SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2013; **6**:401–419
Published online 9 October 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.625

SPECIAL ISSUE PAPER

ALPP: anonymous and location privacy preserving scheme for mobile IPv6 heterogeneous networks†

Sanaa Taha[1,2] and Xuemin (Sherman) Shen[1]*

[1] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada
[2] Faculty of Computers and Information, Cairo University, Cairo, Egypt

ABSTRACT

The integration of mobile IPv6 heterogeneous networks enhances networking performance; however, it also breaks mobile node's anonymity and location privacy. In this paper, we propose an anonymous and location privacy preserving (ALPP) scheme that consists of two complementary subschemes: anonymous home binding update and anonymous return routability. In addition, anonymous mutual authentication and key establishment schemes have been proposed to work in conjunction with ALPP to authenticate a mobile node to its foreign gateway and create a shared key between them. ALPP adds anonymity and location privacy services to mobile IPv6 signaling to achieve mobile senders and receivers' privacy. Unlike existing schemes, ALPP alleviates the trade-off between the networking performance and the achieved privacy level. Combining onion routing and anonymizer in ALPP scheme increases the achieved location privacy level where no entity in the network except the mobile node itself can identify this node's location. Using entropy model, we show that ALPP achieves higher degree of anonymity than the mix-based scheme. The anonymous home binding update and anonymous return routability subschemes require less computation overheads and thwart both internal and external adversaries. Simulation results demonstrate that our schemes have low control packets routing delays and are suitable for the seamless handover. Copyright © 2012 John Wiley & Sons, Ltd.

KEYWORDS

anonymity; location privacy; mobile IPv6 security; heterogeneous networking privacy; next-generation networks

***Correspondence**

Xuemin (Sherman) Shen, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada.
E-mail: xshen@bbcr.uwaterloo.ca

†Part of this paper is published in the 2011 IEEE Global Communication Conference [1].

1. INTRODUCTION

The revolution of next-generation networks enables mobile nodes (MNs) that are equipped with multiple network interfaces to perform seamless handovers across heterogeneous networks [2,3]. A seamless handover [4,5] is a vertical handover process in which an MN roams among different types of networks, such as cellular networks and Wireless Local Area Networks (WLANs), without interrupting this node's active Internet protocol (IP) session. When using this timely restricted handover process, both MN and service provider have some benefits, including low cost, wide coverage, and high bandwidth. Therefore, many applications such as infotainment and video-stream downloading explore seamless handovers to increase networking performance.

Different network layers, including data link, IP, and transport layers, engage in this seamless handover process. However, the integration of these heterogeneous networks is mainly accomplished in the IP layer. The mobile IP is the most famous mobility management protocol that is responsible for managing user's mobility across heterogeneous networks. Therefore, as all share the usage of the mobile IP, these heterogeneous networks are also called "all-IP" networks [6]. We consider the mobile IPv6 protocol [7] because, unlike mobile IPv4 protocol, it introduces the route optimization procedure. This procedure contributes in decreasing networking routing delays and hence permits the mobile IPv6 to achieve seamless handover process for roaming MNs.

Previous studies have attempted to secure the mobile IPv6 networks by focusing on the authentication and integrity problems [8–11]. Moreover, much research work has been done on anonymity and location privacy problems [1,12,13]. The anonymity of a network is the ability to hide a specific item among a group of similar items. The location privacy is the ability to prevent tracking user mobility by using any kind of geolocation schemes. As mentioned in [14] and [15], location privacy threats vary from a simple interfering personal activities,

habits, and socialities to a more dangerous physical attack after identifying a person and his or her favorite locations.

In roaming across "all-IP" networks, as shown in Figure 1, each MN has two different IP addresses: a home address (HoA) and a care-of address (CoA). The HoA is the original MN's address that is received from MN's home agent (HA), which is a router located in the MN's home network. The CoA is acquired from a foreign gateway (FG), which is a router located in the visited network. This CoA is acquired either by stateless configuration [16], using the route advertisement messages that an FG sends periodically, or by stateful configuration, using the Dynamic Host Configuration Protocol (DHCP) [17].

When moving out from its home network to a foreign network, an MN uses the mobile IPv6 control messages, home binding update (HBU), and return routability messages to perform the seamless handover process. The HBU control messages are sent to the MN's HA, whereas the return routability control messages are sent to the MN's correspondents, which are called correspondent nodes (CNs). By sending these control messages, an MN informs both its HA and its CNs about its current location that is represented by its CoA. Therefore, the roaming MN can receive any subsequent messages, destined to its HoA, at this CoA. Both HA and CN create bindings between the MN's HoA and CoA and then transmit any subsequent messages to this CoA instead of transmitting them to the MN's HoA.

In transmitting mobile IPv6 binding update (BU) messages, both MN's HoA and CoA are transmitted as plaintext; hence, they can be revealed by network's entities and attackers to privacy. The MN's HoA and CoA represent its identity and its current location, respectively. Therefore, revealing an MN's HoA means breaking its anonymity, and revealing an MN's CoA means breaking its location privacy. On one hand, some existing anonymity and location privacy schemes [18–21] require intensive computations; hence, they cannot be used in the timely restricted seamless handover processes. On the other hand, some other schemes [22,23] achieve low anonymity and location privacy levels. Therefore, the trade-off between the network performance on one side and the MN's anonymity and location privacy on the other side makes privacy preserving a challenging issue.

The contributions of this paper are twofold. Firstly, on the basis of the onion routing [24] and anonymizer [25], we propose an anonymous and location privacy preserving (ALPP) scheme that consists of two complementary subschemes: anonymous home binding update (AHBU) and anonymous return routability (ARR). Those subschemes efficiently add anonymity and location privacy services to mobile IPv6 HBU and return routability control messages, respectively, to achieve mobile senders and receivers' privacy. In other words, AHBU is used to send AHBU messages to MN's HA, whereas ARR is used to send ARR messages to MN's CN. Using the onion routing, we repeatedly encrypt the transmitted messages at each hop to protect them from traffic analysis adversaries. In addition, we adapt the traditional anonymizer, which is a fixed proxy used to hide the MN's location, by changing this anonymizer at each time the MN roams to a foreign network. Our adaptation for the anonymizer solves the single point of failure problem that occurs with the traditional anonymizer. Secondly, on the basis of the certificate-less public key cryptography (CL-PKC) [26], we propose anonymous authentication and key establishment schemes to work in conjunction with ALPP scheme. The authentication scheme is used to authenticate an MN to its FG while preserving the MN's anonymity. The challenge of proposing such a scheme is the difficulty of constructing a mutual trust between arbitrary nodes, which have not met each other before. The key establishment scheme is used to generate a shared key between an MN and its FG. Using the CL-PKC helps in decreasing the computation overhead of the proposed schemes.

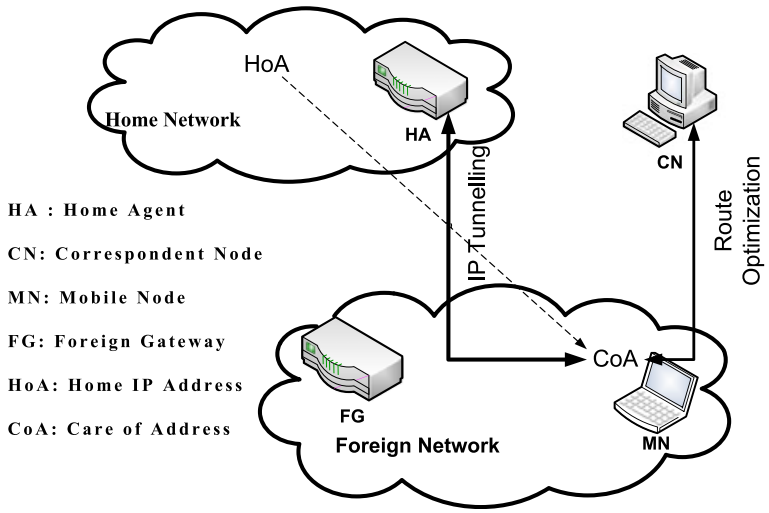

Figure 1. Roaming among mobile IPv6 heterogenous networks.

Security Comm. Networks 2013; **6**:401–419 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

Unlike existing anonymity and location privacy preserving schemes, ALPP scheme alleviates the trade-off between the network performance and the achieved privacy level. We show that the AHBU and ARR subschemes achieve high level of location privacy, where no entity in the network can reveal an MN's location except the MN itself. Moreover, using entropy model, we show that our proposed scheme achieves higher degree of anonymity than the mix-based scheme with one mix server. Additionally, extensive simulation results demonstrate that our subschemes have low routing delays; hence, they can be used during the timely restricted seamless communications. Table I shows the full name for the abbreviations used throughout this paper.

The remainder of the paper is organized as follows. Section 2 reviews the related work. The system models and an overview of the CL-PKC are presented in Section 3. The proposed scheme, ALPP, is presented in Section 4. Section 5 gives the privacy and security analysis, whereas Section 6 presents the performance evaluation and simulation results. Finally, the conclusions and future work are given in Section 7.

2. RELATED WORK

Many anonymity and location privacy schemes in mobile IPv6 networks are based on Chaum's mix [27], which introduces the idea of mix network. A mix network is a group of servers, called mix servers, that decrypts incoming messages and then retransmits them to the destinations in a different order rather than their incoming order. The goal of this mixing is to hide the sender's identity and locations and hence achieving this sender's anonymity and location privacy. The idea of mix network is employed in schemes called cascaded overlay mix network-based location privacy schemes [24,28,29]. Another Chaum's mix-based scheme, called anonymizer, which is based on a single trusted proxy that is used to hide user's identity and location information from a CN, is proposed in [25].

Table I. Acronyms' definitions.

Acronym	Definition
MN	Mobile node
HA	Home agent
CoA	Care-of address
HoA	Home address
FG	Foreign gateway
CN	Correspondent node
HBU	Home binding update
HBA	Home binding acknowledgement
CL-PKC	Certificate-less public key cryptography
CIMT	Care-of Test Init message
CTM	Care-of Test message
IFG	Intermediate FG
HIFG	Home IFG
CIFG	Correspondent IFG
E(K, M)	Encryption of message M by key K
P_{HIFG}	HIFG's public key
P_{CIFG}	CIFG's public key
$K_{MN\text{-}HA}$	Shared secret key between MN and HA
$K_{MN\text{-}FG}$	Shared secret key between MN and FG
t_i	Time stamp
PID_{MN}	MN's pseudo identity
D_{MN}	MN's partial private key

A mix-based scheme is proposed in [21] to achieve anonymity and location privacy for mobile IPv6 BU control messages. A network of mix servers, controlled by a mix center, is deployed and uses (k, n) ElGamal threshold mechanism to decrypt the BU messages received from the roaming MN. This scheme uses the mix network [27] to hide MN's location and a pseudo identity to hide MN's real identity. However, the mix center identifies the MN's HoA, CoA, HA, and FG. Therefore, the mix center can easily violate the MN's privacy. Unlike our proposed scheme, the mix-based scheme cannot be used for the timely restricted seamless communications because it has high routing delays especially with large number of mix servers.

On the basis of the anonymizer [25], a scheme with eight different levels of anonymity and location privacy is proposed in [30]. This scheme introduces a new entity, called information translating proxy (ITP), which works as an anonymizer in a mobile IPv6 network. Each MN shares a secret key with the ITP and uses this key to encrypt the HBU messages at the time of roaming. Instead of sending the BU messages directly to the MN's HA, the MN sends them to the ITP, which removes the MN's identity information and then forwards these messages to the HA. Although it presents a practical solution for location privacy, this scheme is susceptible to a single point of failure, because it uses single trusted anonymizer for all MNs. In our proposed scheme, ALPP, we use the idea of anonymizer; however, we solve the single point of failure problem by changing the anonymizer as MN moves among visited networks.

In [22], the Internet Engineering Task Force group defines the location privacy problem in the mobile IPv6 networks. The problem definition is divided into two main parts: disclosing the CoA to the CN and revealing the HoA to an eavesdropper. Furthermore, the Internet Engineering Task Force group published experimental solutions in [23] to solve only the second part of the problem. Those solutions do not address the first part of the location privacy problem, that is, unveiling the CoA to the CN. Specifically, two schemes are proposed in [23]. The first scheme uses encrypted home address (EHoA) to conceal the HoA from the adversary, whereas the second hide the HoA from the CN. However, EHoA and PHoA schemes achieve only MN's anonymity and assume that MN's location privacy is implicitly achieved. In our proposed scheme, in addition to the problems defined in [22], we solve two more privacy problems: disclosing the CoA to the HA and revealing the HoA to the FG. In Section 5, we show that our proposed subschemes, AHBU and ARR, achieve higher anonymity and location privacy levels than the EHoA and PHoA schemes.

Because the mobile IP address represents both an MN's identity and location, mobile IP-based networks have location privacy problems. Therefore, in [31], a virtual ID is used to represent MN's identity and hence separate this identity from MN's location. Therefore, extra servers are needed to map virtual IDs to MNs' current locations. However, this scheme causes a triangle routing problem because messages sent from the CN are transmitted to the MN's HA before reaching the intended MN. In [32], a name space is used to represent MN's identity, and a new layer, Host Identity Protocol (HIP), is added to the TCP/IP protocol stack. Supporting mobility and multihoming is the main goal for the HIP; additionally, it provides MN's location privacy service. We argue that HIP is a computationally expensive protocol. To initiate a communication between two entities, initiator and responder, HIP uses public key operations for entities identifications and sharing a secret key between these entities. In addition, the responder transmits a puzzle to the initiator in order to authenticate it, where it takes CPU processing time from the initiator to solve this puzzle. Therefore, HIP cannot be used with seamless communications. On the other hand, we show that ALPP scheme alleviates the trade-off between the networking performance and the achieved privacy level.

3. SYSTEM MODELS

3.1. Network model

Our network model, as shown in Figure 2, consists of a group of heterogeneous networks that use the mobile IPv6 protocol as a mobility management protocol. The mobile IPv6 protocol supports the mobile users with mobility services; therefore, mobile users can receive their communication messages while they are roaming to foreign networks. Each network of these heterogeneous networks consists of a number of MNs and a set of gateways. Each gateway has three functions: (1) to work as an HA for MNs that are originally located in its network; (2) to work as an FG for the visitor MNs; and (3) to work as an intermediate foreign gateway (IFG) for MNs that are neither visitors nor originally located in this gateway's network. Each MN defines its HA, located in its home network, and its FG, located in its current visited network. Moreover, the MN also defines a list of all IFGs, which consists of all gateways that are located in all networks, except gateways that are located in both MN's home network and currently visited network.

Using the IPsec Internet key-exchange protocol [33], each MN maintains a secret key, $K_{\text{MN-HA}}$, that is shared permanently between the MN and its HA. $K_{\text{MN-HA}}$ can be changed if the security association between the MN and its HA changes. When roaming to a foreign network, the MN sends a mobile IPv6 HBU and correspondent BU control messages to its HA and its CN, respectively, to inform them about MN's current location. Therefore, as illustrated in Section 1, any subsequent data messages can be directed to the MN's current location (MN's CoA) instead of sending them to MN's HoA. Because of the difficulty of constructing a security association between an MN and a CN, the MN needs to send return routability messages before sending the BU to the CN. In our model, we add anonymity and location privacy to the HBU and return routability messages in order to achieve senders and receivers' privacy. We argue that adding anonymity and location privacy to return routability messages is subsequently achieving privacy to correspondent BU control messages. Therefore, the roaming MN sends AHBU and ARR messages. Figure 3 depicts the control messages that are used in our proposed scheme. Note that the original BU and binding acknowledgement (BA) messages use Encapsulating Security Payload (ESP) protocol in transport mode; however, in our case, we use ESP protocol in tunnel mode.

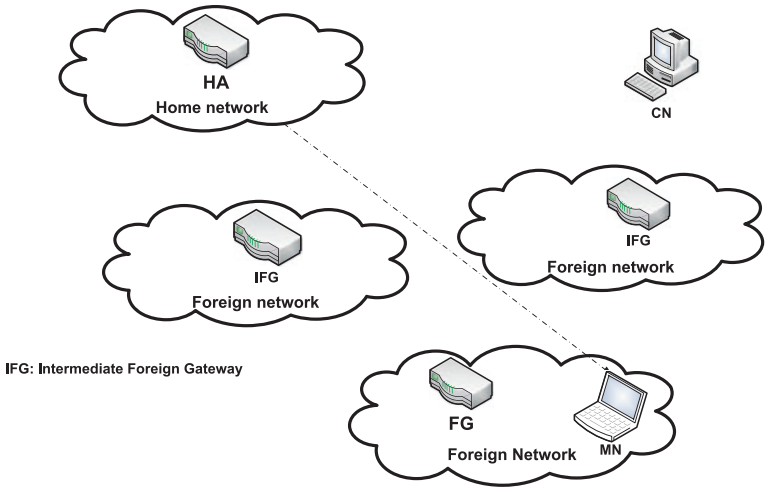

Figure 2. System model.

Security Comm. Networks 2013; **6**:401–419 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

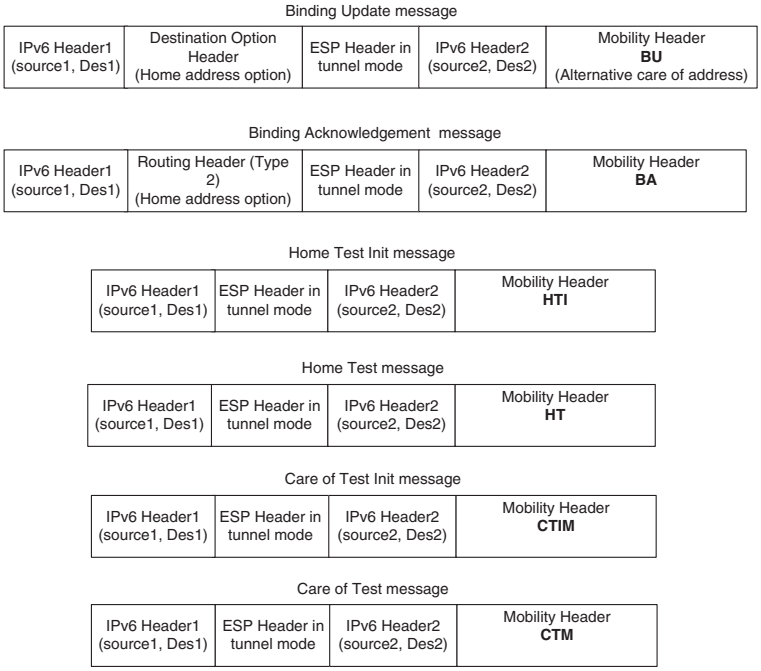

Figure 3. ALPP control messages.

3.2. Threat and trust models

Two kinds of adversaries are defined: external adversary and internal adversary. The external adversary is a passive traffic analysis attacker that analyzes the transmitted packets to deduce useful information about the identities and the locations of the senders. The external adversary investigates the time of each transmitted packet, compares the received and transmitted packets at each hop, and tracks the packet to know its destination.

The internal adversary is a network entity that intentionally observes MNs' identities and locations. In our model, we consider the HA, FG, and CN entities as internal adversaries. These entities may misuse the observed MNs' privacy information and take malicious actions towards these MNs. Therefore, HAs, FGs, and CNs are prevented to learn MNs' private information. However, these entities need to learn MN's locations because they help in MN's mobility management process. To illustrate this contradiction, consider, for instance, an HBU message that is sent from an MN to its HA. The receiver HA needs to know the MN's identity and current location and stores this information in the HA's binding cache. Therefore, the HA can forward any subsequent messages, destined to MN's HoA, to the current MN's CoA. However, at the same time, the HA may maliciously use the MN's information and violate MN's location privacy. To solve this contradiction, we let the internal adversaries to learn only part of the MNs' private information. This part is adequate to perform the MN's mobility management process without violating MN's privacy. HAs and CNs are allowed to know MNs' HoAs; however, they are unable to learn MNs' CoAs and FGs. Moreover, the FG is allowed to know the MN's CoAs, and it should not know the MNs' HoAs. All in all, each internal adversary learns a different part of MN's privacy information. Therefore, internal adversaries may collude with each other to know the whole MN's private information.

We propose a revocable privacy scheme in which one entity, the HA, can reveal the MN's privacy at the time of dispute when the MN repudiates the service. Therefore, we consider the HA as a noncolluder with other entities in the network. Moreover, we consider all other entities, including the FGs, IFGs, and CNs, as untrusted entities, and they may collude with each other to reveal the MN's private information. In addition, there is a trusted third party that generates a group key (K_{group}) for the entire networks. The created K_{group} is securely distributed by some way to all legitimate users in the system.

3.3. Certificate-less public key cryptography

A trusted key generator center (KGC) uses a security parameter, K, and runs a setup algorithm to produce two keys (s, $Param$). The master key, s, is selected randomly from $\mathbb{Z}_q^*$, where q is a large prime with $|q| = K$, and is kept secret at the KGC. The public $Param = \langle \mathbb{G}_1, \mathbb{G}_2, e, n, P, P_0, H_1, H_2 \rangle$ is transmitted to all network users. $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic groups of a large prime order, q, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear pairing function on elliptic curves [34], n is the bit length of the plaintext, P is $\mathbb{G}_1$'s generator, $P_0 = s \times P$, and $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ and $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$ are two hashing functions.

Upon receiving a request from a user A with identity ID_A, the KGC creates A's partial private key, $D_A = s \times \mathbf{Q}_A$, where $\mathbf{Q}_A = H_1(ID_A)$. The KGC securely transmits the partial private key, D_A to A. The D_A is used by A to create its public–private key pair, (P_A, S_A), as follows:

$$\begin{aligned} &x_A \in_R \mathbb{Z}_q^* \\ &S_A = x_A \times D_A \\ &X_A = x_A \times P \\ &Y_A = x_A \times P_0 = x_A \times s \times P \\ &P_A = \langle X_A, Y_A \rangle \end{aligned} \tag{1}$$

This cryptography is called CL-PKC [26] because unlike traditional public key infrastructure, a user A does not need a certificate from a trusted certificate authority. Therefore, the CL-PKC saves the computation overheads needed for certificate distribution and verification. Algorithm 1 presents the certificate-less encryption of a message m that is transmitted to a user A. Notice that the sender uses only A's identity (ID_A) and public key (P_A) to produce a ciphertext, c. In Section 5.2, we prove that if either A's identity or public key is changed by an adversary, then the encryption operation will result a failure operation ($\perp$) or an incorrect ciphertext. Moreover, to decrypt this ciphertext, $c = \langle u, v \rangle$, user A performs only one pairing function to obtain the message, $m = v \oplus H_2(\hat{e}(S_A, u))$.

Algorithm 1: Certificate-less public key encryption

Input: m, ID_A,and P_A
Output: Ciphertext c
if $\hat{e}(X_A, P_0) \neq \hat{e}(Y_A, P)$ **then**
  $c = \perp$
else
  $\mathbf{Q}_A = H_1(ID_A) \in \mathbb{G}_1^*$
  $r \in_R \mathbb{Z}_q^*$
  $c = \langle rP, m \oplus H_2(\hat{e}(\mathbf{Q}_A, Y_A)^r) \rangle$
end

In this paper, we used CL-PKC to generate a shared key between two users, A and B. User A sends its public key along with a random value, T_A, to B, which, in turn, replies with its public key, P_B, and another random number, T_B. $T_A = aP$ and $T_B = bP$, where a and b are randomly chosen by A and B, respectively. Using this transmitted information, both A and B create two keys (K_A, which is generated by A, and K_B, which is generated by B) as follows:

$$K_A = \hat{e}(\mathbf{Q}_B, Y_B)^a . \hat{e}(S_A, T_B) \tag{2}$$

$$K_B = \hat{e}(\mathbf{Q}_A, Y_A)^b . \hat{e}(S_B, T_A) \tag{3}$$

With the pairing function's properties used, it can be shown that both keys are identical as follows:

$$\begin{aligned} K_A &= \hat{e}(\mathbf{Q}_B, Y_B)^a . \hat{e}(S_A, T_B) \\ &= \hat{e}(\mathbf{Q}_B, x_B sP)^a . \hat{e}(x_A s\mathbf{Q}_A, bP) \\ &= \hat{e}(x_B s\mathbf{Q}_B, aP) . \hat{e}(\mathbf{Q}_A, x_A sP)^b \\ &= \hat{e}(S_B, T_A) . \hat{e}(\mathbf{Q}_A, Y_A)^b \\ &= K_B \end{aligned} \tag{4}$$

4. ANONYMOUS AND LOCATION PRIVACY PRESERVING SCHEME

In this section, we propose the ALPP scheme, which is used by an MN when roaming from its home network to another foreign network. As mentioned in Section 1, this period is called seamless handover time where MN needs to continue its connectivity while roaming to a heterogeneous network. To preserve MN's anonymity and location privacy in this timely restricted seamless handover, ALPP performs three stages: the setup, AHBU, and ARR. We consider AHBU and ARR as two subschemes because any one of them can be independently implemented in the network.

4.1. Setup

This stage takes place when an MN roams to a foreign network and becomes under an FG's coverage. On the basis of CL-PKC, this FG works as a KGC for the CL-PKC and periodically transmits its identity and its public $Param = \langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_0, H_1, H_2 \rangle$ to network users. The goals of the setup stage are twofold: (1) to mutually authenticate the MN and FG while keeping MN's anonymity and (2) to establish a shared secret key between those two nodes. The exchanged messages shown in Figure 4 illustrate the mutual authentication as well as the key establishment schemes.

The challenge of the mutual authentication scheme is the difficulty to establish trust between two arbitrary nodes, MN and FG, which have not met each other before. The following steps summarize the setup stage where the first three steps achieve the anonymous mutual authentication scheme and the last step achieves the key establishment scheme.

(1) The roaming MN creates a pseudo identity, PID_{MN}, by concatenating its acquired CoA and a time stamp, that is, $PID_{MN} = CoA \parallel t_i$. Furthermore, the MN encrypts the PID_{MN} by using the group key, K_{group}, and sends the encrypted message to the FG as follows:

$$FG \leftarrow MN : \mathrm{Enc}\left(K_{group}, PID_{MN}\right)$$

With this message sent, the FG guarantees that the MN is a legitimate user. Recall that K_{group} is a secret key shared

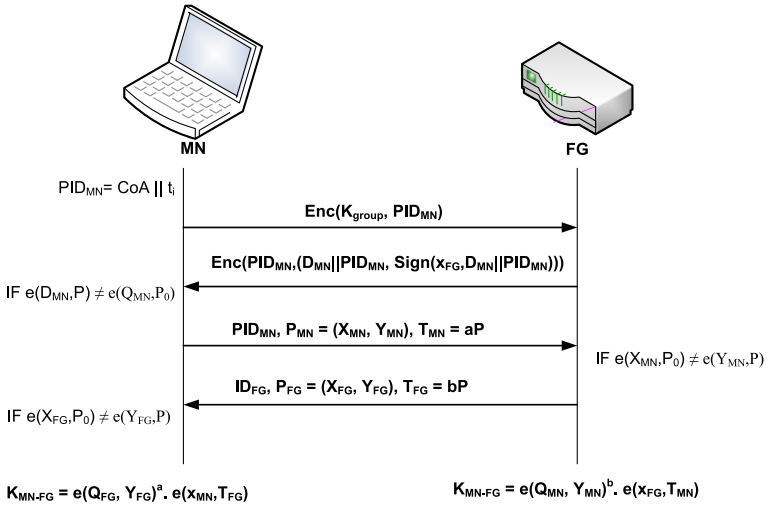

Figure 4. Setup stage.

among all users in the system. The source address of this message is PID_{MN}, and the destination address is the FG's address.

(2) After authenticating the MN as a legitimate user, the FG creates the MN's partial private key, $D_{\mathrm{MN}} = s \times \mathbf{Q}_{\mathrm{MN}}$, where s and $\mathbf{Q}_{\mathrm{MN}}$ are defined in Section 3.3. Furthermore, the FG signs the D_{MN} along with the PID_{MN} and then sends them to the MN after encrypting the whole message by using the MN's pseudo identity, PID_{MN}, as follows:

$$\mathrm{Enc}(PID_{\mathrm{MN}}, (D_{\mathrm{MN}} PID_{\mathrm{MN}}, \mathrm{Sign}(S_{\mathrm{FG}}, D_{\mathrm{MN}} PID_{\mathrm{MN}})) \quad (5)$$

Note that the MN creates different PID_{MN} at each foreign network. The PID_{MN} involves the CoA, which is related to the FG. Therefore, when the MN communicates with a different FG, its CoA changes, and accordingly, the PID_{MN} will be changed. This property increases the MN's anonymity level.

(3) The MN verifies the FG's signature in the received message and then checks the correctness of the received partial private key, D_{MN}, by using the following condition:

$$\text{IF } \hat{e}(D_{\mathrm{MN}}, P) \neq \hat{e}(Q_{\mathrm{MN}}, P_0), \text{wrong } D_{\mathrm{MN}} s$$

After successful verification, the MN generates its public and private keys, P_{MN} and S_{MN}, by using the received partial private key, D_{MN}, as illustrated in (1). When the MN changes its PID_{MN}, the computed public–private key pair will be changed accordingly.

(4) The roaming MN uses the generated public–private key pair to generate a secret key $K_{\mathrm{MN\text{-}FG}}$ shared with its FG as illustrated in Algorithm 2.

Algorithm 2: Mobile node–foreign gateway shared key establishment

Input: PID_{MN}, P_{MN}, P_{FG}
Output: Shared secret key, K_{MN-FG}
$FG \leftarrow MN$:
$PID_{MN}, P_{MN} = (X_{MN}, Y_{MN}), T_{MN} = aP$
if $\hat{e}(X_{MN}, P_0) \neq \hat{e}(Y_{MN}, P)$ **then**
  Return illegal MN
else
  $MN \leftarrow FG$:
  $P_{FG} = (X_{FG}, Y_{FG}), T_{FG} = bP$
  if $\hat{e}(X_{FG}, P_0) \neq \hat{e}(Y_{FG}, P)$ **then**
    Return illegal FG
  else
    at MN:
    $K_{MN-FG} = \hat{e}(\mathbf{Q}_{FG}, Y_{FG})^a . \hat{e}(S_{MN}, T_{FG})$
    at FG: $K_{MN-FG} = \hat{e}(\mathbf{Q}_{MN}, Y_{FG})^b . \hat{e}(S_{FG}, T_{MN})$
  end
end

Note that in the aforementioned steps, both MN and FG authenticate each other. The FG authenticates the MN by both the group key, K_{group}, and the pairing function, $\hat{e}$. In addition, the MN authenticates the FG by verifying FG's signature and checking the correctness of the partial private key that is created by this FG.

4.2. Anonymous home binding update subscheme

The goal of the AHBU subscheme is to add the anonymity and location privacy services to the HBU control messages. The AHBU subscheme involves two main stages: the BU and the BA. In the remainder of the paper,

we consider that the MN's HoA and CoA represent its identity and its current location, respectively.

4.2.1. Anonymous binding update.

In this stage, the roaming MN uses the created shared secret key, $K_{MN\text{-}FG}$, to send anonymous BU messages to its HA, which is located in this MN's home network. As shown in Figure 6, the HBU steps can be summarized as follows:

(1) The roaming MN chooses an IFG, we call it home intermediate foreign gateway (HIFG), from the IFGs list. This HIFG is chosen to be any one of the gateways that are located on the shortest path between the MN's current location and MN HA's address. To choose this HIFG, the MN firstly asks its attached FG to broadcast a route request message to request the shortest routing path to its HA's address. After receiving the route reply message that contains the shortest path, the MN then randomly chooses one gateway from the gateways on the shortest path to be the HIFG. As illustrated later, the MN uses this HIFG as an anonymizer to hide its location from its HA.

(2) The MN creates an updated version of a BU message in which the alternative CoA field contains Enc ($K_{MN\text{-}FG}, PID_{MN}$) instead of a clear form of MN's CoA. The updated BU message contains a clear form of the MN's HoA because, as shown in Figure 5, the BU is encrypted by $K_{MN\text{-}HA}$; therefore, only the MN's HA identifies the MN.

Using the idea of onion routing, the MN then repeatedly encrypts this BU message by using three different keys: (1) the MN's shared key with its HA, $K_{MN\text{-}HA}$; (2) the HIFG's public key, P_{HIFG}; and (3) the MN's shared key with the FG, $K_{MN\text{-}FG}$. The MN then sends this encrypted BU message to its FG by adding the FG's link address to the message's destination MAC address. Figure 5 shows an encrypted BU message when it is transmitted from the MN where the control fields contain the following values:

- Source 1: Enc($K_{MN\text{-}FG}$, PID_{MN})
- Destination 1: HIFG's address
- Source 2: HIFG's address
- Destination 2: HA's address

Note that the source address, source 1, looks like a wrong IPv6 address format; however, thanks to the setup stage, that enables the FG to identify the CoA_{MN}. According to the setup stage, the FG stores a binding between the encrypted address, Enc($K_{MN\text{-}FG}$, PID_{MN}), and COA_{MN} as shown in Table II.

(3) The FG decrypts the received BU message by using its shared key with the MN, $K_{MN\text{-}FG}$, and then sends the decrypted message to the HIFG after adapting the following field:

- Source 1: FG's address

(4) The HIFG decrypts the receiving message by using its public key, P_{HIFG}, and then stores a binding between the encrypted CoA, Enc($K_{MN\text{-}FG}$, PID_{MN}), and the FG's address. Note that PID_{MN} is a concatenation of MN's CoA and a time stamp t_i. Therefore, for any subsequent messages destined to the

Table II. Network bindings.

Entity	Binding(s)
FG	$PID_{MN} \rightarrow CoA_{MN}$, Enc($K_{MN\text{-}FG}$, PID_{MN}) $\rightarrow CoA_{MN}$
HIFG/CIFG	Enc($K_{MN\text{-}FG}$, PID_{MN}) $\rightarrow$ FG's address
HA	$HoA_{MN} \rightarrow$ Enc($K_{MN\text{-}FG}$, PID_{MN}), Enc($K_{MN\text{-}FG}$, PID_{MN}) $\rightarrow$ HIFG's address
CN	$HoA_{MN} \rightarrow$ Enc($K_{MN\text{-}FG}$, PID_{MN}), Enc($K_{MN\text{-}FG}$, PID_{MN}) $\rightarrow$ CIFG's address

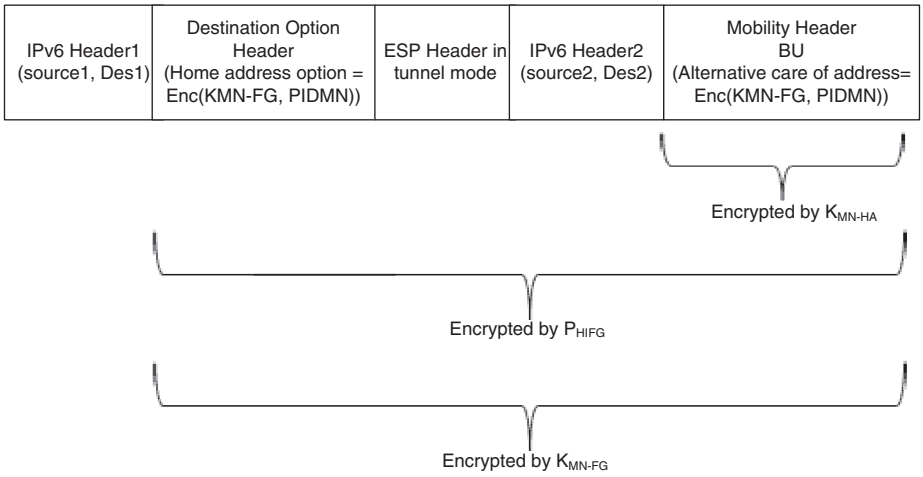

Figure 5. Encrypted binding update message.

encrypted PID_{MN}, the HIFG forwards them to the FG instead. Finally, the HIFG removes the tunneling fields, IPv6 header1 and destination option header, and forwards the remaining message to the address in IPv6 header2, HA's address.

(5) When the HA receives and decrypts the BU message, it contains the following fields:

- Source address: HIFG's address
- Destination address: HA's address
- Alternative CoA: Enc($K_{MN\text{-}FG}$, PID_{MN}).
- HoA destination option: HoA_{MN}

The HA stores a binding between this MN's HoA and the encrypted CoA that represents MN's current location. In this binding, the HA cannot identify the MN's current location because it is an encrypted version of the MN's CoA, Enc($K_{MN\text{-}FG}$, $CoA_{MN}||t_i$). Therefore, the HA stores the HIFG's address as a proxy to reach this encrypted address. Consequently, the HA forwards any subsequent messages, destined to the roaming MN or to the encrypted CoA, to this HIFG's address. Table II shows a summary of stored bindings at each network entity.

4.2.2. Anonymous home binding acknowledgement.

After receiving a BU message, the MN's HA replies by a BA message that is transmitted to the MN. The goal of this message is to inform the MN that the HA creates a binding between the MN's HoA and MN's current location. Therefore, the home binding acknowledgement (HBA) messages complete the mobility management process. As shown in Figure 6, the steps to perform anonymous HBA are as follows:

(1) The HA creates an HBA message as shown in Figure 3, encrypted by the HIFG's public key, and sends it to the HIFG after adding the following fields' values:

- Source 1: HA's address
- Destination 1: HIFG's address
- Source 2: HIFG's address
- Destination 2: Enc($K_{MN\text{-}FG}$, PID_{MN})
- Routing header type 2: Enc($K_{MN\text{-}HA}$, HoA_{MN})

(2) When receiving the HBA message, the HIFG checks its cache memory to identify the corresponding proxy that is attached with the encrypted address, Enc($K_{MN\text{-}FG}$, PID_{MN}). This proxy is the MN's FG; therefore, the HIFG sends the HBA to that FG after encrypting it by using the FG's public key and adapting the following fields:

- Source 1: HIFG's address
- Destination 1: FG's address
- Source 2: FG's address
- Destination 2: Enc($K_{MN\text{-}FG}$, PID_{MN})

(3) The FG decrypts the received message, removes the tunnel fields, IPv6 header1 and routing header type 2, and checks its binding cache to identify the encrypted address Enc($K_{MN\text{-}FG}$, PID_{MN}). The FG then forwards the HBA message to the intended MN's CoA.

4.3. Anonymous return routability subscheme

In mobile IPv4 networking, a roaming MN communicates with a CN by using the reverse tunneling routing method. In this routing, the MN's CoA represents its current location, and the CN does not identify this location. Therefore, instead of sending messages directly to the MN's CoA, the CN transmits these messages to the MN's HA, which eventually forwards the messages to the MN's CoA. This indirectness in routing achieves MN's location privacy because the CN does not realize the MN's movement. However, the reverse tunneling increases the communication routing delay, and it may lead to a triangle routing problem. The worst case of the triangle routing problem occurs when both MN and CN are roaming to the same foreign network. In this case, the CN sends the messages to the MN's HA in home network,

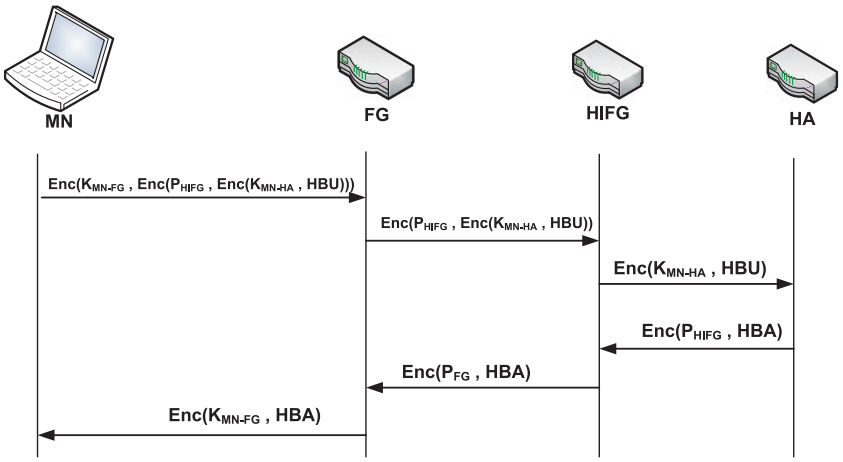

Figure 6. Anonymous home binding update scheme.

which, in turn, forwards the messages again to the same foreign network. This reverse tunneling routing cannot be used with the seamless communications because it increases the handover time and eventually causes a service interruption.

To solve the triangle routing problem, the mobile IPv6 introduces the route optimization routing method. In this routing, the CN identifies the MN's CoA; Therefore, the CN uses the shortest routing path to send messages to the roaming MN. This path is created using the return routability procedure, which is a group of four messages that is exchanged between the MN and the CN. The Home Test Init message (HTIM), the Care-of Test Init message (CTIM), the Home Test message (HTM), and the Care-of Test message (CTM) are the four messages of the return routability procedure. After successful transmission of these messages, the CN creates a binding between the MN's HoA and current location, MN's CoA, so the CN can directly transmit any subsequent messages to the MN's new location. This direct routing method decreases the routing delay; however, it causes an MN's location privacy problem. By monitoring the return routability transmitted messages, the CN as well as an eavesdropper can reveal the MN's anonymity and location privacy.

In this section, the ARR subscheme is proposed to add anonymity and location privacy services to the return routability procedure. In the HTIMs and HTMs, the MN and the CN communicate through the MN's HA (reverse tunneling) to transmit the home keygen token. Similar to BU and BA messages, which are transmitted between MN and HA, the HTIMs and HTMs are transmitted from MN to the HA then to the CN. So, we consider HTIMs and HTMs as BU and BA messages from the transmitted path perspective. Therefore, the AHBU subscheme illustrated in Section 4.2 can be used to add MN's privacy for these two messages. Although the messages' formats are different, we can use the same HIFG to transmit HTIM and HTM from MN to HA. Moreover, in the CTIM and CTM, the care-of keygen token is generated through the direct communication between the MN and the CN. Therefore, the ARR subscheme is proposed to achieve MN's and CN's anonymity and location privacy for both CTIM and CTM transmissions. In the following subsections, two scenarios for the CNs will be presented: a fixed node scenario and a roaming node scenario. In the former scenario, the CN may be a fixed node or an MN that is located in its home network at the time of communication with an MN. In the latter scenario, the CN is an MN that roams to a foreign network.

4.3.1. Fixed correspondent node scenario.

In this scenario, we consider that the MN's and the fixed CN's HoAs are known to each other, However, to achieve location privacy, the MN's current location, CoA_{MN}, is kept unknown to the CN. The ARR scheme consists of two transmitted messages: CTIM and CTM. As shown in Figure 7, the CTIM is transmitted from the MN to the CN. The MN firstly selects an IFG, we call it correspondent intermediate foreign gateway (CIFG). The CIFG is chosen to be located on the shortest path between the MN and the CN. The MN then repeatedly encrypts the message by using three different keys: (1) the public key of the CN's HA, P_{HACN}; 2) the CIFG's public key, P_{CIFG}; and (3) the MN's shared key with its FG, $K_{MN\text{-}FG}$. The MN then sends the encrypted message to the FG in the foreign network, which, in turn, forwards the message to the CIFG, and then the message is forwarded to the CN's HA. Finally, the CN's HA forwards the message to the intended CN.

When receiving the CTM, the CN creates a binding between the MN's HoA and an encrypted version of MN's current address, Enc($K_{MN\text{-}FG}$, PID_{MN}). Furthermore, the CN also stores the address of CIFG as a proxy to reach this encrypted address, Enc($K_{MN\text{-}FG}$, PID_{MN}).

The CN then transmits a CTM to the MN as an acknowledgement for the CTIM. The CN firstly encrypts the CTM by using its shared key with its HA, $K_{HACN\text{-}CN}$, and transmits the encrypted message to its HA. The CN's

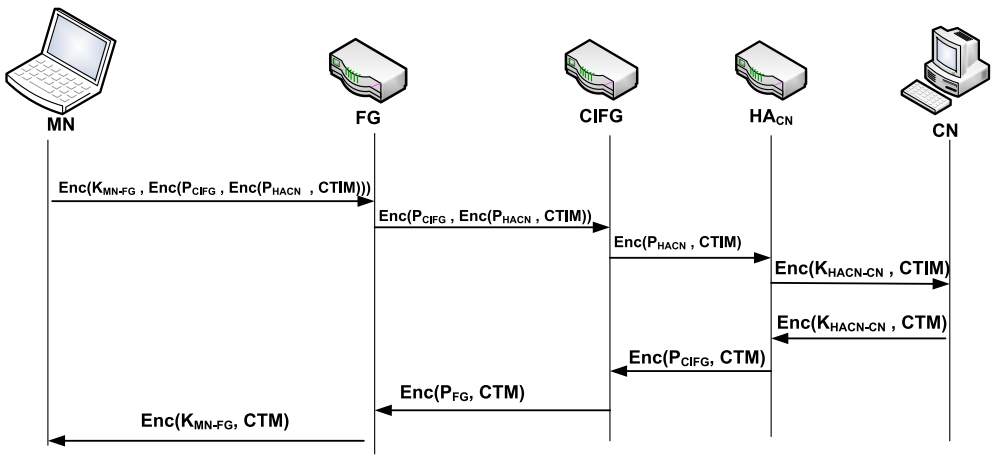

Figure 7. Anonymous return routability, fixed CN.

HA then encrypts the message by CIFG's public key before transmitting it to the CIFG, which, in turn, encrypts and transmits the message to the MN's FG. Finally, the MN's FG encrypts the CTM by using the shared key with that MN, $K_{\text{MN-FG}}$, and then transmits the encrypted CTM to the MN.

4.3.2. Mobile correspondent node scenario.

The mobile CN scenario is more complex than the fixed CN scenario because in this scenario, both the MN and the CN move to two foreign networks. The goal of the ARR scheme here is to achieve MN's and CN's location privacy, which means to hide the two nodes' current locations from each other. Considering an MN as a mobile sender and a CN as a mobile receiver, we here achieve anonymity and location privacy for both mobile senders and mobile receivers.

We consider that both MN's and CN's HoAs are known to each other. As an MN, the CN implements the AHBU scheme, introduced in Section 4.2, to achieve its anonymity and location privacy towards its home network. Furthermore, to achieve the ARR scheme, as shown in Figure 8, the MN sends a CTIM to the CN. Firstly, the CTIM is sent to the CN's HA, which discovers that the CN currently roams to a foreign network. The CN's HA is the one responsible for knowing if CN is fixed or is an MN. As shown in Figure 7, CIFG sends the message to HA_{CN}. If the CN is fixed, the HA_{CN} sends the message to this CN, which is currently located in its network. On the other hand, if the CN roams to a different network, it is assumed that this CN has sent a BU to its HA, HA_{CN}, in an early stage. So, at this time, HA_{CN} forwards the message to $CIFG_{\text{CN}}$, which, in turn, transmits the CTIM to the roaming CN (Figure 8).

On the other way, when the CN sends the CTM to the MN, it is sent directly to the MN's CIFG, $CIFG_{\text{MN}}$. The CTM is not transmitted to the CN's HA because $CIFG_{\text{CN}}$ already knows the $CIFG_{\text{MN}}$'s address; hence, it does not need to ask CN's HA about the $CIFG_{\text{MN}}$'s address. Therefore, the length of the CTM routing path is shorter than the length of the CTIM routing path. The CTM routing path is used for data transmission between roaming MN and CN.

The worst case is when the CN and the MN move to the same foreign network. In this case, the two nodes select either the same FG or different FGs. If both nodes choose the same FG, then only this FG realizes that they are in the same network. Therefore, the FG delivers the messages between the MN and the CN without forwarding them to the corresponding CIFGs. If the two nodes choose two different FGs in the same foreign networks, the MN–CN routing path goes through the corresponding CIFGs, and this leads to high routing delay.

5. PRIVACY AND SECURITY ANALYSIS

5.1. Privacy analysis

In our network, the MN's HoA and CoA represent its identity and its current location, respectively. Therefore, violating an MN's HoA means breaking its anonymity, and violating an MN's CoA means breaking its location privacy.

As in [35], we use the entropy model to measure the degree of anonymity for both our proposed scheme and the mix-based scheme [21]. The degree of anonymity, d, can be measured by the following equation:

$$d = 1 - \frac{H_{\text{M}} - H(X)}{H_{\text{M}}} = \frac{H(X)}{H_{\text{M}}} \tag{6}$$

$H(X)$ is the entropy of the network, which measures the amount of information that an attacker knows about the identity of message's sender. H_{M} is the maximum entropy of the network. Therefore, the degree of anonymity for ALPP scheme can be measured as follows:

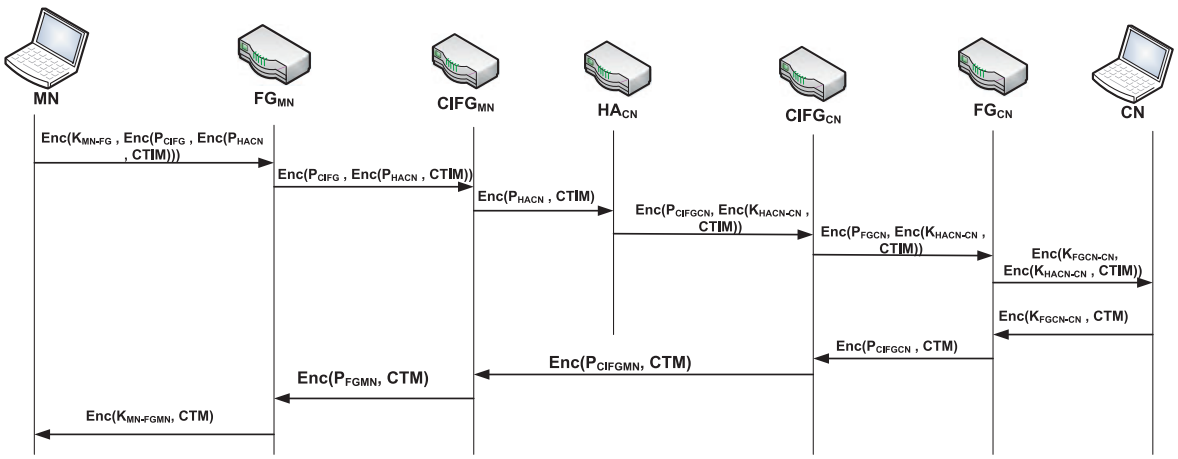

Figure 8. Anonymous return routability, mobile CN.

$$H(X) = \sum_{i=0}^{n}\left[p_i \log\frac{1}{p_i}\right] = \log n$$
$$H_M = \sum_{i=0}^{L.n}\left[p_i \log\frac{1}{p_i}\right] = \log(L.n) \quad (7)$$
$$d = \frac{\log n}{\log(L.n)}$$

where p_i is the probability that a node i is the sender of a message, n is the number of nodes in the home network, and L is the number of networks in the system.

Similarly, the degree of anonymity for the mix-based scheme can be computed as follows:

$$d = \begin{cases} \dfrac{\log m}{\log(L.n)}, & K = 1 \\ \dfrac{\log(K.m)}{\log(L.n)}, & K > 1 \end{cases}$$

where K is the number of mix servers , L is the number of networks in the system, and m is the number of messages that are mixed together at each mix server. The number of mixed messages is an indicator for the number of senders because in mix-based scheme, each sender sends a message at a time to the mix server. Therefore, m also represents the number of senders in the network.

Figure 9 shows the degree of anonymity for our scheme at different values of L and for the mix-based scheme with one mix server ($K=1$). ALPP's degree of anonymity increases as the number of nodes in the home network increases; however, it decreases as the number of networks in the system increases. On the other hand, the degree of anonymity for the mix-based scheme increases as the number of senders increases. For the mix-based scheme, we fix the number of users in one network to be 1000 users. Therefore, for $L=10$, the total number of users is 10,000.

Compared with our proposed scheme, ALPP, the mix-based scheme with one mix server achieves lower level of anonymity when the number of senders is below 1000. Increasing the number of senders in mix-based scheme causes a high delay, as it will be seen later. Moreover, increasing the number of mix servers leads to increasing the level of anonymity; however, it also increases the network delay. This trade-off prevents the mix-based scheme to be used for seamless communications, which require low routing delay to achieve service continuity.

To illustrate the impact of delays on the mix-based scheme, Figure 10 shows the delay of the scheme multiplied by the achieved degree of anonymity. Considering 2 ms for the mix server to send and receive a message, the mix-based scheme with one mix server requires around 1.2 s to serve 1000 senders. This delay increases to around 5 s with increasing the number of mix servers. To achieve higher anonymity by using one mix server, the number of senders, m, that sends messages to this mix server should be increased. For one mix server, m ranges from zero to the total number of users in the system ($L.n$). However, the network delay increases as m increases because the mix server needs to wait until receiving all messages from all senders then mixes and retransmits them. Alternatively, the anonymity level can be increased when the number of mix servers, K, increases. In this case, the number of senders, m, is limited to $0 \le m \le \frac{L.n}{K}$. However, the network delay also increases when the number of mix servers increases because these mix servers work in sequential with each other. As a conclusion, in mix-based scheme, there is a trade-off between the achieved anonymity level and the network delays.

On the other hand, Figure 11 shows the delay of the ALPP scheme multiplied by its degree of anonymity. Compared with the mix-based scheme, our scheme has a delay of 1.5 ms to serve 1000 users, which is 99% less than the mix-based scheme's delay.

The proposed ALPP scheme achieves sender's and receiver's locations privacy by hiding their CoAs from both the HA and the CN. In our network, the CoA and the FG represent a node's location information. The MN's HA cannot determine the MN's CoA because it receives an encrypted address, Enc($K_{MN\text{-}FG}$,PID_{MN}), instead of a plaintext address. Moreover, the HA does not

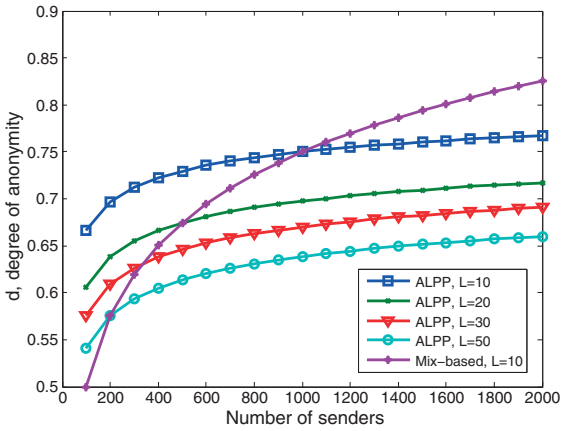

Figure 9. Degree of anonymity.

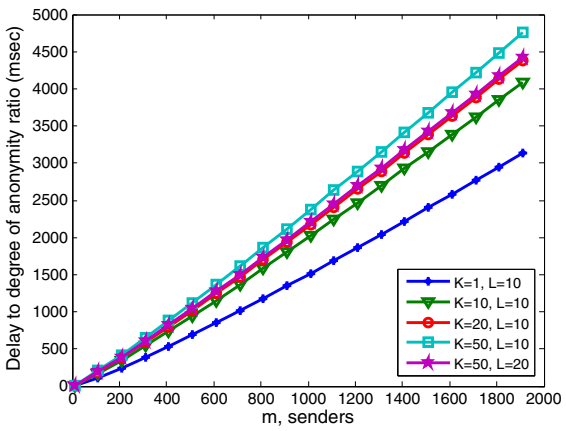

Figure 10. Mix-based's delay-to-degree of anonymity ratio.

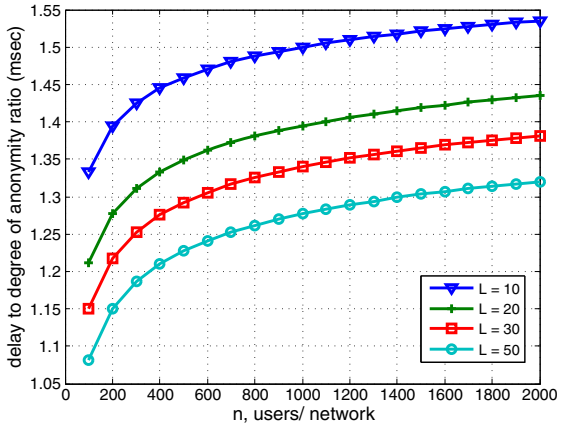

Figure 11. ALPP's delay-to-degree of anonymity ratio.

communicate directly with the FG; they communicate through a proxy, the IFG. Therefore, the HA cannot identify the MN's FG.

Table III shows the MN's information that each entity in the network can acquire. In the table, the header column represents network entities, the header row represents MN's information, K means that the network entity knows the information part, and U means that the information is unknown to the network entity. As it is shown in the table, no network entity except the MN itself can identify this node's location, CoA. Two columns in the table represent MN's location information: CoA and FG columns. The CoA column shows that no entity knows the MN's CoA except this MN and its FG. Although FG specifies MN's location, it does not identify this MN because the MN communicates with the FG by a mean of pseudo identity instead of its real identity. In addition, the FG column shows that the MN, FG, and IFG know MN's FG. The IFG only specifies MN's FG, but it does not identify the MN itself.

5.2. Security analysis

The security of ALPP scheme is based on the security of the proposed key establishment scheme that is illustrated in Algorithm 2. Moreover, the security of the key establishment scheme is based on the hardness of the elliptic curve discrete logarithm problem (ECDLP). In [36], it is proved that ECDLP can be solved in at least subexponential time. ECDLP is a hard problem because there is no polynomial time algorithm that can solve it.

Table III. Mobile node's information knowledge.

	HoA	CoA	HA	FG	CN	IFG
MN	K	K	K	K	K	K
HA	K	U	K	U	K	K
FG	U	K	K	K	U	K
IFG	U	U	K	K	K	K
CN	K	U	K	U	K	K
Adversary	U	U	U	U	U	K

Definition 1. *The elliptic curve discrete logarithm problem (ECDLP) Given P and xP as two points on elliptic curve E, find x, where* $x \in \mathbb{Z}_q^*$.

Theorem 1. *In Algorithm 2, under the assumption that an attacker knows the MN's private key,* S_{MN}*, the attacker is still unable to create the shared key* $K_{MN\text{-}FG}$.

Proof. To create a valid $K_{\text{MN-FG}}$, the attacker needs to compute the following pairing functions:

$$K_{\text{MN-FG}} = \hat{e}(\mathbf{Q}_{\text{FG}}, Y_{\text{FG}})^a.\hat{e}(S_{\text{MN}}, T_{\text{FG}})$$

Because the attacker knows S_{MN}, it easily computes $\hat{e}(S_{\text{MN}}, T_{\text{FG}})$. However, to compute $\hat{e}(\mathbf{Q}_{\text{FG}}, Y_{\text{FG}})^a$, the attacker needs to know the value of a. But the attacker knows only P and $T_{\text{MN}} = aP$. Then this problem is equivalent to ECDLP. Because ECDLP is a hard problem, the attacker cannot create a valid $K_{\text{MN-FG}}$ in a polynomial time.

5.2.1. The traffic analysis attack.

The traffic analysis attacker attempts to capture a group of the transmitted packets and analyze them in order to learn the identity and the location of the MN. The identity of the MN, which is represented by its HoA, is transmitted in an encrypted form. Therefore, the traffic analysis attacker cannot learn the true identity of the MN. Moreover, We use onion routing to prevent the attacker from correlating the input and output messages at a specific hop. For example, the BU messages that are transmitted from an MN are repeatedly encrypted by three different keys: the shared key with the HA, the IFG's public key, and the shared key with the FG. When the FG receives these messages, it decrypts them using the shared key with the MN, and then retransmits the decrypted messages to the IFG. These decrypted messages are indeed encrypted messages by the remaining two keys. Therefore, at each hop, the messages are decrypted by one key then retransmitted to the second hop. Consequently, the attacker cannot identify the MN's movements.

5.2.2. The collusion attack.

The collusion attack may be triggered among the FGs, the IFGs, or the CNs. When our proposed schemes are used, a collusion attacker gains no information about the MN's identity and locations.

If the FGs collude with each other, they would not learn the identity of the MN. In the setup stage, the MN uses a pseudo identity, $PID_{\text{MN}} = CoA \parallel t_i$, to identify itself to the FG. The MN's CoA, which is used to create the PID_{MN}, changes as the MN chooses different FG; hence, MN's PID_{MN} also changes. Therefore, each FG identifies only one PID_{MN} of the MN's pseudo identities. Therefore, it is not possible for the FGs to link all the CoAs to the same MN.

Moreover, the collusion of the IFGs reveals nothing about MN's privacy because they do not directly

communicate with this MN. In our network, IFGs only communicate directly with the HA and the FG. The IFG received an MN's encrypted CoA, which represents the MN's location. Again, when it roams among different foreign networks, the MN acquires different CoAs and encrypts them by different keys. Therefore, if IFGs collude, they cannot link all encrypted CoAs to the same MN. Furthermore, the collusion among the FGs and the IFGs reveals the MN's HA. The knowledge of the MN's HA does not reveal the MN's privacy because we argue that there is at least two nodes in the home network. Therefore, the probability of identifying the MN is as follows:

$$P(MN) = \frac{1}{n}, n \geq 2 \tag{8}$$

where n is the number of nodes in the home network. Therefore, for a large number of nodes located in the MN's home network, the probability of identifying the MN after identifying its network is negligible.

5.2.3. The replay attack.

In the setup stage, an attacker may send a previously transmitted pseudo identity to the FG in order to deceive the FG and learn the MN's partial private key, D_{MN}. In our proposed schemes, the MN's pseudo identity, $PID_{\mathrm{MN}} = CoA \,||\, t_i$, is created by concatenating MN's CoA with the time stamp. The time stamp prevents an attacker from repeating transmission of previous messages. However, any legitimate user who knows the group key can decrypt the message, change the time stamp, and then resend the message again. From Theorem , we prove that even if a legitimate user succeeds to learn the MN's secret key, this user is still unable to create a valid shared key, $K_{\mathrm{MN\text{-}FG}}$.

5.2.4. The man-in-the-middle attack.

A man-in-the-middle attacker may change either MN's identity, PID_{MN}, or public key, P_{MN}, to create a fake session with the FG. We prove by Theorem that if either PID_{MN} or P_{MN} is changed in the middle of transmission, then the key generation algorithm returns "illegal MN".

Theorem 2. *If either PID_{MN} or $P_{MN} = (X_{MN}, Y_{MN})$ is changed by an attacker, then Algorithm 2 returns "illegal MN".*

Proof.

Case 1 If PID_{MN} is changed to PID_{MN}, then from Theorem , an attacker cannot create $K_{\mathrm{MN\text{-}FG}} = \hat{e}(\mathbf{Q}_{\mathrm{FG}}, Y_{\mathrm{FG}})^{a}.\hat{e}(S_{\mathrm{MN}}, T_{\mathrm{FG}})$ because the attacker does not know the values of a and S_{MN}. Then attacker is an illegal MN.

Case 2 If P_{MN} is changed to $P_{\mathrm{MN}} = (X_{\mathrm{MN}}, Y_{\mathrm{MN}})$, then the condition at line 2 of Algorithm 2 is satisfied. This means $\hat{e}(X_{\mathrm{MN}}, P_0) \neq \hat{e}(Y_{\mathrm{MN}}, P)$ Then Algorithm 2 returns "illegal MN".

In addition, an man-in-the-middle attacker may send a fake partial private key, D_{MN}, to the MN in the setup stage. This case also happens if the FG is a malicious node and wants to mislead the MN. The result of this attack leads to an interruption of the MN's IP session. However, in our proposed schemes, the MN authenticates the FG by verifying its signature as it is illustrated in the setup stage. Moreover, the MN also checks the correctness of the partial private key that is received from the FG, using the following condition:

$$\text{IF } \hat{e}(D_{\mathrm{MN}}, P) \neq \hat{e}(\mathbf{Q}_{\mathrm{MN}}, P_0), \text{wrong } D_{\mathrm{MN}}$$

We can show that for a correct D_{MN}, the two pairing functions are identical, as follows:

$$\begin{aligned} \hat{e}(D_{\mathrm{MN}}, P) &= \hat{e}(s \times \mathbf{Q}_{\mathrm{MN}}, P) \\ &= \hat{e}(\mathbf{Q}_{\mathrm{MN}}, s \times P) \\ &= \hat{e}(\mathbf{Q}_{\mathrm{MN}}, P_0) \end{aligned} \tag{9}$$

6. PERFORMANCE EVALUATION

6.1. Computation and communication overhead

Tables IV and V show the computation and communication overheads of the proposed subschemes, AHBU and ARR, compared with those of mix-based scheme [21]with one mix server and the EHoA and PHoA schemes [23]. In addition, we use Cryptoo++ benchmarks [37] to measure the computation time at the MN's side as it is shown in Figure 12. We use ElGamal encryption mechanism for public key encryption operations and Advanced Encryption Standard (AES) scheme for symmetric encryptions. Therefore, in the tables, T_{ElG} represents the time needed for ElGamal encryption operation, T_{Sym} represents the time needed for AES encryption or decryption, T_{pid} and T_{prf} represent the time needed to construct a pseudonym and to generate a random number, and $T_{\mathrm{EHoA\text{-}reg}}$ and $T_{\mathrm{PHoA\text{-}reg}}$ represent the time needed for registering the EHoAs and the PHoAs. For computation overheads, $B_{\mathrm{signaling}}$ represents the bytes needed to send the control information, and $B_{\mathrm{EHoA\text{-}reg}}$ and $B_{\mathrm{PHoA\text{-}reg}}$ represent the bytes needed to send PHoA and EHoA registration messages.

Table IV. AHBU computation and communication overheads.

	Computation	Communication
Mix-based	$3T_{\mathrm{ElG}} + 2T_{\mathrm{Sym}} + 2T_{\mathrm{prf}} + T_{\mathrm{pid}}$	$B_{\mathrm{signaling}}$
AHBU	$T_{\mathrm{ElG}} + 3T_{\mathrm{Sym}}$	$B_{\mathrm{signaling}}$

Table V. ARR computation and communication overheads.

	Computation	Communication
Mix-based	$3T_{\mathrm{Sym}} + T_{\mathrm{Hash}} + 2T_{\mathrm{pid}}$	$B_{\mathrm{signaling}}$
EHOA	$3T_{\mathrm{Sym}} + T_{\mathrm{EHoA\text{-}reg}}$	$B_{\mathrm{EHoA\text{-}reg}}$
PHOA	$T_{\mathrm{pid}} + 2T_{\mathrm{PHoA\text{-}reg}}$	$B_{\mathrm{PHoA\text{-}reg}}$
ARR	$2T_{\mathrm{Sym}} + 2T_{\mathrm{ElG}}$	$B_{\mathrm{signaling}}$

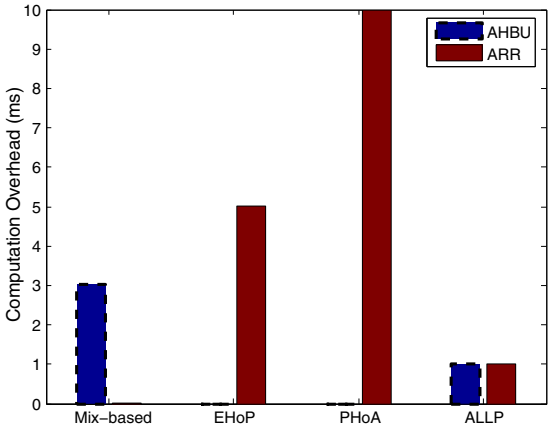

Figure 12. ALPP computation overhead.

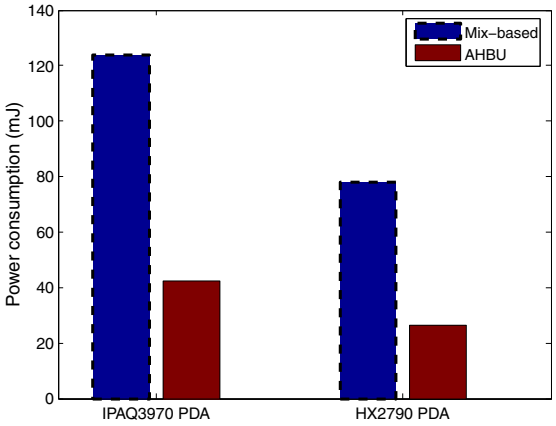

Figure 13. AHBU power consumption.

In Table IV, our AHBU's computation overhead is smaller than mix-based scheme's overhead by 66%. The mix-based scheme requires three public key encryption operations, whereas the AHBU scheme requires only one public key operation.

Table V shows that the ARR subscheme is the second smallest time-consuming scheme after the mix-based scheme. In EHoA and PHoA schemes, an MN needs first to register the EHoA and PHoA before using them. Considering 5 ms for one round-trip time between the MN and its HA, the computation overheads of EHoA and PHoA schemes are much higher than that of the ARR subscheme. ARR's computation overhead is smaller than EHoA's and PHoA's overheads by 79% and 89%, respectively. Figure 12 shows the time consumption for ALPP scheme compared with other schemes.

The measured AHBU and ARR computation overheads do not include the time required for the setup stage, T_{setup}, because this time is only needed once as long as an MN stays in one foreign network. If an MN sends many HBU messages from the same foreign network, then only one T_{setup} is required. The setup time can be measured as follows:

$$T_{\text{setup}} = 2T_{\text{Sym}} + T_{\text{verification}} + 3T_{\text{pairing}} \tag{10}$$

Considering AES mechanism for T_{Sym} and RSA for signature verification time, $T_{\text{verification}}$, the estimated time needed for T_{setup} is around 120 ms. To measure the pairing time, T_{pairing}, we consider a 2.93-GHz processor with the Tate pairing in [38] and obtain 6.83 ms for each pairing function.

6.2. Power consumption

Aiming to compute the energy consumed at MN, we follow the energy costs of cryptographic algorithms that are proposed in [39] for two different Personal Digital Assistants (PDAs), iPAQ3970 from Compaq company in Houston, Texas, USA and Hx2790 from HP company in United Kingdom. As shown in Figure 13, compared with the mix-based scheme, the AHBU subscheme has the lowest energy consumptions for both PDA types. AHBU achieves energy reductions of 65.66% and 66% when using Compaq iPAQ3970 and HP Hx2790, respectively. This is due to using only one public key operation, whereas mix-based scheme uses three public key operations. According to [39], one public key scheme requires 40.87 and 25.87 mJ to encrypt a message in iPAQ3970 and HP Hx2790, respectively.

6.3. Simulation results

On the basis of the anonymizer mechanism [25], we have proposed a new method of routing in which the transmitted BU message is sent to an intermediate node, IFG, instead of sending it to the receiver directly. The selected HIFG works as an anonymizer. Unlike the traditional anonymizer, which is a fixed proxy that serves all nodes and can easily reveal MNs' privacy, our anonymizer changes with each MN, and it cannot reveal the privacy information.

We develop a simulator to compare the effect of the updated routing method that is used by both AHBU and ARR subschemes with that of the original routing method that does not achieve any MN's privacy.

Two kinds of MNs are defined in our simulator. The first type, called the successful node, is the node that succeeds to find an IFG on the shortest path between the communicating parties. The second type, called the failed node, is the node that moves to a neighbor network, so the shortest path length is only one hop, from HA to FG. Therefore, the failed node cannot find an intermediate gateway on the shortest path.

We consider 351 simulation runs where the number of nodes in the system increases from 1000 nodes, in the first run, to 36,000 nodes, in the last run. We consider large number of nodes in order to check the scalability of our proposed scheme. At each run, the maximum node speed ranges from 2 to 20 m/s. The time interval between each run is 10 min. We use the Bellman–Ford routing algorithm for messages routing among gateways. Table VI shows the full simulation parameters.

Table VI. Simulation parameters.

Parameter	Value
System size	5500 m × 5500 m
Network numbers in system	36
Network size	1000 m × 1000 m
Number of nodes per system	1000–36,000 nodes
Overlapping area	100 m
Distribution of nodes	Uniform
Mobility model	Random waypoint model
Nodes maximum speed range	2–20 m/s
Nodes minimum speed	0 m/s
Number of HA per network	one

Figure 14 shows the routing delays of the proposed subschemes, compared with the HBU scheme and the triangle routing that is used by the mobile IPv4 protocol. HBU and triangle routing schemes are used as lower and upper references, respectively. In this figure, we measure the routing delay for a high-density networking, 36,000 nodes density.

As shown in the figure, the proposed subschemes, AHBU, ARR subscheme with fixed CN scenario, and ARR subscheme with mobile CN scenario, have very similar routing delays as the HBU's routing delay. The HBU scheme does not apply any anonymity or location privacy services. This result indicates the ability of using our proposed schemes with scalable networks and real-time applications in which the routing delay is an important factor. The reported difference in the routing delays, between our proposed schemes and the HBU, results from the failed nodes. In our simulation, the failed nodes do not apply our updated routing method because there is no IFG on the shortest path. An alternative solution is that the failed nodes can select any IFG at an adjacent network. In this case, routing delay values may depend on the network traffic because the adjacent network is not located on the shortest path of the two communicating parties.

We also notice that the routing delay of the triangle routing method is larger than our schemes' delays. The triangle routing method achieves an MN's location privacy; however, its high delay prevents it to be used for seamless communications. Our subschemes' routing delays are smaller than the triangle routing delay by an average of 32%.

Figure 15 shows the network routing delay for different network capacities at high node's mobility, 20 m/s. It can be seen that the number of nodes in the network does not have a significant impact on the routing delay. However, the nodes' mobility speed has a large impact on this delay. Compared with the HBU scheme, our proposed subschemes, the AHBU, ARR-fixed CN, and ARR-mobile CN schemes, increase the routing delays by 2.7%, 4%, and 20%, respectively. On the other hand, compared with the triangle routing scheme, our subschemes decrease the routing delays by 42% for AHBU subscheme, 43% for ARR-fixed CN, and 30% for ARR-mobile CN.

Figure 16 shows the number of successful and failed nodes with a 36,000-node system and at different nodes' speeds. It can be seen that the number of failed and successful nodes depends on the speed of the node. With mobility speeds below 8 m/s, the number of the successful nodes increases as the nodes' speeds increase. However, with mobility speeds above 8 m/s, the numbers of successful and failed nodes are fixed. This result confirms that our schemes are more appropriate to be used at high-mobility environments.

Additionally, we obtain the 95% confidence intervals for both the successful nodes' numbers and the average routing delay. Table VII shows the confidence intervals with different system densities and mobility speeds, in which we consider low density as 1000 nodes and high density as 36,000 nodes. Similarly, we consider low mobility as 2 m/s and high mobility as 20 m/s.

7. CONCLUSIONS AND FUTURE WORK

In this paper, on the basis of the onion routing, anonymizer, and CL-PKC, we have proposed the ALPP scheme and

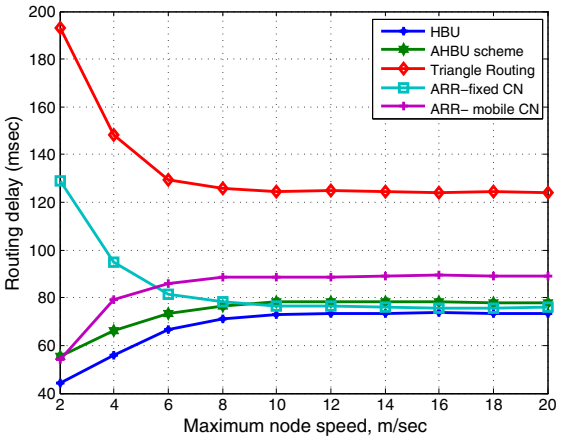

Figure 14. Routing delay at different mobility speeds.

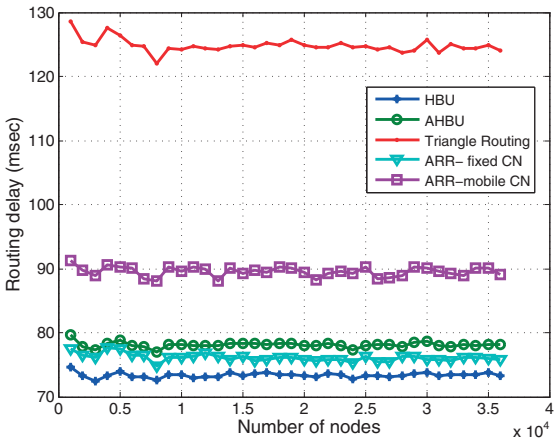

Figure 15. Routing delay with different network capacity.

DOI: 10.1002/sec

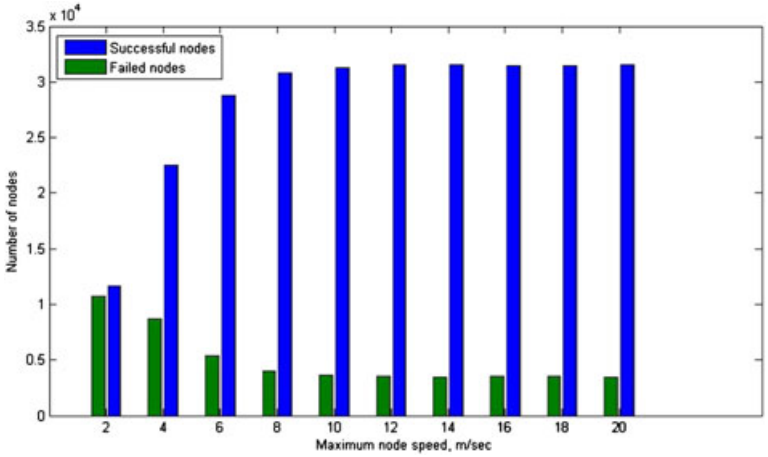

Figure 16. Successful and failed nodes at mobility speeds.

Table VII. 95% confidence interval (CI) of the AHBU subscheme.

	Density	Mobility	Mean	St. Dev	CI
Successful nodes numbers	Low	Low	851.0	45.96	[847.0, 855.0]
	Low	High	862.5	10.53	[861.6, 863.5]
	High	Low	30845	1570	[30708, 30983]
	High	High	31155	108.2	[31146, 31165]
Average delay (ms)	Low	Low	93.92	2.490	[93.70, 94.14]
	Low	High	93.80	1.100	[93.70, 93.90]
	High	Low	93.97	1.970	[93.79, 94.14]
	High	High	94.08	0.200	[94.06, 94.10]

its two complementary subschemes, AHBU and ARR. In addition, we have introduced a mutual authentication scheme as well as a key establishment scheme to be used among arbitrary nodes. Compared with existing anonymity and location privacy schemes, ALPP achieves higher level of anonymity and location privacy for both mobile senders and receivers. Moreover, AHBU and ARR subschemes require less computation overheads than existing schemes. Therefore, our scheme can be implemented for heterogeneous networks where the time of the seamless handover is limited.

In our future work, a mechanism to reduce the time for the setup stage will be designed. We will try to reduce the number of pairing operations that are used to authenticate an MN to its FG. Alternatively, we will try to delete the verification operation in which an MN verifies the FG's signature. Moreover, the proposed scheme will be implemented for different types of mobility management protocols, such as proxy mobile IP and hierarchical mobile IP protocols.

ACKNOWLEDGEMENTS

This research is funded by the University of Waterloo, Canada, and the Egyptian Bureau of Cultural and Education Affairs in Canada.

REFERENCES

1. Taha S, Shen X. Anonymous home binding update scheme for mobile IPv6 wireless networking. *IEEE Global Telecommunications Conference (GLOBECOM 2011)*, Houston, Texas, USA, 2011; 1–5.
2. Shen C, Du W, Atkinson R, Irvine J. A mobility framework to improve heterogeneous wireless network services. *International Journal of Ad Hoc and Ubiquitous Computing* 2011; **7**(1):60–69.
3. Céspedes S, Shen X, Lazo C. IP mobility management for vehicular communication networks: challenges and solutions. *IEEE Communications Magazine* 2011; **49** (5):187–194.
4. Magagula L, Chan H, Falowo O. Achieving seamless mobility through handover coordination in a network-based localized mobility managed heterogeneous environment. *Proc. of IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Istanbul, Turkey, 2010; 2505–2510.
5. Kim KY, Ham KG, Cho KS, Choi SG. Seamless handover scheme based on SIP in wireless LAN. *Proc. of*

13th International Conference on Advanced Communication Technology (ICACT), Korea, 2011; 847–851.
6. Akyildiz I, Xie J, Mohanty S. A survey of mobility management in next-generation all-IP-based wireless systems. *IEEE Wireless Communications* 2004; **11** (4):16–28.
7. Jonhson D, Perkins C, Arkko J. Mobility support in IPv6. *Internet Engineering Task Force, IETF RFC 3775*, 2004. URL www.ietf.org/rfc/rfc4882.txt
8. Taha S, Céspedes S, Shen X. EM3A: efficient mutual multi-hop mobile authentication scheme for PMIP networks. *Proc. of IEEE ICC 2012*, Ottawa, Canada, 2012; to appear.
9. Taleb T, Letaief K. A cooperative diversity based handoff management scheme. *IEEE Transactions on Wireless Communications* 2010; **9**(4):1462–1471.
10. Kavitha D, Murthy K, ul Huq S. Security analysis of binding update protocols in route optimization of MIPv6. *Proc. of International Conference on Recent Trends in Information, Telecommunication and Computing (ITC)*, Kochi, Kerala, India, 2010; 44–49.
11. Ying Q, Feng B. Authenticated binding update in mobile IPv6 networks. *Proc. of 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, Chengdu, China, 2010; 307–311.
12. Lu R, Lin X, Zhu H, Ho P, Shen X. A novel anonymous mutual authentication protocol with provable link-layer location privacy. *IEEE Transactions on Vehicular Technology* 2009; **58**(3):1454–1466.
13. Lu R, Lin X, Luan T, Liang X, Shen X. Pseudonym changing at social spots: an effective strategy for location privacy in VANETs. *IEEE Transactions on Vehicular Technology* 2012; **61**(1):89–96.
14. Krontiris I, Freiling F, Dimitriou T. Location privacy in urban sensing networks: research challenges and directions [security and privacy in emerging wireless networks]. *IEEE Wireless Communications* 2010; **17** (5):30–35.
15. Whalen T. Mobile devices and location privacy: where do we go from here? *IEEE Security and Privacy* 2011; **9**(6):61–62.
16. Thomson S, Narten T. IPv6 stateless address autoconfiguration. *Internet Engineering Task Force, IETF RFC 2462*, 1998. URL www.ietf.org/rfc/rfc2462.txt
17. Droms R, Bound J, Volz B, Lemon T, Perkins C, Carney M. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). *Internet Engineering Task Force, IETF RFC 3315*, 2004. URL: www.ietf.org/rfc/rfc3315.txt
18. Wiangsripanawan R, Safavi-Naini R, Susilo W. Location privacy in mobile IP. *Proc. of 7th IEEE Malaysia International Conference on Communication., Proc. of 13th IEEE International Conference on Networks.*, vol. **2**, 2005; 1120–1125.
19. Fasbender A, Kesdogan D, Kubitz O. Analysis of security and privacy in mobile IP. *Proc. of 4th International Conference on Telecommunication Systems, Modeling and Analysis*, Nashville, TN, USA, 1996; 1–17.
20. Escudero-Pascual A, Hdenfalk M, Heselius P. Flying freedom: location privacy in mobile internetworking. *Proc. of INET2001*, Stockholmsmässan - Stockholm, Sweden, 2001; 25–32.
21. Jiang J, He C, ge Jiang L. A novel mix-based location privacy mechanism in mobile IPv6. *Computers and Security* 2005; **24**(8):629–641.
22. Koodli R. IP address location privacy and mobile IPv6: problem statement. *Internet Engineering Task Force, IETF RFC 4882*, 2007. URL www.ietf.org/rfc/rfc4882.txt
23. Koodli R, Zhao F, Qiu Y. Mobile IPv6 location privacy solutions. *Internet Engineering Task Force, IETF RFC 5726*, 2010. URL www.ietf.org/rfc/rfc5726.txt
24. Goldschlag D, Reed M, Syverson P. Onion routing for anonymous and private Internet connections. *Communications of the ACM* 1999; **42**:39–41.
25. The Anonymizer. URL http://www.anonymizer.com/
26. Al-riyami SS, Paterson KG, Holloway R. Certificateless public key cryptography. *Advances in Cryptology-ASIACRYPT* 2003; **2894**:452–473.
27. Chaum D. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM* 1981; **24**(2):84–90.
28. Dingledine R, Mathewson N, Syverson P. TOR: the second-generation onion router. *Proc. of the 13th Conference on USENIX Security Symposium*, vol. **13**, San Diego, CN, USA, 2004; 21–37.
29. Reiter M, Rubin A. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security (TISSEC)* 1998; **1**(1):66–92.
30. Choi S, Kim K, Kim B. Practical solution for location privacy in mobile IPv6. In *Information Security Applications*. Lecture Notes in Computer Science, Vol. **2908**, Chae KJ, Yung M (eds). Springer: Berlin/ Heidelberg, 2004; 1965–1976.
31. So-In C, Jain R, Paul S, Pan J. Virtual ID: a technique for mobility, multi-homing, and location privacy in next generation wireless networks. *Proc. of 7th IEEE Consumer Communications and Networking Conference (CCNC)*, 2010; 1–5.
32. Nikander P, Gurtov A, Henderson T. Host Identity Protocol (HIP): connectivity, mobility, multi-homing, security, and privacy over IPv4 and IPv6 networks. *IEEE Communications Surveys Tutorials* 2010; **12** (2):186–204.
33. Kaufman C, Hoffman P, Nir Y, Eronen P. Internet Key Exchange Protocol version 2 (IKEv2). *Internet Engineering Task Force, IETF RFC 5996*, 2010. URL http://tools.ietf.org/html/rfc5996

34. Lee E, Lee HS, Park CM. Efficient and generalized pairing computation on Abelian varieties. *IEEE Transactions on Information Theory* 2009; **55** (4):1793–1803. doi:10.1109/TIT.2009.2013048.
35. Diaz C, Seys S, Claessens J, Preneel B. Towards measuring anonymity. In *Privacy Enhancing Technologies*. Springer, 2003; 184–188.
36. Diem C. On the discrete logarithm problem in elliptic curves. *Technical Report*, 2009. URL http://www.mathematik.uni-leipzig.de/MI/diem/preprints/ind-cal-ell-curves.pdf
37. Dai W. Crypto++ 5.6. 0 benchmarks. URL http://www.cryptopp.com/benchmarks.html.
38. Tate pairing. URL http://www.www.shamus.ie/index.php
39. Rifa-Pous H, Herrera-Joancomarti J. Computational and energy costs of cryptographic algorithms on handheld devices. *Future Internet* 2011; **3**(1):31–48.