its CN, which then responds with a binding acknowledgment (BA) message. The fundamental requirement for securing the BU process is that the CN authenticates both the MN and its BU message. Unfortunately, it is very difficult to achieve strong authentication between two previously unknown nodes (MN and CN) where no global security infrastructure is available. Thus, the need has arisen for a security solution to enable sufficient authentication between the CN and the MN without traditional secret or public key-based authentication infrastructures.

Several studies have been conducted to address this security issue. At present, the IETF accepts the RR protocol as the standard for a secure BU process [1]. Besides the RR protocol, various approaches have been proposed based on public key cryptography [2–13] to avoid the need for additional security infrastructures, which generally attempt to associate the MN's HoA with its public key through techniques, such as address-based keys (ABKs) [13], cryptographically generated address (CGA) [15], and purpose-built keys (PBK) [16]. Recently, in order to both improve the security and address the inefficiency problems caused by the RR method, the Optimized Mobile IPv6 (OMIPv6) series have been examined by the network working group in the IETF [3–7]. Like other public key-based approaches, the OMIPv6 series use their own public key techniques to construct a strong secret that is shared between the MN and the CN while at the same time optimizing the RR protocol.

All these methods have attempted to construct a secure BU process between two previously unknown nodes on the assumption that no global security infrastructure is available. Thus, they require no configuration and no trusted entities except for the MN's HA. However, this assumption is not trusted for a network domain where involved nodes enjoy pre-established trust relationships with each other, allowing more efficient methods based on this pre-established trust to be applied. For such cases, the IETF has introduced the static shared key (SSK) protocol, which requires the configuration of a shared secret between the MN and its CN [17]. However, the SSK protocol is vulnerable to the redirection-based flooding and reply attacks, leading to high management costs for the CN. The TBUA protocol [14] avoids such problems by implementing a ticket-based key distribution and a parallel CoA test. This will be described in more detail below after a review of the RR and SSK protocols.

2.1. The RR protocol

The RR protocol enables the CN to verify that the MN is indeed reachable at its claimed CoA, as well as at its HoA. In addition, it allows the two nodes to establish a shared secret, which is then used to authenticate the BU and BA messages.

Figure 1 illustrates the procedure used by the RR protocol, which is composed of the Home Test Init (HoTI), Care-of Test Init (CoTI), Home Test (HoT), and Care-of Test (CoT) messages. Whereas the HoTI and HoT messages are relayed via the HA, the CoTI and CoT messages are exchanged directly between the MN and the CN. In order to start this protocol, the MN sends the HoTI and CoTI messages to its CN simultaneously. In response, the CN transmits the HoT and CoT messages, which include the keygen tokens *Thk* and *Tck*, to the MN. By hashing the tokens together, the MN builds a binding management key *Kbm*, which allows the CN to verify that the MN is addressable at its HoA and CoA can be used to protect the subsequent BU process between the MN and the CN. Once the RR process is complete, the MN executes the binding process by exchanging BU and BA messages with the CN.

Although this method satisfies the security requirements for the RO in MIPv6, it leads to the following problems [1–4]:

- For security reasons, the *Kbm*'s lifetime is limited to a maximum of 420 s. Consequently, *Kbm* must be updated frequently, increasing both the number of mobility signaling messages and the handover latency.
- The method cannot protect messages on either the MN-CN path or the HA-CN path. This vulnerability exposes the RO mode to various security threats every few minutes during a lengthy session.

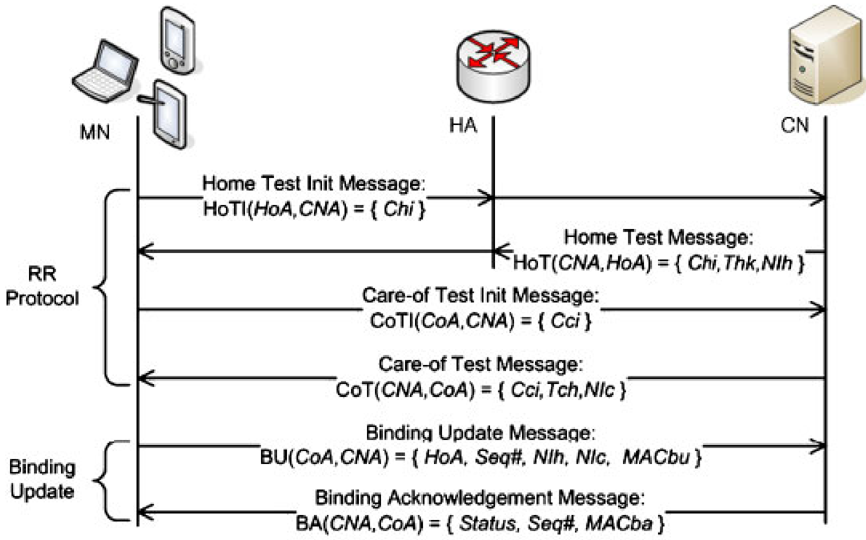
2.2. The SSK protocol

Recently, the IETF proposed the use of the SSK protocol for network environments where each MN can establish trust with its CNs [17]. In particular, this protocol is highly suitable for the case where MNs and CNs are administered within the same domain. As shown in Figure 2, in the SSK protocol, the MN and its CN previously shared the key materials, such as *Kcn*, nonces, and nonce indexes that are used to generate *Kbm*. Through these pre-configured key materials, this protocol can omit the need for signaling messages about the routability tests, thus minimizing the handover latency and the number of signaling messages.

Although this protocol achieves good efficiency, it suffers from the following problems:

- Each CN incurs the additional cost of previously configuring and maintaining the key materials for its MNs, which can be severe in environments where every CN can be an MN.
- The elimination of the routability tests leaves this method vulnerable to redirection-based flooding (RBF) attacks [2] launched maliciously by a legitimate MN.
- This method depends on the use of sequence numbers to prevent reply attacks. When the sequence number rolls over, the involved nodes must configure new key materials.

DOI: 10.1002/dac


MN
HA
CN
Home Test Init Message:
HoTI(*HoA*,*CNA*) = { *Chi* }
Home Test Message:
HoT(*CNA*,*HoA*) = { *Chi*,*Thk*,*NIh* }
RR Protocol
Care-of Test Init Message:
CoTI(*CoA*,*CNA*) = { *Cci* }
Care-of Test Message:
CoT(*CNA*,*CoA*) = { *Cci*,*Tch*,*NIc* }
Binding Update Message:
BU(*CoA*,*CNA*) = { *HoA*, *Seq#*, *NIh*, *NIc*, *MACbu* }
Binding Update
Binding Acknowledgement Message:
BA(*CNA*,*CoA*) = { *Status*, *Seq#*, *MACba* }

- SA: source Address, DA: destination Address
- Msg(SA,DA): It means that the message Msg is sent from SA to DA
- *Chi*: home init cookie, *Cci*: care-of init cookie
- *CNA*: The IPv6 address of the correspondent node.
- *Kcn*: The CN's secret key
- *NIh*: home nonce index, *NIc*: care-of nonce index
- *Nh*: home nonce , *Nc*: care-of nonce
- *Thk*: home keygen token = *First (64, HMAC_SHA1 (Kcn, (HoA | Nh | 0))*
- *Tck*: care-of keygen token = *First (64, HMAC_SHA1 (Kcn, (CoA | Nc | 1))*
- *Kbm*: *SHA1 (Thk | Tch)*
- *Seq#*: Sequence number of the binding update message
- *MACbu*: *First (96, HMAC_SHA1 (Kbm, CoA | CNA |* BU*)*
- *MACba*: *First (96, HMAC_SHA1 (Kbm, CoA | CNA |* BA*)*
- *Status*: This value indicates the disposition of the binding update

Figure 1. The procedure of RR protocol.

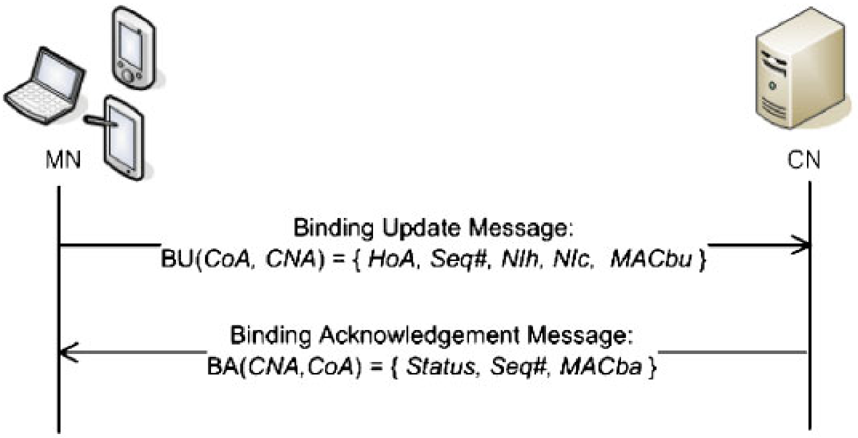
2.3. The TBUA protocol

The TBUA protocol improves the SSK protocol by employing an HA as a ticket issue server so that the HA pre-shares a secret key with each CN. Using a pre-shared key enables the HA to securely distribute *Kbmperm*, a long-term key for binding management, between its MN and CN and makes it possible for each MN to launch a BU process with CNs that have established trust relationships with its own HA. Thus, by utilizing a ticket issue server each CN can eliminate the cost of pre-configuring and maintaining the key materials for its MNs. This method prevents RBF attacks by using the CoA test, which results in one additional round trip time (RTT) delay. In order to optimize the CoA test, it adopts the EBU and credit-based authorization (CBA) techniques [14].

Figure 3 shows the procedure followed by the TBUA protocol, which is divided into three phases: ticket issue, EBU, and complete binding update (CBU).

- **Ticket issue phase**: In this phase, when requested by the MN, the HA generates *Kbmperm*, a long-term key for BUs, and issues a ticket that includes the generated key in encrypted form. The MN uses both the long-term key and the ticket to perform BUs with the CN. In order to

DOI: 10.1002/dac


MN
CN
Binding Update Message:
BU(*CoA, CNA*) = { *HoA, Seq#, NIh, NIc, MACbu* }
Binding Acknowledgement Message:
BA(*CNA,CoA*) = { *Status, Seq#, MACba* }

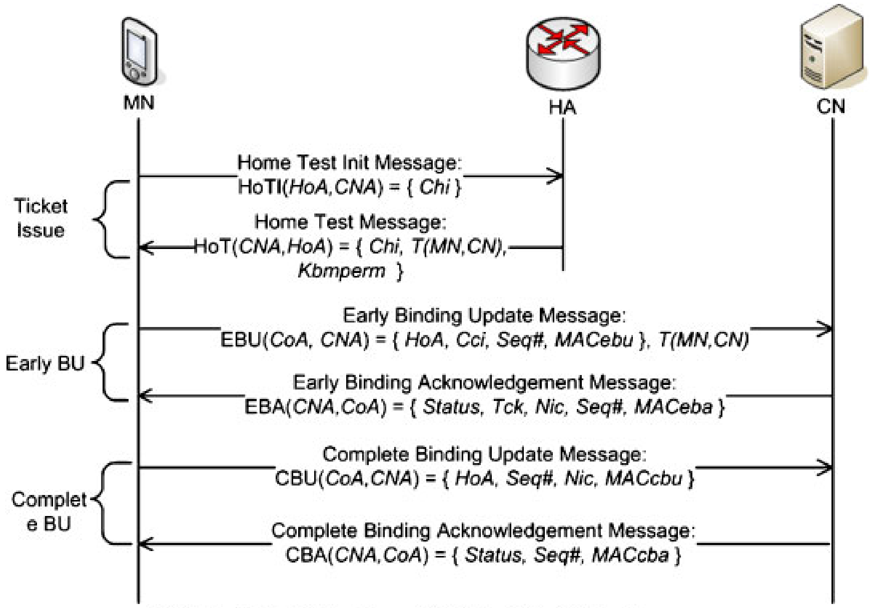
- *Kcn*: The CN`s secret key
- *NIh*: home nonce index, NIc: care-of nonce index
- *Nh*: home nonce, Nc: care-of nonce
- *Kcn, NIh, NIc, Nh* and *Nc* are preshared between a MN and its CN
- *Thk*: home keygen token = *First (64, HMAC_SHA1 (Kcn, (HoA* | *Nh* | *0))*
- *Tck*: care-of keygen token = *First (64, HMAC_SHA1 (Kcn, (CoA* | *Nc* | *1))*
- *Kbm: SHA1 (Thk | Tch)*
- *Seq#*: Sequence number of the binding update message
- *MACbu: First (96, HMAC_SHA1 (Kbm, CoA* | *CNA* | BU*)*
- *MACba: First (96, HMAC_SHA1 (Kbm, CoA* | *CNA* | BA*)*
- *Status*: This value indicates the result of the binding update

Figure 2. The procedure of SSK protocol.

initialize this method, the MN sends the CN a HoTI message, which is forwarded via the HA. When arriving at the MN's home network, the message is intercepted by the HA, which then checks if there is a secret key pre-shared between itself and the CN. If no such key exists, the RR protocol is performed from this point. Otherwise, the HA generates *Kbmperm* and issues a ticket for the MN. As depicted in Figure 3, the ticket is composed of the MN's HoA, the HA's IPv6 address, the CN's IPv6 address, the lifetime, *EKey* and *MACticket*. As *EKey* and *MACticket* are computed through *Khc*, the CN having *Khc* can verify the ticket and retrieve *Kbmperm* from it. Instead of forwarding the HoTI message to the CN, the HA responds to the MN with a HoT message that includes *Kbmperm* and the ticket. Once the MN receives the ticket from the HA, the MN can omit this phase in each subsequent BU process until the ticket expires.

- **EBU phase**: After the first phase, the MN and the CN execute the EBU phase by exchanging the EBU and Early Binding Acknowledgement (EBA) messages. During this phase, the CoA test is applied to prevent the RBF attacks. Especially, the CoA is performed in parallel with the data transmission from and to the MN's new CoA while minimizing the handover delay caused by itself. That is, the MN starts the data transmission immediately after sending the EBU message to the CN whereas the CN starts the data transmission immediately after sending the EBA message to the MN. In order to initiate the EBU phase, the MN sends the

DOI: 10.1002/dac


MN
HA
CN
Home Test Init Message:
HoTI(HoA,CNA) = { Chi }
Ticket Issue
Home Test Message:
HoT(CNA,HoA) = { Chi, T(MN,CN), Kbmperm }
Early Binding Update Message:
EBU(CoA, CNA) = { HoA, Cci, Seq#, MACebu }, T(MN,CN)
Early BU
Early Binding Acknowledgement Message:
EBA(CNA,CoA) = { Status, Tck, Nic, Seq#, MACeba }
Complete Binding Update Message:
CBU(CoA,CNA) = { HoA, Seq#, Nic, MACcbu }
Complet e BU
Complete Binding Acknowledgement Message:
CBA(CNA,CoA) = { Status, Seq#, MACcba }

- *CNA*: the CN's IPv6 address, *HAA*: the HA's IPv6 address
- *Chi*: home init cookie, *Cci*: care-of init cookie
- *Kbmperm*: a long-term key for binding management
- *Kcn*: The CN's secret key
- *NIc*: care-of nonce index
- *Nc*: care-of nonce
- *Tck*: care-of keygen token = *First (64, HMAC_SHA1 (Kcn, (CoA | Nc | 1))*
- *Kbm: SHA1 (Tck | Kbmperm)*
- *MACebu: First (96, HMAC_SHA1 (Kbmperm, CoA | CNA |* EBU*)*
- *MACeba: First (96, HMAC_SHA1 (Kbmperm, CoA | CNA |* EBA*)*
- *MACcbu: First (96, HMAC_SHA1 (Kbm, CoA | CNA |* CBU*)*
- *MACcba: First (96, HMAC_SHA1 (Kbm, CoA | CNA |* CBA*)*
- *T(MN,CN) = HoA|HAA|CNA|LT|Ekey|MACticket*
- *HoA*: the MN's home address
- *LT*: the period during which the ticket is valid
- *Khc*: a secret key shared between the HA and the CN
- *Ekey = E_{Khc}(Kbmperm)*
- *MACticket = HMAC(Khc, HoA|HAA|CNA|LT|Ekey)*

Figure 3. The procedure of TBUA protocol.

CN the EBU message and its own ticket. When receiving them, the CN uses *Khc*, a secret key pre-shared between itself and the HA, to verify the ticket. If the verification is successful, the CN decrypts *EKey* with *Khc* to retrieve *Kbmperm*, which is then used to check if the EBU message is valid. In the case of the valid EBU message, the CN not only learns the MN's new CoA but also believes that the MN is the legitimate owner of the HoA. While starting to use the new CoA from this time, the CN concludes this phase by sending the MN the EBA, including *Tck*, a care-of keygen token.

- **CBU phase**: Even after the second phase is completed, although it knows the MN's new CoA, the CN still cannot be sure that the MN is actually present at the new address. Thus, the MN must prove that it is really reachable at its claimed CoA by performing a CBU phase. In order to start this phase, the MN sends the CN a CBU message, which can be authenticated through *MACcbu* computed with *Kbm*. Because *Kbm* is derived from *Tck* in addition to *Kbmperm*, a valid *MACcbu* assures the CN that the MN has received the EBA message at its claimed CoA. Thus, if the CBU message is verified successfully, the CN believes the MN's presence at the new CoA and it concludes this phase by responding to the MN with a CBA message. As mentioned above, during the second-phase data transmission the MN's CoA is not verified, which makes this approach vulnerable to the misuse of unverified CoAs. To solve this security problem, the CBA technique is adopted [6, 14] to limit the amount of data transmission until the CBU phase finishes. Consequently, if the amount of data transmitted exceeds a specified value, the RO mode is postponed until the CBU message is verified successfully.

3. CONTEXT-AWARE TICKET-BASED BINDING UPDATE AUTHENTICATION

3.1. Motivation

As shown in Figure 4, a malicious MN, i.e. attacker, can launch attacks by sending its CNs the false BU messages indicating its handover to a CoA owned by the victim node during the BU process. Such attacks flood the victim node and their networks with data packets sent from CNs and are known as RBF attacks. In order to defend against such RBF attacks, it is vital to verify if the malicious MN is really present at its claimed CoA.

As a CoA validation method, the CoA test or the parallel CoA test is mainly employed for securing the BU process [1–5]. However, both such tests sacrifice efficiency to gain powerful security.

- **CoA test**: In this method, which is adopted by the RR protocol, the CoA validation is performed in such a way that the CN sends the CoTI message to the claimed CoA and receives the CoT message from it. This test is simple to implement and strictly validates the CoA by introducing one additional RTT latency.
- **Parallel CoA test**: This method, which is adopted by the Enhanced Route Optimization (ERO) and TBUA protocols, allows the MN and the CN to conduct both the BU and the

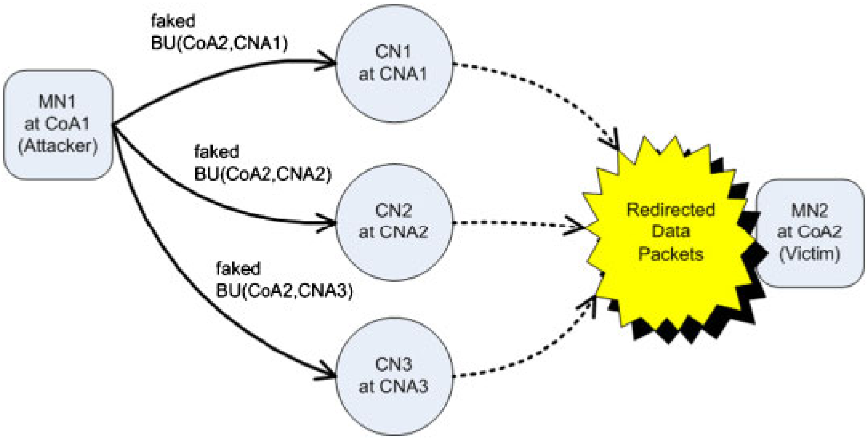


Figure 4. The redirection-based flooding attack.

DOI: 10.1002/dac

CoA test in parallel. It therefore sacrifices security to reduce 1 RTT latency. Here, the BU is divided into two phases: EBU and CBU. First, the MN and CN conduct the EBU based on the pre-shared secret (*Kbmperm*) in addition to the CoA test. During this phase, the data packets start to be transferred between the MN and CN without validation about the claimed CoA. However, the number of packets is restricted based on the CBA scheme to prevent an RBF attack. Second, the MN and the CN exchange the CBU and acknowledge messages protected by the binding management key derived from the pre-shared secret and the care-of-keygen token. If this CBU is successful, the MN is then permitted to receive data packets from the CN without any restriction. However, this method finds it difficult to identify RBF attacks within limited conditions.

Existing BU security protocols, including the TBUA protocol, always perform a fixed CoA validation without considering the MN's context, such as trust, location, current time, and so forth. In such fixed approaches, the CN must execute its own CoA validation (CoA test or parallel CoA test) even in highly trusted situations, or bypass the CoA validation and thus operate in an unsafe situation. Thus, it is difficult for them to maintain a reasonable trade-off between security and efficiency. We therefore incorporated the concept of context-awareness into our new authentication method. Security and efficiency for the BU procedure can be optimized concurrently if the CN can select the appropriate CoA validation based on the individual MN's context. For example, the CN can skip the CoA validation when it believes that the MN is in a highly trusted situation. Note that the CN can evaluate the level of trust for the MN according to the context. Consequently, the caTBUA protocol, which improves the TBUA protocol to perform the context-aware CoA validation, can lead to a notable improvement in security and efficiency compared with the existing fixed CoA validation approaches. In the caTBUA protocol, the CN is flexible enough to select its CoA validation according to the context of each MN. We also propose the use of the limited CoA method, which forces the MN to generate its own new CoA based on its previous CoA and its history of its visiting the network. This makes it impossible for a malicious MN to intentionally select the CoA of the victim node as its own new CoA for the RBF attack. Thus, this method prevents an RBF attack from being launched by the particular MN without incurring additional latency. However, it still introduces some vulnerabilities to network basis attacks.

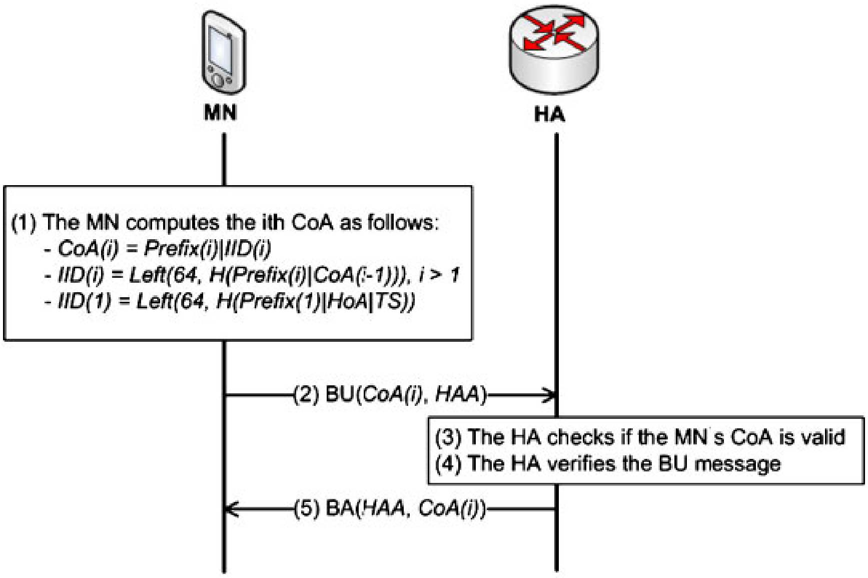
The proposed caTBUA protocol extends the previously proposed TBUA protocol, so that the following basic rules are taken into consideration when constructing the caTBUA protocol.

- The MN generates a new CoA based on the previous CoA and the prefix of its current visiting network, after which the CN and the HA verify if the MN's CoA follows the address generation rule.
- When the CN receives the EBU message during the BU process, the appropriate CoA validation is conducted depending on each MN's context in view of the network trust achieved so far.

3.2. The limited CoA method

In our proposed protocol, every MN generates its new CoA as illustrated in Figure 5. The HA verifies the MN's newly generated CoA whenever the MN executes the home registration. The CN also verifies the newly generated CoA whenever the MN executes the CN registration. For the first CN registration, where the CN cannot verify the MN's CoA, the HA adds the MN's current CoA to the ticket after validating the address. Thus, the MN's ticket allows the CN to check whether the MN's CoA is correctly generated at the first CoA registration. In this manner, the limited CoA method can prevent any attempted to the RBF attack intended for the particular MN. However,

DOI: 10.1002/dac



(1) The MN computes the ith CoA as follows:
- *CoA(i) = Prefix(i)|IID(i)*
- *IID(i) = Left(64, H(Prefix(i)|CoA(i-1))), i > 1*
- *IID(1) = Left(64, H(Prefix(1)|HoA|TS))*

(3) The HA checks if the MN's CoA is valid
(4) The HA verifies the BU message

- *CoA(i)* is the CoA generated for the i_{th} time
- *Prefix(i)* shows the prefix information on its i_{th}
- *TS* denotes a time stamp
- *HAA* denotes the IPv6 address of the HA

Figure 5. The home registration.

this method cannot defend against network basis RBF attacks where a legitimate MN generates a faked CoA using the prefix of the target network. Therefore, it is desirable for the limited CoA method to be used in conjunction with the CoA test or the parallel CoA test.

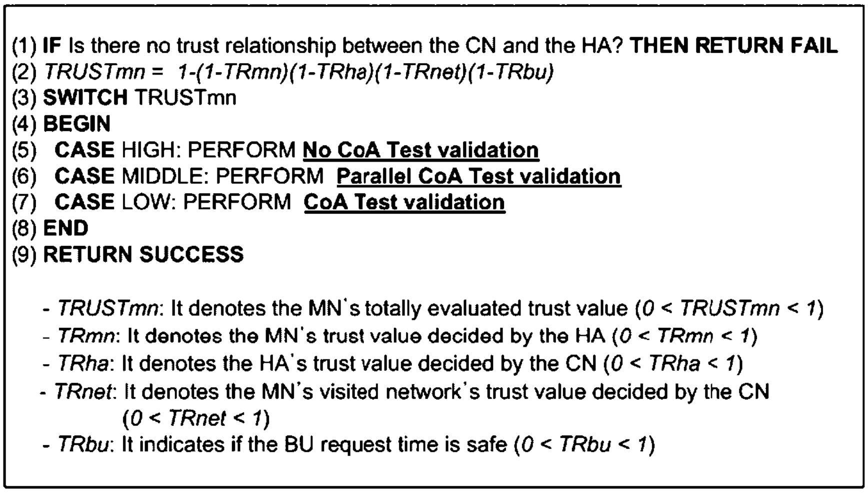
3.3. *The algorithm for the caTBUA protocol*

This subsection describes an algorithm for the proposed caTBUA protocol. The main context information about the MN can be summarized as follows:

- **MN's trust**: This indicates the MN's trust degree estimated by the HA and is transferred to the CN in the MN's ticket.
- **HA's trust**: This signifies the HA's trust degree and is determined based on the trust relationship previously established between the HA and the CN. For the trust relationship, they should share a secret key for the BU process.
- **Foreign network's trust**: This represents the trust of the network where the MN is currently located and is decided based on the trust relationship previously established between the CN and the access router (AR) in the MN's current visiting network. For the trust relationship, they should share a secret key for the BU process.
- **Requested time**: This is the MN's BU request time.

Figure 6 shows the context-based BU algorithm that dynamically determines the proper CoA validation according to the MN's context information. In the BU process, the CN calculates the

DOI: 10.1002/dac



```
IF Is there no trust relationship between the CN and the HA? THEN RETURN FAIL
$TRUSTmn = 1-(1-TRmn)(1-TRha)(1-TRnet)(1-TRbu)$
SWITCH TRUSTmn
BEGIN
    CASE HIGH: PERFORM No CoA Test validation
    CASE MIDDLE: PERFORM Parallel CoA Test validation
    CASE LOW: PERFORM CoA Test validation
END
RETURN SUCCESS
```

- *TRUSTmn*: It denotes the MN's totally evaluated trust value ($0 < TRUSTmn < 1$)
- *TRmn*: It denotes the MN's trust value decided by the HA ($0 < TRmn < 1$)
- *TRha*: It denotes the HA's trust value decided by the CN ($0 < TRha < 1$)
- *TRnet*: It denotes the MN's visited network's trust value decided by the CN ($0 < TRnet < 1$)
- *TRbu*: It indicates if the BU request time is safe ($0 < TRbu < 1$)

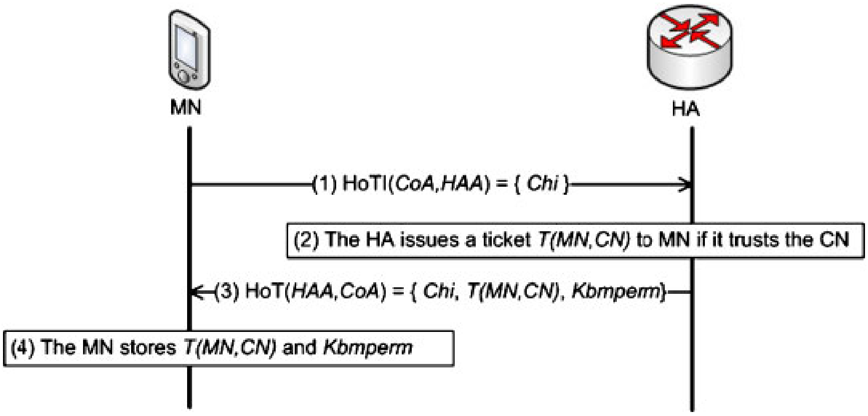
Figure 6. The context-aware BU algorithm.

network trust for the MN (*TRUSTmn*) after the MN transfers the EBU message to the CN. If *TRUSTmn* is high, the CoA validation is skipped. If *TRUSTmn* is intermediate, the parallel CoA test is conducted. Otherwise, the full CoA test is performed. To calculate *TRUSTmn*, each CN decides the values to be used, for example *TRmn*, *TRha*, *TRnet*, *TRbu*, and *Lcoa*, along with its policy.

3.4. *Two phases of caTBUA protocol*

The proposed caTBUA protocol has two phases: ticket issue and context-aware BU phases.

- **Ticket issue phase**: As illustrated in Figure 7, the ticket issue phase is similar to that of the TBUA protocol and is only performed at the first CN registration. The MN exchanges the long-term key, *Kbmperm*, with the CN during this phase, which will thereafter be omitted. Unlike the TBUA protocol, the new protocol lets the MN's ticket $T(MN, CN)$ include the current CoA and the trust (*TRmn*) of the MN. Thus, if the ticket is valid, the CN can believe the MN's current CoA while obtaining *TRmn*, one of the MN's context information values.
- **Context-aware BU phase**: As shown in Figure 8, the MN starts this phase by transferring the EBU message to the CN. If the AR in the MN's visiting network has an existing trust relationship with the CN (in other words, the AR and the CN have already agreed to share a key, *Kac*), it uses *Kac* to compute *MACar*, which is then sent to the CN together with the EBU message. In the case of the first CN registration, the CN should verify the MN's ticket $T(MN, CN)$ before the EBU message. It then extracts *Kbmperm* from the ticket, and verifies the EBU message. If the verification is successful, $T(MN, CN)$ and *Kbmperm* are stored at the CN. Thus, $T(MN, CN)$ is no longer accompanied by the EBU message. Upon receipt of *MACar*, the CN validates it while evaluating the EBU message. Based on the result, the *TRnet* value that is part of the MN's context information is calculated. After confirming the EBU message, the CN checks to see whether the MN's new CoA can be generated by the limited CoA method. Finally, the CN assesses the MN's context information using the suggested algorithm as shown in Figure 6, and concludes an appropriate CoA validation through the


MN
HA
(1) HoTI(CoA,HAA) = { Chi }
(2) The HA issues a ticket T(MN,CN) to MN if it trusts the CN
(3) HoT(HAA,CoA) = { Chi, T(MN,CN), Kbmperm}
(4) The MN stores T(MN,CN) and Kbmperm

- *T(MN,CN)* = *HoA|CoA|HAA|CNA|LT|Ekey|TR|MACticket*
- *HoA*: the MN's home address
- *HAA*: the HA's IPv6 address, CNA: the CN's IPv6 address
- *LT*: the period during which the ticket is valid
- *Khc*: a secret key shared between the HA and the CN
- *Ekey* = *Ekhc(Kbmperm)*
- *TRmn*: MN's trust degree
- *MACticket* = *HMAC(Khc, HoA|CoA|HAA|CNA|LT|Ekey|TRmn)*

Figure 7. The ticket issue phase in the caTBUA.

calculated net trust *TRUSTmn*. If the value of *TRUSTmn* is high, the EBA message has a status value indicating that the CoA validation is not needed. If the value of *TRUSTmn* is intermediate or low, the parallel CoA test or CoA test, respectively, will be performed. At this time the EBA includes *Kck*, the care-of keygen token, for those tests while indicating that the CoA validation is needed.

4. ANALYTICAL MODEL

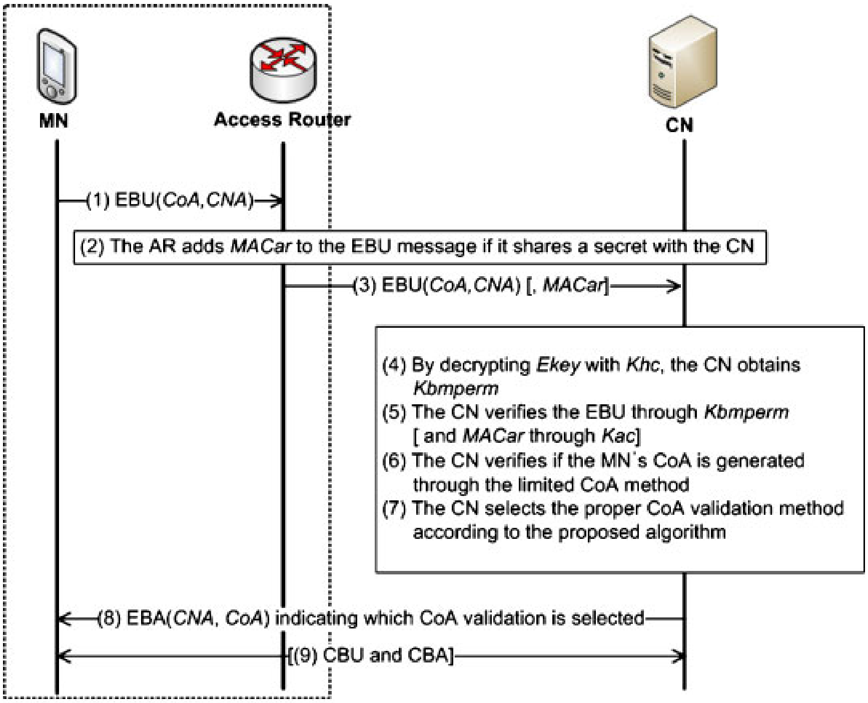
In this section, we provide an analytical model including the mobility and network topology models that are used as our performance analysis environments.

4.1. Mobility model

As our mobility model, we use a probabilistic random walk mobility model, where the MN is assumed to be moving with a particular speed and in a particular direction for a given interval time [18, 19], and adopt the Markov chain with only two states to present the movement probabilities. Let p be the probability that the MN stays within the current network. Then, $1-p$ is the probability that the MN moves to another network. The following transition probability matrix presents the movement probabilities for the MN.

$$p_{i,j} = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}. \tag{1}$$

DOI: 10.1002/dac


MN
Access Router
CN
(1) EBU(*CoA,CNA*)
(3) EBU(*CoA,CNA*) [, *MACar*]
(8) EBA(*CNA, CoA*) indicating which CoA validation is selected
[(9) CBU and CBA]

(2) The AR adds *MACar* to the EBU message if it shares a secret with the CN

(4) By decrypting *Ekey* with *Khc*, the CN obtains *Kbmperm*
(5) The CN verifies the EBU through *Kbmperm* [and *MACar* through *Kac*]
(6) The CN verifies if the MN's CoA is generated through the limited CoA method
(7) The CN selects the proper CoA validation method according to the proposed algorithm

- AR: an access router involved in the foreign network visited by the MN
- *Kac*: a secret key shared between the AR and the CN
- EBU(*CoA, CNA*) = { *HoA, Cci, Seq#, MACebu* }[, *T(MN,CN)*]
- EBA(*CNA,CoA*) = { *Status, [Tck,] Nic, Seq#, MACeba* }
- *MACar: First (96, HMAC_SHA1 (Kac, CoA* | *CN* | EBU*)*
- *MACebu*: *First (96, HMAC_SHA1 (Kbmperm, CoA* | *CNA* | EBU*)*
- *MACeba*: *First (96, HMAC_SHA1 (Kbmperm, CoA* | *CNA* | EBA*)*
* Other notations are as same as those which are used in the T-BUA protocol
* After the first CN registration, the EBU message doesn't include *T(MN,CN)*

Figure 8. The context-aware BU phase in the caTBUA.

Let π_0 and π_1 be the long-term steady state probabilities that an MN stays in the current network and the MN moves to another network, respectively. Then, π_0 and π_1 are given by

$$\pi_0 = p\pi_0 + (1-p)\pi_1, \tag{2}$$

$$\pi_1 = (1-p)\pi_0 + p\pi_1, \tag{3}$$

where $\pi_0 + \pi_1 = 1$.

4.2. Network topology model

We consider the simple network topology shown in Figure 9. The MN in Figure 9 attaches to one of the ARs through the wireless interface. In addition, we assume that the HA and the CN

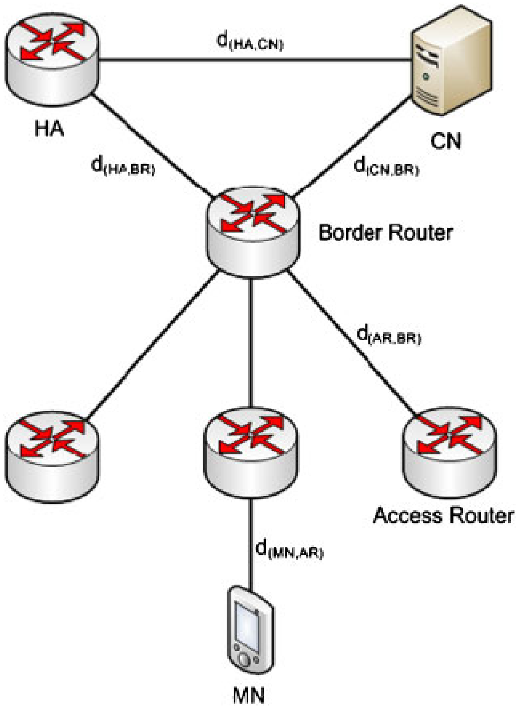


Figure 9. The network topology.

are connected to the networks via the wired interface. Suppose $d_{(x,y)}$ is a hop distance between x and y. Then, the following notations are used in this paper [20].

- $d_{(\mathrm{HA,CN})}$: 10 hops.
- $d_{(\mathrm{HA,BR})}$: 4 hops.
- $d_{(\mathrm{CN,BR})}$: 3 hops.
- $d_{(\mathrm{AR,BR})}$: 2 hops.
- $d_{(\mathrm{MN,AR})}$: 1 hop.

We also consider the transmission unit costs for both wired links and wireless links. The following notations are used to express the transmission unit costs [21–23].

- α: transmission unit cost in a wired link, 1.
- β: transmission unit cost in a wireless link, 1.5.

5. PERFORMANCE ANALYSIS

In this section, we develop cost models that estimate the authentication costs for given protocols and derive a function for calculating authentication message transmission latency. Finally, we

DOI: 10.1002/dac

provide the numerical results that compare the proposed caTBUA protocol with the RR, SSK, and TBUA protocols.

5.1. Total authentication cost

In this subsection, we develop the total authentication cost models for each protocol. All authentication protocols are assumed to have the same authentication cost between the MN and the HA. Thus, in our total authentication cost models, we concentrate upon the total authentication cost occurring between the MN and the CN. Note that the total authentication cost involves all authentication-related costs so that the registration cost for the CN is also involved. The total authentication cost, $C_{\text{total}}^{(\cdot)}$, consists of the authentication cost, $C_{\text{auth}}^{(\cdot)}$, and the authentication refresh cost, $C_{\text{reau}}^{(\cdot)}$. Accordingly, the total authentication cost is calculated as [21, 22]

$$C_{\text{total}}^{(\cdot)} = C_{\text{auth}}^{(\cdot)} + C_{\text{reau}}^{(\cdot)}, \tag{4}$$

where $(\cdot)$ is the protocol indicator. Suppose T and R are the average resident time and the authentication refresh time, respectively. Then, by the long-term steady state probabilities presented in Section 4.1, $C_{\text{auth}}^{(\cdot)}$ and $C_{\text{reau}}^{(\cdot)}$ in Equation (4) are rewritten as follows:

$$C_{\text{auth}}^{(\cdot)} = \frac{\pi_1 \cdot C_{\text{auth}'}^{(\cdot)}}{T} \quad \text{and} \quad C_{\text{reau}}^{(\cdot)} = \frac{\pi_0 \cdot C_{\text{auth}'}^{(\cdot)} \cdot \varpi}{T}, \tag{5}$$

where $\varpi = \lfloor T/R \rfloor$. Suppose MSG$(x, y)$ is a message sent from x to y. Then, $S_{\text{MSG}(x,y)}$ means a message size sent from x to y. For instance, $S_{\text{BU(MN,HA)}}$ is the size of BU message sent from the MN to the HA. Then, the following notations are used in this paper [23, 24].

- $S_{\text{BU(MN,CN)}}$: 72 bytes.
- $S_{\text{BA(CN,MN)}}$: 72 bytes.
- $S_{\text{HoTI(MN,HA)}}$: 128 bytes.
- $S_{\text{HoTI(HA,CN)}}$: 56 bytes.
- $S_{\text{HoT(CN,HA)}}$: 64 bytes.
- $S_{\text{HoT(HA,MN)}}$: 136 bytes.
- $S_{\text{CoTI(MN,CN)}}$: 56 bytes.
- $S_{\text{CoT(CN,MN)}}$: 64 bytes.
- S_{ticket}: 84 bytes.

In the above notations, S_{ticket} is the size of ticket used in the TBUA and the proposed caTBUA protocols.

5.1.1. The RR protocol. As a default authentication protocol for MIPv6, the RR protocol is used. Suppose n is the number of MNs in the networks. The authentication cost for the RR protocol, $C_{\text{auth}}^{(\text{rr})}$, is expressed as follows:

$$C_{\text{auth}}^{(\text{rr})} = \sum_{i=1}^{n} \left[\frac{\pi_1 \cdot C_{\text{auth}'}^{(\text{rr})}}{T} \right], \tag{6}$$

DOI: 10.1002/dac

where $C^{(\mathrm{rr})}_{\mathrm{auth}'}$ is the sum of the RR procedure cost, $C^{(\mathrm{rr})}_{\mathrm{rr}}$, and the registration procedure cost, $C^{(\mathrm{rr})}_{\mathrm{reg}}$. Then, $C^{(\mathrm{rr})}_{\mathrm{rr}}$ is expressed as follows:

$$\begin{aligned} C^{(\mathrm{rr})}_{\mathrm{rr}} &= \sigma_{(\mathrm{MN,HA})}\cdot S_{\mathrm{HoTI(MN,HA)}} + \sigma_{(\mathrm{HA,CN})}\cdot S_{\mathrm{HoTI(HA,CN)}} + \sigma_{(\mathrm{MN,CN})}\cdot S_{\mathrm{CoTI(MN,CN)}} \\ &\quad + \sigma_{(\mathrm{CN,HA})}\cdot S_{\mathrm{HoT(CN,HA)}} + \sigma_{(\mathrm{HA,MN})}\cdot S_{\mathrm{HoT(HA,MN)}} + \sigma_{(\mathrm{CN,MN})}\cdot S_{\mathrm{CoT(CN,MN)}}, \end{aligned} \tag{7}$$

where $\sigma_{(x,y)}$ is a distance coefficient between x and y. For instance, $\sigma_{(\mathrm{MN,HA})}$ is calculated as $\alpha\cdot d_{(\mathrm{MN,AR})} + \beta\cdot(d_{(\mathrm{AR,BR})} + d_{(\mathrm{HA,BR})})$. $C^{(\mathrm{rr})}_{\mathrm{reg}}$ is expressed as follows:

$$C^{(\mathrm{rr})}_{\mathrm{reg}} = \sigma_{(\mathrm{MN,CN})}\cdot S_{\mathrm{BU(MN,CN)}} + \sigma_{(\mathrm{CN,MN})}\cdot S_{\mathrm{BA(CN,MN)}}. \tag{8}$$

The authentication refresh cost for the RR protocol, $C^{(\mathrm{rr})}_{\mathrm{reau}}$, is expressed as follows:

$$C^{(\mathrm{rr})}_{\mathrm{reau}} = \sum_{i=1}^{n}\left[\frac{\pi_0\cdot C^{(\mathrm{rr})}_{\mathrm{auth}'}\cdot\varpi}{T}\right], \tag{9}$$

where $C^{(\mathrm{rr})}_{\mathrm{auth}'} = C^{(\mathrm{rr})}_{\mathrm{rr}} + C^{(\mathrm{rr})}_{\mathrm{reg}}$.

5.1.2. The SSK protocol. The SSK protocol uses the pre-shared key materials for generating *kbm*. This eliminates the costs associated with the RR procedure. The authentication cost for the SSK protocol, $C^{(\mathrm{ssk})}_{\mathrm{auth}}$, is expressed as follows:

$$C^{(\mathrm{ssk})}_{\mathrm{auth}} = \sum_{i=1}^{n}\left[\frac{\pi_1\cdot C^{(\mathrm{ssk})}_{\mathrm{auth}'}}{T}\right], \tag{10}$$

where $C^{(\mathrm{ssk})}_{\mathrm{auth}'} = C^{(\mathrm{ssk})}_{\mathrm{reg}}$. Then, $C^{(\mathrm{ssk})}_{\mathrm{reg}}$ is expressed as follows:

$$C^{(\mathrm{ssk})}_{\mathrm{reg}} = \sigma_{(\mathrm{MN,CN})}\cdot S_{\mathrm{BU(MN,CN)}} + \sigma_{(\mathrm{CN,MN})}\cdot S_{\mathrm{BA(CN,MN)}}. \tag{11}$$

The authentication refresh cost for the SSK protocol, $C^{(\mathrm{ssk})}_{\mathrm{reau}}$, is expressed as follows:

$$C^{(\mathrm{ssk})}_{\mathrm{reau}} = \sum_{i=1}^{n}\left[\frac{\pi_0\cdot C^{(\mathrm{ssk})}_{\mathrm{auth}'}\cdot\varpi}{T}\right]. \tag{12}$$

5.1.3. The TBUA protocol. The TBUA protocol has three phases, namely the ticket issue, EBU, and CBU phases. The ticket issue phase, which exchanges the HoTI message and the HoT message including the ticket between the MN and the HA, is only performed once per CN. Similarly, the EBU message including the ticket is performed for one time per CN. With this in mind, suppose $C^{(\mathrm{tbua})}_{\mathrm{ti}}$, $C^{(\mathrm{tbua})}_{\mathrm{ebu}}$, and $C^{(\mathrm{tbua})}_{\mathrm{cbu}}$ are the ticket issue cost, EBU cost, and CBU cost, respectively. Then, the authentication cost for the TBUA protocol, $C^{(\mathrm{tbua})}_{\mathrm{auth}}$, is expressed as:

$$C^{(\mathrm{tbua})}_{\mathrm{auth}} = \sum_{i=1}^{n}\left[\frac{\pi_1\cdot C^{(\mathrm{tbua})}_{\mathrm{auth}'}}{T}\right], \tag{13}$$

where $C^{(\mathrm{tbua})}_{\mathrm{auth}'} = C^{(\mathrm{tbua})}_{\mathrm{ti}} + C^{(\mathrm{tbua})}_{\mathrm{ebu}} + C^{(\mathrm{tbua})}_{\mathrm{cbu}}$. Let φ be an initial value per CN. Then, $C^{(\mathrm{tbua})}_{\mathrm{ti}}$ is expressed as follows:

$$C^{(\mathrm{tbua})}_{\mathrm{ti}} = \varphi \cdot (\sigma_{(\mathrm{MN,HA})} \cdot S_{\mathrm{HoTI(MN,HA)}} + \sigma_{(\mathrm{HA,MN})} \cdot S_{\mathrm{HoT(HA,MN)}} + S_{\mathrm{ticket}}), \tag{14}$$

where $\varphi = 1$ means that the authentication for the CN is the first time. For the case where the authentication for the CN is not being performed for the first time, φ is set as 0. Similarly, $C^{(\mathrm{tbua})}_{\mathrm{ebu}}$ is expressed as follows:

$$C^{(\mathrm{tbua})}_{\mathrm{ebu}} = \sigma_{(\mathrm{MN,CN})} \cdot (S_{\mathrm{BU(MN,CN)}} + \varphi \cdot S_{\mathrm{ticket}}) + \sigma_{(\mathrm{CN,MN})} \cdot S_{\mathrm{BA(CN,MN)}}, \tag{15}$$

where S_{ticket} is involved in $C^{(\mathrm{tbua})}_{\mathrm{ebu}}$ only when the authentication for the CN is being performed for the first time. Then, $C^{(\mathrm{tbua})}_{\mathrm{cbu}}$ is expressed as follows:

$$C^{(\mathrm{tbua})}_{\mathrm{cbu}} = \sigma_{(\mathrm{MN,CN})} \cdot S_{\mathrm{BU(MN,CN)}} + \sigma_{(\mathrm{CN,MN})} \cdot S_{\mathrm{BA(CN,MN)}}. \tag{16}$$

The authentication refresh cost for the TBUA protocol, $C^{(\mathrm{tbua})}_{\mathrm{reau}}$, is expressed as follows:

$$C^{(\mathrm{tbua})}_{\mathrm{reau}} = \sum_{i=1}^{n} \left[\frac{\pi_0 \cdot C^{(\mathrm{tbua})}_{\mathrm{auth}'} \cdot \varpi}{T} \right]. \tag{17}$$

5.1.4. The caTBUA protocol. In the proposed caTBUA protocol, the MN's context information is carefully considered to execute appropriate authentication protocols. As explained in Section 4, the class of the MN is determined by *TRUSTmn*. The authentication cost for the caTBUA protocol, $C^{(\mathrm{catbua})}_{\mathrm{auth}}$, is expressed as follows:

$$C^{(\mathrm{catbua})}_{\mathrm{auth}} = \sum_{i=1}^{n \cdot c_{\mathrm{h}}} \left[\frac{\pi_1 \cdot C^{(\mathrm{ssk})}_{\mathrm{auth}'}}{T} \right] + \sum_{i=1}^{n \cdot c_{\mathrm{m}}} \left[\frac{\pi_1 \cdot C^{(\mathrm{tbua})}_{\mathrm{auth}'}}{T} \right] + \sum_{i=1}^{n \cdot c_{\mathrm{l}}} \left[\frac{\pi_1 \cdot C^{(\mathrm{rr})}_{\mathrm{auth}'}}{T} \right], \tag{18}$$

where c_{h}, c_{m}, and c_{l} are the ratios of high class, middle class, and low class, respectively.

The authentication refresh cost for the caTBUA protocol, $C^{(\mathrm{catbua})}_{\mathrm{reau}}$, is expressed as follows:

$$C^{(\mathrm{catbua})}_{\mathrm{reau}} = \sum_{i=1}^{n \cdot c_{\mathrm{h}}} \left[\frac{\pi_0 \cdot C^{(\mathrm{ssk})}_{\mathrm{auth}'} \cdot \varpi}{T} \right] + \sum_{i=1}^{n \cdot c_{\mathrm{m}}} \left[\frac{\pi_0 \cdot C^{(\mathrm{tbua})}_{\mathrm{auth}'} \cdot \varpi}{T} \right] + \sum_{i=1}^{n \cdot c_{\mathrm{l}}} \left[\frac{\pi_0 \cdot C^{(\mathrm{rr})}_{\mathrm{auth}'} \cdot \varpi}{T} \right]. \tag{19}$$

5.2. Total authentication message transmission latency

The authentication message transmission latency intensely affects the system performance. Here we define the total authentication message transmission latency as the sum of the delay of authentication-related message exchange $D^{(\cdot)}_{\mathrm{auth}}$ and the delay of registration-related message exchange $D^{(\cdot)}_{\mathrm{regi}}$. Then, the cumulative total authentication message transmission latency, $L^{(\cdot)}_{\mathrm{total}}$, is expressed as follows:

$$L^{(\cdot)}_{\mathrm{total}} = \sum_{i=1}^{n} \mu [D^{(\cdot)}_{\mathrm{auth}} + D^{(\cdot)}_{\mathrm{regi}}], \tag{20}$$

where μ is the number of handovers. Note that the cumulative total authentication message transmission latency represents the cumulative total authentication message transmission latency for all MNs. The following notations are used to express the transmission bandwidth, link, and processing latencies.

- B_{wd}: wired link bandwidth, 100 Mbps.
- B_{wl}: wireless link bandwidth, 11 Mbps.
- L_{wd}: wired link latency, 0.5 ms.
- L_{wl}: wireless link latency, 2 ms.
- P_{t}: routing lookup and processing delay, 0.001 ms.

Then, we define the following functions for calculating the message transmission times over wired and wireless links. Suppose $T(S_{\mathrm{MSG}(x,y)})$ and $\widetilde{T}(S_{\mathrm{MSG}(x,y)})$ are the functions for wired and wireless links, respectively.

$$T(S_{\mathrm{MSG}(x,y)}) = d_{(x,y)} \cdot \left[\frac{S_{\mathrm{MSG}(x,y)}}{B_{\mathrm{wd}}} + L_{\mathrm{wd}} \right] + P_{\mathrm{t}} \left[d_{(x,y)} + 1 \right], \tag{21}$$

$$\widetilde{T}(S_{\mathrm{MSG}(x,y)}) = \frac{S_{\mathrm{MSG}(x,y)}}{B_{\mathrm{wl}}} + L_{\mathrm{wl}}, \tag{22}$$

where $S_{\mathrm{MSG}(x,y)}$ is the message size sent from x to y.

5.2.1. The RR protocol. In the RR protocol, the delay of authentication-related message exchange, $D^{(\mathrm{rr})}_{\mathrm{auth}}$, is expressed as follows:

$$\begin{aligned} D^{(\mathrm{rr})}_{\mathrm{auth}} = {} & \widetilde{T}(S_{\mathrm{HoTI(MN,AR)}}) + T(S_{\mathrm{HoTI(AR,HA)}}) + T(S_{\mathrm{HoTI(HA,CN)}}) + \widetilde{T}(S_{\mathrm{CoTI(MN,AR)}}) \\ & + T(S_{\mathrm{CoTI(AR,CN)}}) + T(S_{\mathrm{HoT(CN,HA)}}) + T(S_{\mathrm{HoT(HA,AR)}}) + \widetilde{T}(S_{\mathrm{HoT(AR,MN)}}) \\ & + T(S_{\mathrm{CoT(CN,AR)}}) + \widetilde{T}(S_{\mathrm{CoT(AR,MN)}}). \end{aligned} \tag{23}$$

The delay of registration-related message exchange, $D^{(\mathrm{rr})}_{\mathrm{regi}}$, is expressed as follows:

$$D^{(\mathrm{rr})}_{\mathrm{regi}} = \widetilde{T}(S_{\mathrm{BU(MN,AR)}}) + T(S_{\mathrm{BU(AR,CN)}}) + T(S_{\mathrm{BA(CN,AR)}}) + \widetilde{T}(S_{\mathrm{BA(AR,MN)}}). \tag{24}$$

Throughout Equations (23) and (24), the cumulative total authentication message transmission latency, $L^{(\mathrm{rr})}_{\mathrm{total}}$, is expressed as follows:

$$L^{(\mathrm{rr})}_{\mathrm{total}} = \sum_{i=1}^{n} \mu[D^{(\mathrm{rr})}_{\mathrm{auth}} + D^{(\mathrm{rr})}_{\mathrm{regi}}]. \tag{25}$$

5.2.2. The SSK protocol. In the SSK protocol, there is no need for additional authentication-related message exchanges. Accordingly, only the delay of registration-related message exchange, $D^{(\mathrm{ssk})}_{\mathrm{regi}}$, is involved in the cumulative total authentication message transmission latency, $L^{(\mathrm{ssk})}_{\mathrm{total}}$, as follows:

$$L^{(\mathrm{ssk})}_{\mathrm{total}} = \sum_{i=1}^{n} \mu[D^{(\mathrm{ssk})}_{\mathrm{regi}}], \tag{26}$$

where $D^{(\mathrm{ssk})}_{\mathrm{regi}}$ is calculated the same as Equation (24).

DOI: 10.1002/dac

5.2.3. The TBUA protocol. The delay of authentication-related message exchange affecting the authentication message transmission latency, $D_{\mathrm{auth}}^{(\mathrm{tbua})}$, is expressed as follows:

$$D_{\mathrm{auth}}^{(\mathrm{tbua})} = \varphi \cdot [\widetilde{T}(S_{\mathrm{HoTI(MN,AR)}}) + T(S_{\mathrm{HoTI(AR,HA)}}) \\ + T(S_{\mathrm{HoT(HA,AR)}} + S_{\mathrm{ticket}}) + \widetilde{T}(S_{\mathrm{HoT(AR,MN)}} + S_{\mathrm{ticket}})]. \tag{27}$$

The registration-related message exchange consists of the EBU and CBU phases. From the perspective of authentication message transmission latency, the MN can communicate with its CN after the completion of EBU phase. Accordingly, the delay of registration-related message exchange, $D_{\mathrm{regi}}^{(\mathrm{tbua})}$, is expressed as follows:

$$D_{\mathrm{regi}}^{(\mathrm{tbua})} = \widetilde{T}(S_{\mathrm{BU(MN,AR)}} + \varphi \cdot S_{\mathrm{ticket}}) + T(S_{\mathrm{BU(AR,CN)}} + \varphi \cdot S_{\mathrm{ticket}}) \\ + T(S_{\mathrm{BA(CN,AR)}}) + \widetilde{T}(S_{\mathrm{BA(AR,MN)}}). \tag{28}$$

Throughout Equations (27) and (28), the cumulative total authentication message transmission latency, $L_{\mathrm{total}}^{(\mathrm{tbua})}$, is expressed as follows:

$$L_{\mathrm{total}}^{(\mathrm{tbua})} = \sum_{i=1}^{n} \mu [D_{\mathrm{auth}}^{(\mathrm{tbua})} + D_{\mathrm{regi}}^{(\mathrm{tbua})}]. \tag{29}$$

5.2.4. The caTBUA protocol. The cumulative total authentication message transmission latency for the caTBUA protocol, $L_{\mathrm{total}}^{(\mathrm{catbua})}$, is expressed as follows:

$$L_{\mathrm{total}}^{(\mathrm{catbua})} = \sum_{i=1}^{n \cdot c_{\mathrm{h}}} \mu \left[D_{\mathrm{regi}}^{(\mathrm{ssk})} \right] + \sum_{i=1}^{n \cdot c_{\mathrm{m}}} \mu \left[D_{\mathrm{auth}}^{(\mathrm{tbua})} + D_{\mathrm{regi}}^{(\mathrm{tbua})} \right] + \sum_{i=1}^{n \cdot c_{\mathrm{l}}} \mu \left[D_{\mathrm{auth}}^{(\mathrm{rr})} + D_{\mathrm{regi}}^{(\mathrm{rr})} \right], \tag{30}$$

where the ratios for the class of the MN are considered.

5.3. Numerical results

The numerical results for the authentication cost and transmission latency calculations described above are presented in this subsection. In the case of the proposed caTBUA protocol, several scenarios for the MNs' context information are considered and the ratios of *TRUSTmn* are therefore given for three cases where the ratios of c_{h}, c_{m}, and c_{l} are as 7:3:2, 4:4:4, and 2:3:7.

5.3.1. Authentication cost. In Figure 10, the impacts of the long-term steady state probability that the MN moves to another network (π_1), the average resident time (T), and the number of MNs (n) on the authentication cost are investigated separately.

First, in the example shown in Figure 10(a), we set n and T as 12 and 450 s, respectively. Then, π_1 is increased to show the variations of the authentication cost. As π_1 increases, the authentication costs for all protocols are increased because the probability that the MN moves to another network increases. As the figure shows, the SSK protocol provides the best performance compared to others because the SSK protocol uses pre-shared key materials. In this case, the TBUA protocol

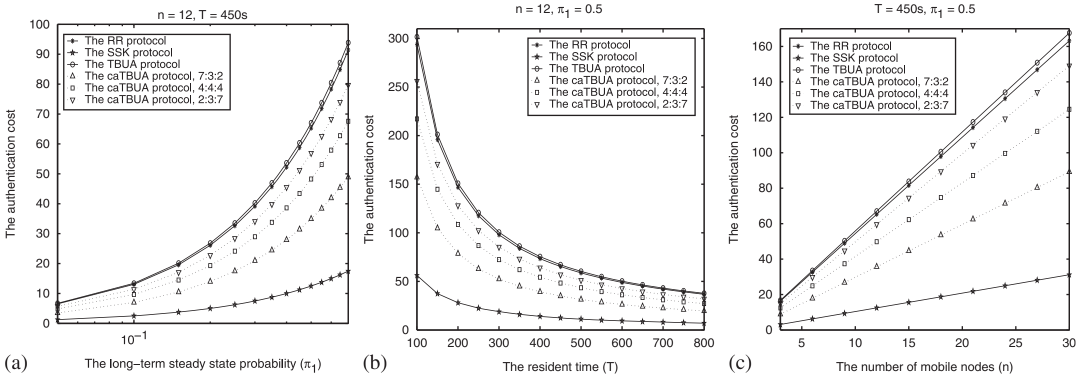


Figure 10. The variations of the authentication cost: (a) an increment of π_1; (b) an increment of T; and (c) an increment of n.

consumes more authentication costs than others due to its extra ticket issue, EBU, and CBU costs. This phenomenon gets larger as π_1 increases. In the case of the proposed caTBUA protocol, the performance is the best when the ratio of *TRUSTmn* is set as 7:3:2. This is because as the value of *TRUSTmn* is high, there is no need to perform the high-level authentication, such as the RR and TBUA protocols. Next, we set n and π_1 as 12 and 0.5, respectively. Then, T is increased to show the variations of the authentication cost. As T increases, the authentication costs for all protocols are decreased. The impact of the average resident time is shown in Figure 10(a). As with the results presented in Figure 10(a), the SSK protocol shows the best performance compared to others and the TBUA protocol consumes more of the authentication costs. Finally, the impact of the number of MNs is examined in Figure 10(a) with $T = 450$ s and $\pi_1 = 0.5$. As we can see, the SSK protocol provides the best performance, whereas the TBUA protocol requires more than others.

Now, we investigate the impacts of the long-term steady state probability, where the MN maintains the same network π_0, the authentication refresh time (R), and the number of MNs (n) on the authentication refresh cost. As we can see in Figure 11(a–c), the SSK protocol provides the best performance, whereas the TBUA protocol requires the most resources. These results follow the same pattern as those presented in Figure 10. Considered together, the results presented in Figures 10 and 11 confirm that the proposed caTBUA protocol provides a good balance between security and efficiency. In cases where the value of *TRUSTmn* is high, the caTBUA protocol incurs the same authentication cost as the SSK protocol, but preserves security due to its ability to select the appropriate authentication method based on the MN's context information.

5.3.2. Authentication message transmission latency. In the example shown in Figure 12(a), we set the number of handovers as 5 and change the number of MNs. Once again, the SSK protocol provides the best performance, whereas the RR protocol requires the highest transmission latency. In the case of the proposed caTBUA protocol, the performance is best when the ratio of *TRUSTmn* is set as 4:4:4. In addition, for the cases where the ratios are 4:4:4 and 7:3:2, both provide a better performance than either the RR or the TBUA protocols.

In Figure 12(b), we set the number of MNs as 12 and change the number of handovers. For this scenario, in the case of the proposed caTBUA protocol, the performance is best when the

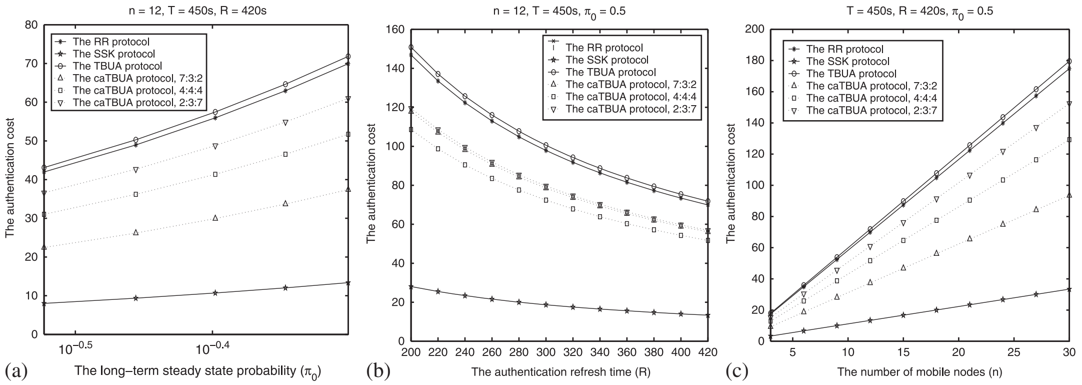


Figure 11. The variations of the authentication refresh cost: (a) an increment of π_0; (b) an increment of R; and (c) an increment of n.

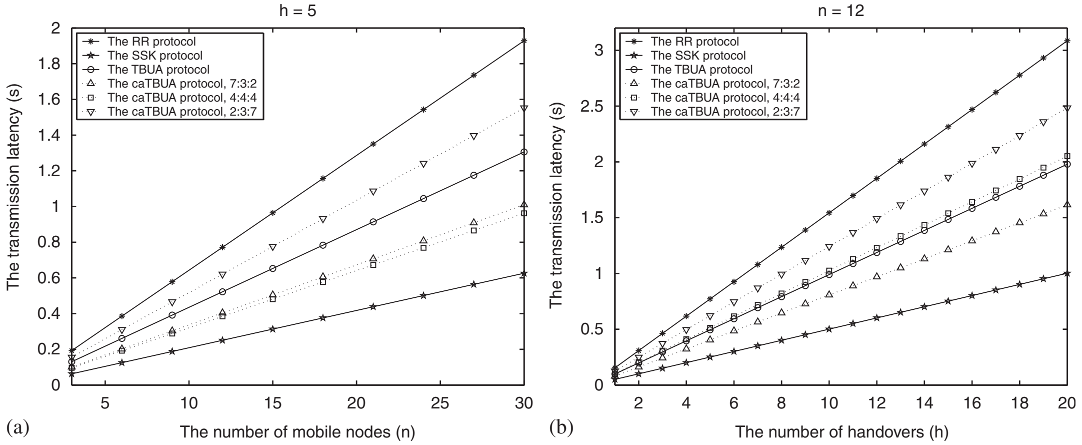


Figure 12. The cumulative total authentication message transmission latency: (a) the cumulative total authentication message transmission latency as a function of the number of mobile nodes and (b) the cumulative total authentication message transmission latency as a function of the number of handovers.

ratio of *TRUSTmn* is set as 7:3:2. Unsurprisingly, of all the protocols the SSK protocol once again provides the best performance.

6. CONCLUSION

In MIPv6-based network environments, the BU is uniquely used as a building block. Although many BU authentication protocols have been proposed to protect the malicious BU process from the various attacks, most of them cannot maintain a good balance in the presentation of different performance factors, such as security and efficiency. In this paper, we have introduced a caTBUA protocol for trust-enabled network environments. The proposed caTBUA protocol selectively performs the

DOI: 10.1002/dac

CoA validation based on the network trust value which is calculated by the MN's context value, i.e. trust, location, current time, during the BU process. In order to show how our protocol optimizes security and efficiency together, we presented the numerical analysis with comparisons between our protocol and others. The presented numerical results corroborate that the proposed caTBUA protocol, which has the ability to select the appropriate authentication method based on the MN's context information, yields a better performance compared with the existing BU authentication protocols while providing a balance between security and efficiency.

ACKNOWLEDGEMENTS

The authors thank the anonymous reviewers for their valuable comments on this paper.

REFERENCES

1. Johnson D, Perkins C, Arkko J. Mobility support in IPv6. *IETF RFC 3775*, June 2004.
2. Ren K, Lou W, Zeng K, Bao F, Zhou J, Deng RH. Routing optimization security in mobile IPv6. *Computer Networks* 2006; **50**(13):2401–2419.
3. You I. Improving the CGA-OMIPv6 protocol for low-power mobile nodes. *Proceedings of International Conference on Computational Science and Its Applications* (*ICCSA*) *2006*, Glasgow, U.K., May 2006; 336–343.
4. You I, Lim J. Advanced agent-delegated route optimization protocol for efficient multimedia services at low-battery devices. *Proceedings of International Multimedia Modeling Conference* (*MMM*) *2007*, Singapore, January 2007; 479–486.
5. Haddad W, Madour L, Arkko J, Dupont F. Applying cryptographically generated addresses to optimize MIPv6 (CGA-OMIPv6). *IETF Internet Draft*, draft-haddad-mip6-cga-omipv6-04 (work in progress), November 2005.
6. Arkko J, Vogt C, Haddad W. Enhanced route optimization for mobile IPv6. *IETF RFC 4866*, May 2007.
7. Dupont F, Haddad W. Optimizing mobile IPv6 (OMIPv6). *IETF Internet Draft*, draft-dupont-mipshop-omipv6-00 (work in progress), February 2006.
8. O'Shea G, Roe M. Child-proof authentication for MIPv6 (CAM). *ACM Computer Communications Review* 2001; **31**(2):4–8.
9. Roe M, Aura T, O'Shea G, Arkko J. Authentication of mobile IPv6 binding updates and acknowledgments. *IETF Internet Draft*, draft-roe-mobileip-updateauth-02 (work in progress), March 2002.
10. Montenegro G, Castelluccia C. Crypto based identifiers (CBIDs): concepts and applications. *ACM Transactions on Information and System Security* 2004; **7**(1):97–127.
11. You I, Cho K. A security proxy based protocol for authenticating the mobile IPv6 binding updates. *Proceedings of International Conference on Computational Science and its Applications* (*ICCSA*) *2004*, Assisi, Italy, May 2004; 167–174.
12. Haddad W, Krishnan S, Dupont F. Mobility signaling delegation in OptiSEND. *IETF Internet Draft*, draft-haddad-mipshop-mobisig-del-02 (work in progress), October 2006.
13. Okazaki S, Desai A, Gentry C, Kempf J, Silverberg A, Yin YL. Securing MIPv6 binding updates using address based keys (ABKs). *IETF Internet Draft*, draft-okazaki-mobileip-abk-01 (work in progress), October 2002.
14. You I. A ticket based binding update authentication method for trusted nodes in mobile IPv6 domain. *Proceedings of IFIP International Conference on Embedded and Ubiquitous Computing* (*EUC*) *2007*, Taipei, Taiwan, December 2007; 808–819.
15. Aura T. Cryptographically generated addresses (CGA). *IETF RFC 3972*, March 2005.
16. Bradner S, Mankin A, Schiller JI. A framework for purpose-built keys (PBK). *IETF Internet Draft*, draft-bradner-pbk-frame-06 (work in progress), June 2003.
17. Perkins C. Securing mobile IPv6 route optimization using a static shared key. *IETF RFC 4449*, June 2006.
18. Camp T, Boleng J, Davies V. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing* 2002; **2**(5):483–502.
19. Kim JH, Hong CS, Shon T. A lightweight NEMO protocol to support 6LoWPAN. *ETRI Journal* 2008; **30**(5): 685–695.
20. Pack S, Shen X, Mark JW, Pan J. Adaptive route optimization in hierarchical mobile IPv6 networks. *IEEE Transactions on Mobile Computing* 2007; **6**(8):903–914.

DOI: 10.1002/dac

21. Lee J-H, Lim H-J, Chung T-M. A competent global mobility support scheme in NETLMM. *International Journal of Electronics and Communications*, DOI: 10.1016/j.aeue.2008.07.010, On-line published, 2008.
22. Lee J-H, Chung T-M, Gundavelli S. A comparative signaling cost analysis of hierarchical mobile IPv6 and proxy mobile IPv6. *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) 2008*, Cannes, France, September 2008; 1–6.
23. Lee J-H, Han Y-H, Gundavelli S, Chung T-M. A comparative performance analysis on hierarchical mobile IPv6 and proxy mobile IPv6. *Telecommunication Systems* 2009; **41**(4):279–292.
24. Grech S, Poncela J, Serna P. An analysis of mobile IPv6 signaling load in next generation mobile networks. *Proceedings of IFIP TC6/WG6.8 Conference on Mobile and Wireless Communication Networks (MWCN) 2004*, Paris, France, October 2004; 71–82.

AUTHORS' BIOGRAPHIES

Ilsun You received his MS and PhD degrees in Computer Science from Dankook University, Seoul, Korea in 1997 and 2002, respectively. Since March 2005, he has been an Assistant Professor in the School of Information Science at the Korean Bible University, South Korea. His main research interests include network security and authentication. He is a member of the IEICE, KIISC, and KSII.

Jong-Hyouk Lee received his BS degree in Information System Engineering from Daejeon University, Daejeon, Korea, in 2004 and his MS degree in Computer Engineering at Sungkyunkwan University, Suwon, Korea, in 2007. He obtained his PhD degree in Electrical and Computer Engineering at Sungkyunkwan University in 2010. He worked as an intern for IMARA Team, INRIA, France, in 2009. He received Excellent Research Awards (two times) from Department of Electrical and Computer Engineering, Sungkyunkwan University. He received Best Paper Award from *International Conference on Systems and Networks Communications* 2008. Currently, he is a postdoctoral researcher in IMARA Team, INRIA, France. He is now developing a solution to make efficient and secure communications for NEMO-based vehicular networks. His research interests include mobility management, security, and performance analysis based on protocol operation for next-generation wireless mobile networks.

Bonam Kim received the PhD degree in Computer Science and Software Engineering from the Auburn University, Alabama, U.S.A. in 2006. She joined the School of Electrical and Computer Engineering, Chungbuk National University in March 2007. Dr Kim is very active in international academic activities and has participated in the organization of many international workshops. Her current research interests are in the areas of wireless *ad hoc* and sensor networks, network security, and MIPv6.

DOI: 10.1002/dac