A Novel Anonymous Mutual Authentication Protocol With Provable Link-Layer Location Privacy

Rongxing Lu, Xiaodong Lin, *Student Member, IEEE*, Haojin Zhu, Pin-Han Ho, *Member, IEEE*, and Xuemin (Sherman) Shen, *Senior Member, IEEE*

***Abstract*—Location privacy of mobile users (MUs) in wireless communication networks is very important. Ensuring location privacy for an MU is an effort to prevent any other party from learning the MU's current and past locations. In this paper, we propose a novel anonymous mutual authentication protocol with provable link-layer location privacy preservation. We first formulate the security model on the link-layer, forward-secure location privacy, which is characterized by the fact that even when an attacker corrupts an MU's current location privacy, the attacker should be kept from knowing how long the MU has stayed at the current location. Then, based on the newly devised keys with location and time awareness, a novel anonymous mutual authentication protocol between the MUs and the access point (AP) is proposed. To the best of our knowledge, this is the first developed anonymous mutual authentication scheme that can achieve provable link-layer, forward-secure location privacy. To improve efficiency, a *Preset in Idle* technique is exercised in the proposed scheme, which is further compared with a number of previously reported counterparts through extensive performance analysis.**

***Index Terms*—Anonymous mutual authentication, link layer, provable forward-secure location privacy.**

I. Introduction

The IEEE 802.11 (or Wi-Fi)-based wireless access points (APs), which are also known as public *wireless local area networks* (WLANs), have been widely available in heavily populated regions, such as airports, restaurants, cafes, libraries, and hotels. It is envisioned that 802.11-based WLANs will be interconnected as wireless mesh networks and will possibly form the backbone of future wireless metropolitan-area networks (WMANs). The emergence of many mission-critical and personalized applications, which run on the pervasive mobile stations with increasing strength, functionalities, and long-lasting power, further magnifies the importance and nonreplaceability of the wireless Internet access.

One of the most critical requirements for the success of metropolitan-area wireless communications networks is in the aspect of secure communication and location privacy protection. It is well known that messages sent over the wireless channels, due to their inherent broadcast nature of wireless signals, are vulnerable to the following three types of security threats: eavesdropping, modification, and impersonation. Extensive efforts have been made by both industry and academia on developing trusted security mechanisms and protocols to ensure secure communications over wireless networks. Basically, wireless security can be categorized into two aspects: authentication and confidentiality. In the conventional Internet, an *access control list* (ACL) has been devised in the media access control (MAC) layer for providing authentication, whereas 802.11 *Wired Equivalent Privacy* (WEP) is used to achieve confidentiality. However, due to the intrinsic vulnerabilities in wireless networks, an MAC spoofing attack can easily be launched, by which a malicious party can easily impersonate another user and/or intercept a legitimate connection between two parties, whereas WEP has been proven to be insecure and vulnerable due to its weak static shared secret key between the AP and the mobile users (MUs). Recently, *Wi-Fi Protected Access* has been proposed to increase the level of data protection and access control for existing and future WLAN-based communication systems.

Due to the fast booming wireless Internet access markets, MUs begin to worry about the commercial misuse of their personal data such as names, ages, genders, personal preferences, and residences. Since the environments of WMANs are getting more heterogeneous and complicated, it has been observed that an increasing demand on the privacy regulations that enforce *Wireless Internet Service Providers* to adopt appropriate administrative, technical, and physical security measures to protect user privacy exists. Thus, how to preserve user privacy is an emerging issue in the progress of wireless Internet development and becomes more important as users largely rely on the Internet services in their daily lives. However, the current WLAN technology cannot avoid the leakage of an MU's location information such that an adversary can easily track the MU's physical location. For example, in Fig. 1, an adversary can discover when an MU visited Wi-Fi Cafe A and how long the MU stayed. The adversary can also obtain when the MU moved to Wi-Fi Cafe B. Even if a trusted security mechanism is in place, the adversary can still track the location of the MU over time by tracing the MAC address at the link layer, which is the unique address of the network device that can be used to identify the MU. Intuitively, the MU could frequently change his MAC address to avoid being traced (although it is technically difficult to achieve this by a general user). However, the MU could

Manuscript received June 4, 2007; revised February 23, 2008 and April 7, 2008. First published May 14, 2008; current version published March 17, 2009. This work was supported by research grants from the Provincial Centre of Excellence Communications and Information Technology Ontario (CITO), Canada. The review of this paper was coordinated by Prof. Y.-B. Lin.

The authors are with the Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: rxlu@bbcr.uwaterloo.ca; xdlin@bbcr.uwaterloo.ca; h9zhu@bbcr.uwaterloo.ca; pinhan@bbcr.uwaterloo.ca; xshen@bbcr.uwaterloo.ca).

Digital Object Identifier 10.1109/TVT.2008.925304

0018-9545/$25.00 © 2008 IEEE

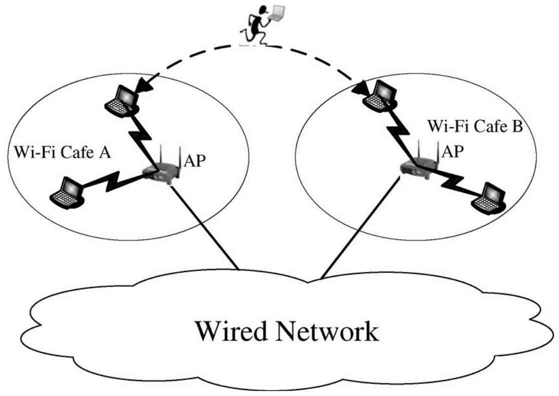


Fig. 1. IEEE 802.11-based wireless hotspot.

still be vulnerable to attacks with network traffic analysis. The reason for this is that, generally, an MU's Internet surfing habits and preferences are deterministic and almost not time-varying. In addition, dynamically changing MAC addresses affects the physical network interface reconfiguration and is not possible for the majority of MUs. Even if changing MAC addresses is possible, the MU can still be traced during the period when the MU's MAC address is static. Thus, the threat on location privacy cannot easily be mitigated by using the state-of-the-art available techniques. The task of location privacy preservation is still an open issue in spite of its imminent importance.

Since the lost of location privacy may infringe the user benefits and/or result in other negative aftereffects, a number of studies have aimed at the location privacy issues in mobile wireless networks, and some countermeasures under different application scenarios have been proposed [1]–[13]. However, most of these studies for location privacy were conducted at either on or above the network layer. When an attacker is present at the link-layer domain, attacks can be launched by tracking the location of any victim transparent to all the higher layer privacy preservation strategies.

Recently, a new approach to location privacy at the link layer has been proposed in [14], where the transmitted link-layer packets are encrypted by a shared session key between the MU and the AP. Due to the nature of broadcast in wireless networks, any party located in the same link layer can receive these encrypted packets. However, without the corresponding session key, no one except the real destination can recover the encrypted packet. Therefore, the location privacy in the presence of link-layer tracing can be protected. However, such an approach does not consider the situation that once the current shared session key is compromised, all the previous packets' privacy may be disclosed. As a result, the location privacy information on how long the MU has stayed at the current location will be deprived.

To improve the resilience on the compromise of the current link-layer location privacy, we are committed to developing a novel anonymous mutual authentication protocol with provable perfect forward link-layer location privacy. The proposed protocol is characterized by employing a suite of defense-in-depth strategies such that although an attacker has corrupted the protection of the current location information of an MU, he still cannot know how long the MU has stayed at the current location. Our main contributions are fourfold.

First, the security notions that model the perfect forward link-layer location privacy will be created. With the proposed model, the location privacy is defined such that although an MU's current location is compromised for some period of time, the attacker could be resisted from obtaining how long the MU has stayed in the current location. Thus, the MU's historical location information is taken as an important issue to be protected and is treated independently from the current physical location information. To the best of our knowledge, this is the first effort in the literature that formalizes user link-layer location privacy.

Second, with the proposed security model, we will design a novel anonymous mutual authentication protocol between the MUs and the AP by using location- and time-aware key techniques. The proposed location- and time-aware keys can help the MUs and the AP authenticate each other and negotiate a shared session key without knowing the real identity of each other. Compared with group signature techniques that were designed for similar purposes [15]–[18], the proposed anonymous mutual authentication protocol is much more efficient and is the first anonymous mutual authentication protocol dedicated for IEEE 802.11-based WLAN applications.

Third, this paper serves as the research effort that formally proves the link-layer forward-secure location privacy and quantitatively defines the user location privacy at the link layer. By applying the provable security technique, we will mathematically identify that the forward-secure location privacy is tightly related to the semantic security of the symmetric encryption scheme.

Finally, we will exercise the *Preset in Idle* technique in the proposed protocol, where some offline encryption/decryption precomputation values are precalculated and stored in a *Preset Pool* (PP), aiming to significantly accelerate the subsequent packet processing. The *Preset in Idle* technique can also effectively mitigate the vicious impacts due to packet loss and ensure the forward security of the shared key between the MUs and the AP. Detailed performance evaluation will demonstrate that the proposed protocol is much more efficient than the previously reported counterparts in [14].

The remainder of this paper is organized as follows: In Section II, we present our system formulation, where the network model, attack model, secure requirements, and formal notions of perfect forward link-layer location privacy are described. Some preliminaries and background knowledge of this paper are provided in Section III, including the bilinear group, semantic secure symmetric encryption, the secure hash function, and the forward security technique. The proposed anonymous mutual authentication protocol with provable link-layer, forward-secure location privacy is given in Section IV, followed by security analysis of the proposed protocol in Section V. In Section VI, detailed performance evaluation is provided. Finally, we draw our conclusions in Section VII.

II. Network Model and Security Notions

In this section, we model the security properties that are targeted in the proposed anonymous mutual authentication and link-layer, forward-secure location privacy protocol. We first

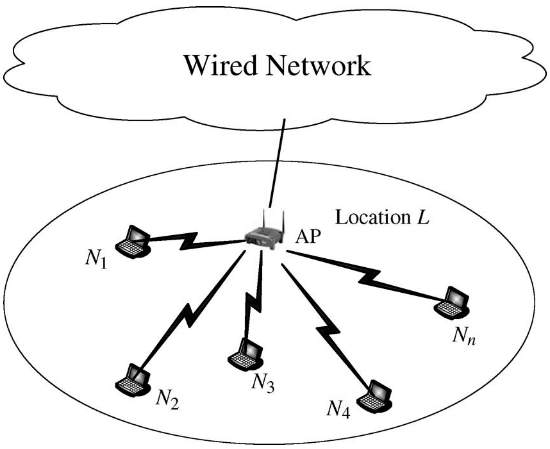


Fig. 2. Network model under consideration.

give a high-level description on the proposed model, followed by definition and formulation on the link-layer forward-secure location privacy. We then describe the types of attack of interest in this paper and present the security notions for such attacks.

A. Network Model

Without loss of generality, the last hop of a wireless access network, such as WLANs, is taken into consideration in this paper, where a single party, namely, {AP}, communicates with a group of MUs, which is denoted by $\mathcal{N} = \{N_1, N_2, \ldots, N_n\}$, as shown in Fig. 2. In addition, a malicious attacker (denoted as $\mathcal{A}$) is within the same link-layer domain that can grab the link-layer packets (or specifically the MAC layer protocol data units), whose attack capabilities will be defined later.

1) AP: An AP primarily serves as a gateway connecting the MUs within its transmission range to the Internet, where the MUs and the AP have to mutually authenticate each other in prior. The AP is assumed to be actively powered with sufficient computation capability.

2) MU: At any point in time, an MU is either idle or sending/receiving packets to/from the AP within its range and is provided with a successful mutual authentication process. We assume that the MU is relative energy constrained and could easily be compromised by a malicious attacker.

3) Network Characteristics: Communications at the link layer are performed via an open and broadcast medium such that all the MUs and the AP in the link-layer domain can listen to all the launched messages. In this case, the AP is the only network entity that can be accessed by the MU, whereas the AP can communicate with any MU in the communication range.

B. Attack Model

Before formalizing the link-layer forward-secure location privacy, we first categorize the link-layer attackers into *passive and active*, which are denoted as $\mathcal{A}^-$ and $\mathcal{A}^+$, respectively.

1) Passive Attacker: From the perspective of a passive attacker, once he hooks the link-layer domain, he is able to get access to all the on-fly packets in the domain. The general attack goal of the passive attacker could be to track the interested MU and deprive the location information of the MU by identifying the source and destination of the launched packets.

2) Active Attacker: The goal of the active attacker is the same as that of the passive attacker, except that he can also corrupt some MUs and launch a denial-of-service (DoS) attack.

1) *Corrupt nodes.* $\mathcal{A}^+$ can dynamically corrupt an MU, followed by tracking the packets transmitted between the AP and the MU. Note that the AP in reality is not easy to corrupt.
2) *Launch a DoS attack.* To protect the location privacy, there is no source or destination address marked on the transmitted link-layer packets. Therefore, an active attacker may launch a DoS attack without being tracked. For example, flooding the network with bogus or stored broadcast packets would be a very effective DoS attack against the entire network.

Obviously, an active attacker is stronger than a passive attacker. Thus, in our study, we only consider active attackers. Since our goal is to model the link-layer forward-secure location privacy, the proposed protocol will focus on the impact by an active attacker in corrupting some MUs.

C. Security Requirements

To achieve perfect forward link-layer location privacy, there is no any explicit identity adopted for a specific MU and AP. Instead, a more secure technique is taken for achieving anonymous mutual authentication, which is expected to better protect the MU's location privacy. In the following, we list some desired security requirements in the development of our protocol.

1) Achieve mutual authentication between the MU and the AP without disclosing the real identity of the MU to the other parties in the link-layer domain.
2) Obtain a secure session key shared between the MU and the AP before packet transmission.
3) Hide the link-layer traffic to protect the MU's traffic from direction inference.
4) An attacker can neither distinguish a transmitted packet nor trace any possible transmitted packet.
5) An attacker cannot learn the previous packets to analyze the MU's historical location information, such as how long the MU had stayed at this location, even if the current packet and the current key of the MU are compromised.

D. Security Notions on Forward-Secure Location Privacy

The notions of *forward-secure location privacy* are defined corresponding to the given security requirements. One of the intrinsic properties of link-layer forward-secure location privacy indicates that all the preceding data packets should remain in privacy protection, even if the current packet's location privacy has been cracked and/or the current key for encryption has been compromised. With this, the MU's previous private location information can be well protected such that the attacker can never get a chance to trace the MU.

Definition 1 (Link-Layer, Forward-Secure Location Privacy): The security of link-layer forward-secure location privacy is defined on a *game* played among an adversary (denoted as $\mathcal{A}$), a group of MUs (denoted as $\mathcal{N} = \{N_1, N_2, \ldots, N_n\}$), and the access point (denoted as AP). In the game, the adversary $\mathcal{A}$ interacts with a hypothetical probabilistic algorithm, called a *challenger*, which may respond to queries made by the adversary $\mathcal{A}$. In the *game* initialization phase, each N_i in $\mathcal{N}$ for $i = 1, \ldots, n$, is assumed to have made the anonymous mutual authentication protocol with AP. A session key (denoted as K_i), which could repeatedly be updated, is shared between N_i and AP for $i = 1, \ldots, n$. For example, the session key in time period $T-1$ (denoted as $K_{i(T-1)}$) is taken as K_i, and the session key updated in time period T (denoted as K_{iT}) is computed based on the session key of $T-1$ through a one-way hash function.

In the *game* running phase, an active adversary (denoted as $\mathcal{A}$) may make the following queries to its challenger:

$Execute()$: This query models a passive attack, where $\mathcal{A}$ can get access to all the launched packets in the link-layer domain due to the nature of broadcast medium.

$H\text{-}corrupt(N_i)$: $\mathcal{A}$ thoroughly corrupts N_i and obtains its initial shared key K_i. This query models the most serious corruption.

$M\text{-}corrupt(N_i)$: $\mathcal{A}$ corrupts N_i at time T and obtains the shared session key K_{iT}.

$L\text{-}corrupt(C)$: $\mathcal{A}$ corrupts an encrypted transmitted packet C. This case could easily happen in the real world when the accidental storage error takes place at the MU.

$Test(m)$: After $\mathcal{A}$ $H\text{-}corrupts$ a number of i_h MUs denoted as $\{N_1', N_2', \ldots, N_{i_h}'\}$, it chooses a packet m and makes a $Test(m)$ query to the other MUs $\mathcal{N}' = \mathcal{N}/\{N_1', N_2', \ldots, N_{i_h}'\}$ to its challenger. The size of $\mathcal{N}'$ is now $n - i_h$, in which the MUs that made the $M\text{-}corrupt$ queries and the uncorrupted MUs are included. Since the goal is to achieve the perfect forward location privacy, we consider an extreme case in our model by assuming that $\mathcal{A}$ can $M\text{-}corrupt$ all the MUs in the link-layer domain and know the identities of all MUs in $\mathcal{N}'$. This can make our model sufficiently cope with all the situations in achieving the link-layer forward-secure location privacy.

When receiving $Test(m)$, $\mathcal{A}$'s challenger randomly chooses an MU N_w in $\mathcal{N}'$, where $1 \le w \le n - i_h$. If N_w has been queried by $M\text{-}corrupt(N_w)$ in some time period T, a session key in the period j denoted as K_{wj} is chosen, where $j < T$. Otherwise, a session key K_{wj} in any period j is chosen. The challenger uses the key K_{wj} to encrypt the packet m into C and returns the encrypted packet C to the adversary $\mathcal{A}$. Finally, the adversary $\mathcal{A}$ returns his guess to the MU.

The success of $\mathcal{A}$ in the game is quantified in terms of $\mathcal{A}$'s advantage in correctly guessing the MU, i.e., its ability to guess the integer w. We define $\mathcal{A}$'s guessing advantage as

$$\mathbf{Adv}_{\mathcal{A}} = (n - i_h) \times \Pr\left[\mathcal{A}(m, C) = w\right] - 1.$$

We claim that the protocol can achieve the *forward-secure location privacy* if $\mathbf{Adv}_{\mathcal{A}}$ is negligible.

III. Preliminaries

In this section, we introduce the building blocks of the proposed protocol, which include bilinear groups [19], [20], semantic secure symmetric encryption [21], secure hash function [22], and forward security technique [23], [24] to achieve provable link-layer, forward-secure location privacy.

A. Bilinear Groups

Let $\mathbb{G}$ and $\mathbb{G}'$ be two cyclic additive groups and $\mathbb{G}_T$ be a cyclic multiplicative group of the same prime order q, i.e., $|\mathbb{G}| = |\mathbb{G}'| = |\mathbb{G}_T| = q$. Let P be a generator of $\mathbb{G}$, P' be a generator of $\mathbb{G}'$, and ψ be an isomorphism from $\mathbb{G}'$ to $\mathbb{G}$, with $\psi(P') = P$. An efficient admissible bilinear map $e : \mathbb{G} \times \mathbb{G}' \to \mathbb{G}_T$ with the following properties: 1) Bilinear: For all $P_1 \in \mathbb{G}$, $Q_1 \in \mathbb{G}'$, and $a, b \in \mathbb{Z}_q^*$, $e(aP_1, bQ_1) = e(P_1, Q_1)^{ab}$. 2) Nondegenerate: There exist $P_1 \in \mathbb{G}$ and $Q_1 \in \mathbb{G}'$ such that $e(P_1, Q_1) \neq 1_{\mathbb{G}_T}$. 3) Computable: An efficient algorithm exists to compute $e(P_1, Q_1)$ for any $P_1 \in \mathbb{G}$ and $Q_1 \in \mathbb{G}'$. Such an admissible bilinear map $e : \mathbb{G} \times \mathbb{G}' \to \mathbb{G}_T$ can be constructed by the modified Weil or Tate pairings on elliptic curves. As mentioned in [25], the Tate pairing on Miyaji–Nakabayashi–Takano (MNT) curves [26] gives an efficient implementation. We define a bilinear parameter generator $\mathcal{G}en$ that takes a security parameter k as input and outputs a 7-tuple $(q, \mathbb{G}, \mathbb{G}', \mathbb{G}_T, e, P, P')$ as the bilinear parameters, including a prime number q with $|q| = k$, three cyclic groups $\mathbb{G}$, $\mathbb{G}'$, and $\mathbb{G}_T$ of the same order q, an admissible bilinear map $e : \mathbb{G} \times \mathbb{G}' \to \mathbb{G}_T$, and generators P and P' of $\mathbb{G}$ and $\mathbb{G}'$, respectively.

B. Semantic Secure Symmetric Encryption

1) Definition 2 (Symmetric Encryption Scheme): Given a security parameter k, a symmetric encryption scheme π is defined by two algorithms $(\mathbf{E}, \mathbf{D})$, parameterized by a key k uniformly distributed in $\{0,1\}^k$. Let l be the bit length of the message encrypted. Then, the (randomized) encryption algorithm $\mathbf{E}_k$, on input of the message $m \in \{0,1\}^l$, and a nonce r, outputs a ciphertext c. The (deterministic) decryption algorithm $\mathbf{D}_k$, on input of a ciphertext c, outputs the corresponding message m or $\perp$ if c is invalid. In addition, the symmetric encryption scheme should hold the consistence constraint: Given $k \in \{0,1\}^k$, for all $m \in \{0,1\}^l$, and any nonce r, we have $m = \mathbf{D}_k(\mathbf{E}_k(m, r))$.

The natural security notion for the symmetric encryption is *semantic security*, where the ciphertext does not help learn any information about the plaintext. Before the formal definition of *semantic security* is introduced, a *game* taken as an analogy is described as follows: The adversary in the *game* plays a role as a ciphertext distinguisher, which is denoted as $\mathcal{D}$, which first chooses a message $m \in \{0,1\}^l$ and sends it to $\mathcal{D}$'s challenger. The challenger then adaptively responds with several ciphertexts $\{c_1, c_2, \ldots, c_t\}$ based on $\mathcal{D}$'s request, where each $c_i = \mathbf{E}_k(m, r_i)$ for fixed key k and different nonce r_i. At the end, a coin b with two faces "0" and "1" is flipped at the challenger. If it lands $b = 1$, then a real ciphertext c of m is sent back to $\mathcal{D}$. Otherwise, a random number c generated according to the distribution the same as that of the ciphertexts is sent

$$\begin{array}{c} \text{Period}_0 \ \| \ \text{Period}_1 \ \| \ \text{Period}_2 \ \| \quad \| \ \text{Period}_T \\ K_{i0} \rightarrow \boxed{h} \rightarrow K_{i1} \rightarrow \boxed{h} \rightarrow K_{i2} \rightarrow \boxed{h} \cdots \boxed{h} \rightarrow K_{iT} \\ \| \qquad \| \qquad \| \qquad \| \\ \text{Evolution: } K_{i0} = K_i,\ K_{ij} = h\left(K_{i(j-1)}\right),\ \text{where } j \geq 1 \end{array}$$

Fig. 3. Key evolution mechanism.

back. Finally, the distinguisher $\mathcal{D}$ outputs a guess $b' \in \{0,1\}$ and wins the game if $b = b'$. We define the distinguisher $\mathcal{D}$'s advantage in attacking the symmetric encryption scheme π as

$$\mathbf{Adv}_{\pi,\mathcal{D}}(k,l) = 2 \times \Pr[b' = b] - 1.$$

The probability is over the random bits used by the *game*.

Definition 3 (Semantic Security): The symmetric encryption scheme $\pi = (k, l, \mathbf{E}, \mathbf{D})$ is *semantically secure* if for all distinguisher $\mathcal{D}$, the function $\mathbf{Adv}_{\pi,\mathcal{D}}(k,l)$ is negligible.[1]

C. Secure Hash Function

A one-way hash function $h()$ is said to be secure if the following properties are satisfied: 1) $h()$ can take a message of arbitrary length as input and produce a message digest of a fixed-length output. 2) Given x, it is easy to compute $h(x) = y$. However, it is hard to compute $h^{-1}(y) = x$ given y. 3) Given x, it is computationally infeasible to find $x' \neq x$ such that $h(x') = h(x)$ [22].

D. Forward Security

To achieve the forward-secure location privacy, the proposed protocol takes advantages of a forward security key technique [23], [24], which ensures that any past key could not be learned even if an attacker had obtained the current key [28]. In this case, the forward-secure location privacy is achieved.

A forward security key is usually created by way of a key evolution mechanism [29]. Fig. 3 illustrates its main idea. In a forward-secure key scheme, the time during which a shared key K_i between AP and an MU denoted as N_i is supposed to be functionally divided into T periods, where AP and N_i share an evolving key denoted as $K_{ij}, j = 0, 1, \ldots, T$, corresponding to each time period. The key of the current period is computed from the key of the previous period by way of a one-way function $h()$, e.g., $K_{ij} = h(K_{i(j-1)})$. Thus, the leakage of current key K_{ij} does not lead to exposure of previous keys.

IV. Proposed Protocol

The proposed protocol is designed to achieve anonymous mutual authentication between the AP and each MU with link-layer forward-secure location privacy. The network architecture under consideration is shown in Fig. 4. In the key predistribution phase, the trusted authority (denoted as TA) first delegates a location-aware key (denoted as LK) to AP and distributes a time-aware key (denoted as TK) to the MU (denoted as N_i). Then, when N_i enters into a location (denoted as L), N_i and AP perform the anonymous mutual authentication and negotiate a secure session key (denoted as K_i), by which the subsequently transmitted MAC packets can be kept in privacy. To achieve the forward-secure location privacy, a key evolution mechanism is employed on K_i. To speed up the MAC packet process, a *Preset in Idle* technique is devised, which is an offline encryption/decryption precomputation technique aiming to significantly improve the efficiency.

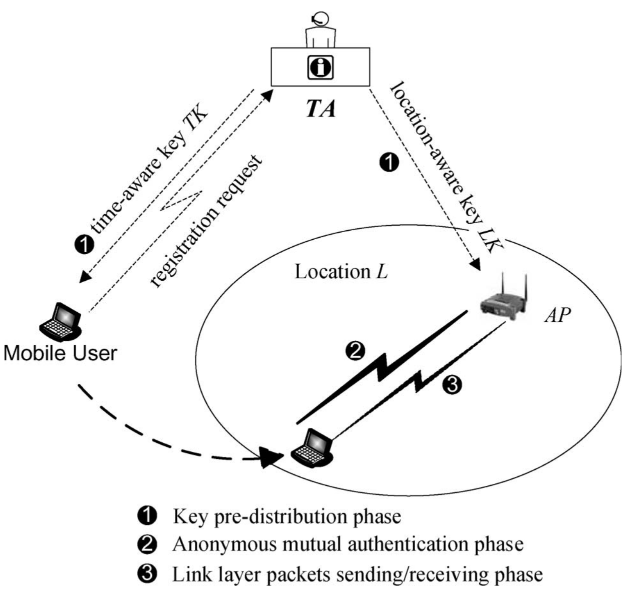

TA
❶ time-aware key TK
registration request
❶ location-aware key LK
Location L
Mobile User
AP
❷
❸
❶ Key pre-distribution phase
❷ Anonymous mutual authentication phase
❸ Link layer packets sending/receiving phase

Fig. 4. Proposed anonymous mutual authentication protocol.

TABLE I
NOTATION AND DESCRIPTION

Notations	Descriptions
k, l:	two security parameters, $k \approx 160$, and $l \gg k$.
h, f:	two secure cryptographic one-way hash functions, where $h, f : \{0,1\}^k \rightarrow \{0,1\}^k$.
$\lvert x \rvert$:	length of a message x $(= 1 + \lfloor \log_2 x \rfloor$ for $x \geq 1)$.
$x \| y$:	concatenation two messages x and y.
$x \oplus y$:	bit-wise xor two messages x and y, where $\lvert x \rvert = \lvert y \rvert$.
$\mathbf{E}_{\mathrm{k}}(m, r)$:	symmetric encryption, encrypt a message m with key k and a nonce r, where $\lvert m \rvert = l$.
MP:	MAC packet, where $\lvert \mathsf{MP} \rvert = l$.
RID:	random identifier, which is the most important key factor in our protocol design, where $\lvert \mathsf{RID} \rvert = k$.
PRNG(r) :	a pseudo-random number generator, with r as a seed, where $\lvert r \rvert = k$, outputs a l-bit pseudo-random number [30].
EF:	encrypted factor, where $\lvert \mathsf{EF} \rvert = l$.

TABLE II
MESSAGE FORMAT

Packet Type	Random Identifer	Encrypted MAC Packet
1 bit	160 bits	l bits

A. Notations and Message Format

The notations for introducing the proposed protocol are listed in Table I, whereas the message format is shown in Table II. In the proposed message format, the first field is a 1-bit binary packet type identifier, which is "1" if the anonymous packet is an authentication request packet, and "0" otherwise. The

[1] Note that a heuristic argument can easily show that such a variant definition of *semantic security* is equivalent to the traditional one [21] and is similar to the session key's *semantic security* in a key-exchange protocol [27].

second field is the random identifier. The last field contains the encrypted MAC packet.

B. Key Predistribution Phase

With TA, the keys to all the APs and MUs are created and distributed via the network. Prior to the key predistribution, TA first initializes the system parameters as follows:

1) Generate the 7-tuple $(q, \mathbb{G}, \mathbb{G}', \mathbb{G}_T, e, P, P')$ by running the bilinear parameter generator $\mathcal{G}en(k)$.
2) Choose a random number $s \in \mathbb{Z}_q^*$ as the *master key* and compute the corresponding public parameters $P'_{\text{pub}} = sP' \in \mathbb{G}'$ and $P_{\text{pub}} = \psi(P'_{\text{pub}}) = sP \in \mathbb{G}$.
3) Choose the pseudorandom number generator $\mathbf{PRNG}()$ and hash functions h, f, $H : \{0,1\}^* \to \mathbb{G}'$, and $F : \mathbb{G}_T \to \{0,1\}^l$ as the public parameters.

Let AP be assigned to work at location L by TA. With the location information, TA calculates the location-aware key $LK = sH(L) \in \mathbb{G}'$ and then distributes all public system parameters $(q, \mathbb{G}, \mathbb{G}', \mathbb{G}_T, e, P, P', P'_{\text{pub}}, P_{\text{pub}}, h, f, H, F)$ and location-aware key LK to AP.

When an MU N_i registers himself to the system, TA and N_i first negotiate a proper valid period T, then TA computes the time-aware key $TK = sH(T) \in \mathbb{G}'$ for N_i and distributes the same public parameters, valid period T, and time-aware key TK to N_i with a secure channel. N_i can use TK to authenticate himself in T without disclosing his real identity.

Note that due to the hardness of the discrete logarithm problem in $\mathbb{G}'$, it is computationally infeasible to deduce the master key s from either LK or TK. Thus, although AP and N_i are compromised, an adversary still cannot get the master key s.

C. Anonymous Mutual Authentication Phase

Our protocol is characterized by its simplicity and request-response features, where only two packets between the AP and each MU are exchanged. Suppose that an MU denoted as N_i owns the time-aware key TK within time period T. When N_i enters into L, the mutual authentication procedure is initiated, where a session key denoted as K_i is created. Note that in the mutual authentication process, both N_i and AP do not know the real identity of the other. Fig. 5 summarizes the proposed anonymous mutual authentication protocol, which is further explained in the list that follows.

1) N_i first chooses two nonces $r_1, j \in \mathbb{Z}_q^*$ and computes $C_1 = r_1P$. Then, N_i uses the location information L to compute $R_1 = F(e(P_{\text{pub}}, H(L))^{r_1})$, and $C_2 = R_1 \oplus M$, where $M = j\Vert T\Vert\text{Timestamp}$. Note that if M is less than l bits, the padding will be appended. At the end, according to the message format in Table II, N_i broadcasts $C = \boxed{1}\boxed{C_1}\boxed{C_2}$ in the location L.
2) When AP receives $C = \boxed{1}\boxed{C_1}\boxed{C_2}$, it first uses its location-aware key LK to compute $R'_1 = F(e(C_1, LK))$ and then recovers $M = j\Vert T\Vert\text{Timestamp}$ by computing $C_2 \oplus R'_1$. If either T or the timestamp is overdue, AP will not process this packet; otherwise, AP computes $R_3 = F(e(\psi(H(T)), LK)^j)$, $C_3 = f(j)$, then chooses a nonce $r_2 \in \mathbb{Z}_q^*$, and computes $C_4 = R_3 \oplus M'$, where $M' = r_2P\Vert L\Vert\text{Timestamp}'$. According to the message format in Table II, AP broadcasts $C = \boxed{0}\boxed{C_3}\boxed{C_4}$ in the location L. In the end, AP computes the shared session key $K_i = r_2C_1 = r_1r_2P$.
3) When a packet $C = \boxed{0}\boxed{C_3}\boxed{C_4}$ comes within its expected time, N_i first checks whether $C_3 \overset{?}{=} f(j)$. If it does hold, N_i takes this packet as the authentication response packet, recovers $M' = r_2P\Vert L\Vert\text{Timestamp}'$ by computing $C_4 \oplus R'_3$, and computes the shared session key $K_i = r_1r_2P$ in the end.

N_i authorized within T		AP at location L
1) $r_1, j \in \mathbb{Z}_q^*$, $C_1 = r_1P$		
$R_1 = F(e(P_{pub}, H(L))^{r_1})$		
$M = j\Vert T\Vert\text{Timestamp}$		
$C_2 = R_1 \oplus M$		
$C = \boxed{1}\boxed{C_1}\boxed{C_2}$	$\xrightarrow{C}$	
2)		$R'_1 = F(e(C_1, LK))$
		$M = C_2 \oplus R'_1$
		$M \to j\Vert T\Vert\text{Timestamp}$
		$R_3 = F(e(\psi(H(T)), LK)^j)$
		$C_3 = f(j)$, $r_2 \in \mathbb{Z}_q^*$
		$M' = r_2P\Vert L\Vert\text{Timestamp}'$
		$C_4 = R_3 \oplus M'$
	$\xleftarrow{C}$	$C = \boxed{0}\boxed{C_3}\boxed{C_4}$
3) check $C_3 \overset{?}{=} f(j)$		$K_i = r_2C_1 = r_1r_2P$
$R'_3 = F(e(\psi(TK), H(L))^j)$		
$M' = C_4 \oplus R'_3$		
$M' \to r_2P\Vert L\Vert\text{Timestamp}'$		
$K_i = r_1r_2P$		

Fig. 5. Anonymous mutual authentication protocol.

Correction: Obviously, due to the bilinear pairing property, the correction of the protocol will hold based on the following two relations:

$$\begin{aligned} e\left(P_{\text{pub}}, H(L)\right)^{r_1} &= e\left(sP, r_1H(L)\right) \\ &= e\left(r_1P, sH(L)\right) = e(C_1, LK) \quad (1) \\ e\left(\psi(TK), H(L)\right)^j &= e\left(s\psi\left(H(T)\right), H(L)\right)^j \\ &= e\left(\psi\left(H(T)\right), sH(L)\right)^j \\ &= e\left(\psi\left(H(T)\right), LK\right)^j . \quad (2) \end{aligned}$$

Security Analysis: The proposed anonymous mutual authentication protocol is based on the bilinear groups by considering the advantages of its high security assurance with smaller sized keys and less bandwidth consumption. The mutual authentication is accomplished in a request-response manner between the AP and each MU. Detailed analysis on security is given in the list that follows.

1) *N_i can explicitly authenticate AP but need not care the real identity of AP.* In the request packet $C = \boxed{1}\boxed{C_1}\boxed{C_2}$, (C_1, C_2) is actually the *identity-based encryption's* ciphertext [19] with respect to the location information L. Without knowing the corresponding location-aware key $LK = sH(L)$, it is infeasible for an adversary to recover the correct nonce j embedded

in C_2. Therefore, when the MU receives the response packet $C = \boxed{0}\boxed{C_3}\boxed{C_4}$, it can check whether $C_3 = f(j)$ within a reasonable time. If it holds, the MU can explicitly authenticate AP that is really located at L. In the authentication protocol, AP's real identity will not be disclosed during the authentication phase. Therefore, the anonymity of AP can be achieved. Note that although AP is anonymous, it can only perform its right in location L because its location-aware key $LK = sH(L)$ is valid only in L.

2) *AP can implicitly authenticate N_i without knowing its real identity*. Note that to implement *implicit authentication*, it always requires establishing a confidential session key between the two parties. In the response packet $C = \boxed{0}\boxed{C_3}\boxed{C_4}$ of our protocol, the static shared key $e(\psi(H(T)), LK) = e(\psi(H(L)), TK)$ between N_i and AP has been embedded in C_4. If N_i is indeed authorized at time period T, it can compute the shared session key $K_i = r_1 r_2 P$. However, if N_i is not authorized at time period T, it cannot recover $r_2 P$ from C_4 without the time-aware key $TK = sH(T)$. Subsequently, the MU also cannot compute the shared session key $K_i = r_1 r_2 P$. Thus, implicit authentication is achieved in our protocol. Obviously, AP can authenticate N_i, which is actually authorized at time period T and has no idea on the real identity of N_i.

With the aforementioned analysis, we claim that our protocol can achieve anonymous mutual authentication between N_i and AP. After a successful mutual authentication process, a session key $K_i = r_1 r_2 P$ is generated based on the nonces r_1 and r_2. The security of K_i is analyzed in the list that follows.

1) *Session key security:* The first security requirement for a session key is that the session key K_i is only known by N_i and AP. Clearly, since r_1 and r_2 are only known by N_i and AP, respectively, even if an adversary obtains $r_1 P$ and $r_2 P$, the adversary still cannot compute $K_i = r_1 r_2 P$ because the computational Diffie–Hellman problem is hard in $\mathbb{G}$.
2) *Perfect forward secrecy:* Perfect forward secrecy here means that the compromise of either the location-aware key LK or the time-aware key TK does not affect the security of the previous session keys. Since the compromise of a session key also requires the knowledge of two nonces r_1 and r_2 used in the session, all previous session keys are thus secure.

D. Link-Layer Packets Sending/Receiving Phase

After N_i and AP at location L made the implicit mutual authentication and negotiated a secure shared session key K_i, they can start to transmit data packets with link-layer security. To speed up the packet process and achieve the perfect forward location privacy, we propose a *Preset in Idle* technique that is embedded in the proposed protocol.

1) Preset in Idle Technique: The main idea of this technique is that each MU and AP maintain a PP, which stores all possible offline encryption/decryption precomputation values, to accelerate the subsequent packet process. Once N_i and AP have negotiated a session key K_i and a shared nonce j, both of them set a synchronization sequence number $(\mathbf{SN})$ s_i, where the following initial values are computed:

TABLE III
PP MESSAGE FORM

Pseudo Node Name (PNN)	SN	RID	Key	EF
PN_i	$s_i = j$	r_{ij}	K_{ij}	c_{ij}

$$\begin{aligned} s_i &= j \\ K_{ij} &= h(K_i) \\ r_{ij} &= f(K_{ij}) \\ c_{ij} &= \mathbf{E}_{K_{ij}}\left(\mathbf{PRNG}(r_{ij}), s_i\right). \end{aligned} \tag{3}$$

The entry $(PN_i, s_i, r_{ij}, K_{ij}, c_{ij})$ is stored in the PP, which has a form shown in Table III.

In the PP, the first field denotes the identifier of the MUs in the same link layer. Because AP does not know the real identity of the MUs, this field is thus filled with pseudoidentifiers chosen by AP (denoted as PN_i), corresponding to each MU in the mutual authentication phase, by which AP can communicate with the MUs. However, this field does not exist at the MU side because each MU exclusively communicates with AP. The size of PP (denoted as s_p) is a critical parameter, which is constrained by the miniature of mobile devices. Nonetheless, the APs with a larger value of s_p can yield higher efficiency and better tackle the packet loss issue. The impact of having different values of s_p is also related to the traffic patterns, where n_p packets in a row between a source and a destination with $n_p \leq s_p$ can lead to high robustness and efficiency. In this case, the loss of previous n_l packets (where $n_l \leq s_p$) can be seamlessly tackled. Therefore, a compromise between the overhead and benefits gained by allocating a large PP size should be carefully initiated.

Assume that the maximum number of continuous packets with the same source and destination is N_P. Because of the possible poor wireless channel conditions, the packet loss probability is assumed to be ρ, where $0 < \rho < 1$. Let X be the number of lost packets among the total N_P packets, which follow a binomial distribution $\mathfrak{B}(N_P, \rho)$, i.e.,

$$P\{X = \kappa\} = \binom{N_P}{\kappa} \rho^{\kappa} (1-\rho)^{N_P - \kappa}, \qquad \kappa = 0, 1, 2, \ldots, N_P. \tag{4}$$

Then, we have

$$\mathbf{E}(X) = N_P \cdot \rho \qquad \mathbf{Var}(X) = N_P \cdot \rho \cdot (1-\rho) \tag{5}$$

which means that the average number of possible lost packets is $\lambda = N_P \cdot \rho$. Therefore, the size of PP should be set to $s_p \geq \lambda + 1$ to efficiently tackle the packet loss issue.

When N_i or AP is idle, it can run Algorithm 1 (A1) to store the preset values in the PP. Table IV exemplifies a PP fully filled with preset values, where j is the current sequence number (denoted as s_i). The PP is dynamically updated and filled up with more preset values by invoking A1 when the number of N_i's entry in the PP in the idle state is less than s_p. When

TABLE IV
PRESET VALUES STORED IN THE PP

PNN	SN	RID	Key	EF
PN_i	$s_i = j$	r_{ij}	K_{ij}	c_{ij}
PN_i	$j+1$	$r_{i(j+1)}$	$K_{i(j+1)}$	$c_{i(j+1)}$
PN_i	$j+2$	$r_{i(j+2)}$	$K_{i(j+2)}$	$c_{i(j+2)}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
PN_i	$j+s_p-1$	$r_{i(j+s_p-1)}$	$K_{i(j+s_p-1)}$	$c_{i(j+s_p-1)}$

the number of N_i's entry in the PP in the state of sending or receiving packets is less than $\lambda+1$, A1 will be invoked to fill to at least $\lambda+1$ entries to tackle the packet loss issue.

```
Algorithm 1: [A1] PresetInIdle()
  Data: $(j, K_{ij})$, where $K_{ij}$ is the shared key between $N_i$
        and $AP$ under sequence number $s_i = j$.
  Result: $(PN_i, j+1, r_{i(j+1)}, K_{i(j+1)}, c_{i(j+1)})$.
  begin
    $s_i = j+1$
    $K_{i(j+1)} = h(K_{ij})$
    $r_{i(j+1)} = f(K_{i(j+1)})$
    $c_{i(j+1)} = \mathbf{E}_{K_{i(j+1)}}(\mathbf{PRNG}(r_{i(j+1)}), s_i)$
    return $(PN_i, s_i, r_{i(j+1)}, K_{i(j+1)}, c_{i(j+1)})$
  end
```

2) Sending a Unicast Packet: Algorithm 2 (A2) describes how to efficiently send a unicast encrypted packet. Because of the PP, the algorithm can efficiently send n_p packets in a row for $n_p \leq s_p - \lambda$.

```
Algorithm 2: [A2] Sending a Unicast Packet
  Data: Intercept the MAC packet $m$ from MAC to PHY.
  Result: Send the encrypted packet $C$ using the PHY
          mechanisms.
  begin
    if is the AP then                              /* is Access Point */
      Determine the destination node $N_i$ by the pseudo-
      name $PN_i$
      Fetch the current entry $(PN_i, j, r_{ij}, K_{ij}, c_{ij})$ and
      remove it from the PP
      if the number of entries in the PP is less
      than $\lambda+1$ then
        Invoke PresetInIdle() to fill to $\lambda+1$
      end
      Update the current entry as
      $(PN_i, j+1, r_{i(j+1)}, K_{i(j+1)}, c_{i(j+1)})$
    else                                           /* is the MU */
      Directly fetch the current entry
      $(PN_i, j, r_{ij}, K_{ij}, c_{ij})$ and remove it from the
      database
      if the number of entries in the PP is less
      than $\lambda+1$ then
        Invoke PresetInIdle() to fill to $\lambda+1$
      end
      Update the current entry as
      $(PN_i, j+1, r_{i(j+1)}, K_{i(j+1)}, c_{i(j+1)})$
    end
    $c = m \oplus c_{ij}$                          /* encryption process */
    return the encrypted packet $C$ as
                              | 0 | $r_{ij}$ | $c$ |
  end
```

3) Receiving a Unicast Packet: Algorithm 3 (A3) describes how to efficiently receive a unicast packet. Due to the PP technique, A3 can be adopted to efficiently receive n_p continuous packets, for $n_p \leq s_p - \lambda$. In addition, only if the number of lost packets n_l is less than s_p can the packet loss event be seamlessly tackled.

```
Algorithm 3: [A3] Receiving a Unicast Packet
  Data: Intercept the encrypted packet $C$ from PHY to
        MAC.
  Result: Send the recovered packet $m$ to the MAC level,
          or do nothing if $\perp$.
  begin
    Parse the encrypted packet $C$ as
                              | 0 | $r_{ij}$ | $c$ |
    Look up the entry $(PN_i, j, r_{ij}, K_{ij}, c_{ij})$ in the
    PP with the search condition $r_{ij} = \mathbf{RID}$
    if $(PN_i, j, r_{ij}, K_{ij}, c_{ij})$ is found then
      if it is the current entry then              /* normal */
        Fetch the entry $(PN_i, j, r_{ij}, K_{ij}, c_{ij})$, remove
        it from the PP
        if the number of entries in the PP is less
        than $\lambda+1$ then
          Invoke PresetInIdle() to fill to $\lambda+1$
        end
        Update the new current entry
        $(PN_i, j+1, r_{i(j+1)}, K_{i(j+1)}, c_{i(j+1)})$
      else                                         /* packet loss */
        Fetch the entry $(PN_i, j, r_{ij}, K_{ij}, c_{ij})$, remove
        $PN_i$'s all previous entries and current entry
        from the PP
        if the number of entries in the PP is less than $\lambda+1$
        then
          Invoke PresetInIdle() to fill to $\lambda+1$
        end
        Update the new current entry
        $(PN_i, j+1, r_{i(j+1)}, K_{i(j+1)}, c_{i(j+1)})$
      end
    else                                           /* no entry found */
      The encrypted packet $C$ is not for me
      return $\perp$
    end
    set $m = c \oplus c_{ij}$                      /* decryption */
    if m isn't detected error then
      return $m$
    else                                           /* packet erroe */
      return $\perp$
    end
  end
```

4) Remark on Random Identifier ($\mathbf{RID}$)*:* In the proposed protocol, the random identifier $\mathbf{RID}$ is a unique device among all the counterpart studies for identifying the encrypted packets. The value of $\mathbf{RID}$ is computed by a secure hash function

f, whose domain is $\{0,1\}^k$. Therefore, there are 2^k possible values for $\mathbf{RID}$. Here, we first evaluate how many packets are possibly transmitted when collision occurs on $\mathbf{RID}$. Let $D(\varrho)$ denote the probability that at least one collision occurs on $\mathbf{RID}$ after ϱ packets were launched, and let D_i be the event that the ith packet collides with one of the previous packets on $\mathbf{RID}$. Then, $\Pr[D_i]$ is upper bounded by $(i-1)/2^k$, and

$$\begin{aligned} D(\varrho) &= \Pr[D_1 \vee D_2 \vee \cdots \vee D_\varrho] \\ &\leq \Pr[D_1] \vee \Pr[D_2] \vee \cdots \vee \Pr[D_\varrho] \\ &\leq \frac{0}{2^k} + \frac{1}{2^k} + \cdots + \frac{\varrho-1}{2^k} = \frac{\varrho(\varrho-1)}{2^{k+1}} \end{aligned} \tag{6}$$

which means that the upper bound of the collision probability $D(\varrho)$ grows with $O(\varrho^2 2^{-k})$. When $D(\varrho) \to 1/2$, $\varrho^2 \approx 2^k$. Therefore, with the security parameter $k \approx 160$, after almost $\varrho \approx 2^{k/2} = 2^{80}$ total packets transmitted in the same link-layer domain, a collision on $\mathbf{RID}$ may occur with a probability approaching $1/2$. Therefore, the mechanism of random identifier $\mathbf{RID}$ in our protocol is feasible.

V. Security Analysis

In this section, the security of the proposed protocol is analyzed. We first formally prove the perfect forward link-layer location privacy in the proposed protocol and then discuss its resilience against the possible DoS attacks.

A. Perfect Forward Location Privacy

Proposition 1: The proposed protocol can achieve the link-layer forward-secure location privacy.

Suppose that there are n MUs (denoted as $\mathcal{N} = \{N_1, N_2, \ldots, N_n\}$) and one corresponding AP (denoted as AP) in a common link-layer domain, where each MU secretly communicates with AP using its prenegotiated keys. Assume that there exists a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ that enters into the same link-layer cloud and can crack the forward-secure location privacy within time τ with nonnegligible advantage probability $\mathbf{Adv}_{\mathcal{A}} = \epsilon$ after i_h $H\text{-}corrupt$ queries and other $Execute$, $M\text{-}corrupt$, and $L\text{-}corrupt$ queries (as defined in Section II-D). Then, we can use $\mathcal{A}$ to construct another PPT distinguisher $\mathcal{D}$, which can break the semantic security of the symmetric encryption π with another nonnegligible probability.

The proposed link-layer forward-secure location privacy aims to achieve the strongest location privacy among all the reported schemes at the link layer, since the compromise of the current packet's privacy does not affect the previous packets' privacy. Although an adversary corrupts an MU, the adversary still cannot learn how long the MU has stayed at the current location, which is considered as important location privacy information. Therefore, our goal is to guarantee the previous packets' privacy, for which our strategy is to prove the location privacy in an extreme case that the adversary $\mathcal{A}$ knows all MUs' identities, as shown in Section II-D. If the perfect forward location privacy can be proved in such an extreme case, the perfect forward location privacy will hold in any other case.

In the following, we will describe in detail how to construct such a distinguisher $\mathcal{D}$ from the adversary $\mathcal{A}$'s capability.

1) $\mathcal{A}$ randomly chooses a packet $m \in \{0,1\}^l$ and makes $Test(m)$ query to $\mathcal{D}$.
2) $\mathcal{D}$ first chooses a random number $r \in \{0,1\}^k$, randomly selects one MU denoted as N_w from $\mathcal{N}' = \mathcal{N}/\{N'_1, N'_2, \ldots, N'_{i_h}\}$, where $1 \leq w \leq n - i_h$, and then sends (r, N_w) to its challenger.
3) Assume that N_w was asked $M\text{-}query$ at synchronization sequence number j once. Then, the challenger should choose one previous key k, whose synchronization sequence number is less than j. According to the $\mathcal{D}$'s request, the challenger first returns several ciphertexts $c_1, c_2, \ldots, c_t$ corresponding to $\mathbf{PRNG}(r) \in \{0,1\}^l$ to the distinguisher $\mathcal{D}$, where $c_i = \mathbf{E}_k(\mathbf{PRNG}(r), r_i)$, for nonce r_i. At some point, the challenger flips a coin b. If it lands $b = 1$, the challenger computes $c^\star = \mathbf{E}_k(\mathbf{PRNG}(r), r')$ for some nonce r'. If the coin lands $b = 0$, a random number $c^\star \in \{0,1\}^l$ is chosen. In the end, the challenge $c^\star$ is returned to $\mathcal{D}$.
4) After receiving $c^\star$, $\mathcal{D}$ computes the encrypted packet C as follows:

$$C = \begin{array}{|c|c|c|}\hline 0 & r & m \bigoplus c^\star \\ \hline \end{array}$$

and sends C to the adversary $\mathcal{A}$.
5) At the end, $\mathcal{A}$ returns a number j, $1 \leq j \leq n - i_h$, to $\mathcal{D}$. $\mathcal{A}$ returns 0 if it cannot identify the source of the packet C. The distinguisher $\mathcal{D}$ outputs $b = 1$ if $j = w$, outputs 0 if $j = 0$, and outputs $1/0$ with equal probability otherwise, respectively.

Since $\mathcal{A}$ has the advantage $\mathbf{Adv}_{\mathcal{A}} = \epsilon$ to crack the forward-secure privacy

$$\mathbf{Adv}_{\mathcal{A}} = (n - i_h) \times \Pr\left[\mathcal{A}(m, C) = w\right] - 1 \tag{7}$$

we have

$$\begin{aligned} \Pr\left[\mathcal{A}(m, C) = w\right] &= \frac{1}{n - i_h} + \frac{\mathbf{Adv}_{\mathcal{A}}}{n - i_h} \\ &= \frac{1}{n - i_h} + \frac{\epsilon}{n - i_h}. \end{aligned} \tag{8}$$

In sequence

$$\begin{aligned} &\Pr\left[\mathcal{D}(c, r) = b | b = 1\right] \\ &\quad = \Pr\left[\mathcal{D}(c, r) = b | b = 1, \mathcal{A}(m, C) = \omega\right] \\ &\qquad + \Pr\left[\mathcal{D}(c, r) = b | b = 1, \mathcal{A}(m, C) \neq \omega, \neq 0\right] \\ &\quad \geq 1 \cdot \left(\frac{1}{n - i_h} + \frac{\epsilon}{n - i_h}\right) \\ &\qquad + \frac{1}{2} \cdot \left(1 - \frac{1}{n - i_h} - \frac{\epsilon}{n - i_h}\right) \\ &\quad \geq \frac{1}{2} + \frac{1}{2(n - i_h)} + \frac{\epsilon}{2(n - i_h)}. \end{aligned} \tag{9}$$

If $b = 0$, all MUs are equally taken from the viewpoint of $\mathcal{A}$, and $\mathcal{A}$ can do no better than random guessing.

Averaging over $\mathcal{D}$'s random choices of w, $1 \leq w \leq n - i_h$, we obtain

$$\begin{aligned}
&\Pr\left[\mathcal{D}(c,r) = b | b = 0\right] \\
&\quad = \Pr\left[\mathcal{D}(c,r) = b | b = 0, \mathcal{A}(m,C) = \omega\right] \\
&\qquad + \Pr\left[\mathcal{D}(c,r) = b | b = 0, \mathcal{A}(m,C) \neq \omega\right] \\
&\quad \geq 0 \cdot \frac{1}{n - i_h} + \frac{1}{2} \cdot \left(1 - \frac{1}{n - i_h}\right) \\
&\quad \geq \frac{1}{2} - \frac{1}{2(n - i_h)}. \qquad (10)
\end{aligned}$$

By combining the results in (9) and (10), we will have

$$\begin{aligned}
\Pr\left[\mathcal{D}(c,r) = b\right] &= \frac{1}{2} \cdot \Pr\left[\mathcal{D}(c,r) = b | b = 1\right] \\
&\quad + \frac{1}{2} \cdot \Pr\left[\mathcal{D}(c,r) = b | b = 0\right] \\
&\geq \frac{1}{2} \cdot \left(\frac{1}{2} + \frac{1}{2(n - i_h)} + \frac{\epsilon}{2(n - i_h)}\right) \\
&\quad + \frac{1}{2} \cdot \left(\frac{1}{2} - \frac{1}{2(n - i_h)}\right) \\
&= \frac{1}{2} + \frac{\epsilon}{4(n - i_h)}. \qquad (11)
\end{aligned}$$

Then

$$\begin{aligned}
\mathbf{Adv}_{\pi,\mathcal{D}}(k,l) &= 2 \times \Pr[b' = b] - 1 \\
&= 2 \times \Pr\left[\mathcal{D}(c,r) = b\right] - 1 \\
&\geq 2 \times \left(\frac{1}{2} + \frac{\epsilon}{4(n - i_h)}\right) - 1 \\
&= \frac{\epsilon}{2(n - i_h)}. \qquad (12)
\end{aligned}$$

Therefore, the distinguisher $\mathcal{D}$ can break the *semantic security* of the symmetric encryption π with a nonnegligibly advantage probability $\mathbf{Adv}_{\pi,\mathcal{D}}(k,l) \geq \epsilon/2(n - i_h)$, whereas the cost time is almost the same as that spent by the adversary $\mathcal{A}$, i.e., $\tau' \approx \tau$, which contradicts the assumption that the symmetric encryption π is semantic secure. Therefore, it deduces that the perfect forward location privacy of each packet is provably secure in our security model. Because the security proof is carried in the extreme case, the perfect forward location privacy should hold in other cases where not all the MUs are compromised.

B. Secure Against the DoS Attacks

An active attacker $\mathcal{A}$ can launch possible DoS attacks without being tracked, because of the location privacy. The proposed protocol is also secure against the DoS attacks to some extent, which will be discussed as follows. If the attacker $C = \boxed{\mathcal{A}}\boxed{\mathbf{RID}}\boxed{c}$, according to A3, each node in the same link-layer domain would not waste the battery power to process it because the random identifier $\mathbf{RID}$ does not exist in its PP.

TABLE V
CRYPTOGRAPHIC OPERATION'S EXECUTION TIME

	Descriptions	Execution Time
T_{pm}:	The time for one point multiplication in $\mathbb{G}$	0.6 ms
T_{mh}:	The time for one MapToPoint hash	3.9 ms
T_{pa}:	The time for one pairing operation	4.5 ms

TABLE VI
COMPUTATION PERFORMANCE

	MU	AP
Time Complexity	$2T_{\mathsf{pm}} + T_{\mathsf{mh}} + 2T_{\mathsf{pa}}$ / T_{pm}	$2T_{\mathsf{pm}} + T_{\mathsf{mh}} + 2T_{\mathsf{pa}}$ / T_{pm}
Rough Overhead	14.1 ms / 0.6 ms	14.1 ms / 0.6 ms

$\mathcal{A}$ could randomly choose n_r random identifier $\mathbf{RID}$'s from the domain $\{0,1\}^k$ and launch a DoS attack by broadcasting n_r bogus packets. Based on our *Preset in Idle* technique, the total number of random identifiers stored in all PPs is $n \cdot s_p$ on average. In this case, the probability of at least one $\mathbf{RID}$ happening to exist in some MU's PP can be computed as follows:

$$\Pr(\mathcal{A}) = 1 - \frac{\binom{n \cdot s_p}{0}\binom{2^k - n \cdot s_p}{n_r}}{\binom{2^k}{n_r}} = \frac{\binom{2^k}{n_r} - \binom{2^k - n \cdot s_p}{n_r}}{\binom{2^k}{n_r}}. \qquad (13)$$

Clearly, since $2^k \gg n \cdot s_p$ and $2^k \gg n_r$, the term $\binom{2^k}{n_r} - \binom{2^k - n \cdot s_p}{n_r}$ almost approaches 0; thus, the probability $\Pr(\mathcal{A})$ in (13) is negligible. Therefore, the DoS attack is also invalid.

VI. PERFORMANCE EVALUATION

In this section, we examine the performance of the proposed protocol, where the anonymous mutual authentication phase is evaluated first, followed by the analysis on the subsequent sending/receiving packet phase.

A. Evaluation on the Anonymous Mutual Authentication Phase

Since the scalar point multiplication in $\mathbb{G}$, MapToPoint hash, and pairing computations dominate each party's computation overhead in the anonymous mutual authentication phase, we only count the number of these operations in the assessment of computation performance. Table V gives the observed processing time (in milliseconds) for an MNT curve [26] of embedding degree $k_d = 6$ and 160-bit q. The implementation was executed on a 3.0-GHz Intel Pentium 4 machine [31]. Table VI outlines the computation performance results, where the data before "/" denote the total computation time performed by each party, and the data following "/" denote the computation time that can be performed offline. By considering the overhead estimated in Table VI, the proposed anonymous mutual authentication protocol between each MU and the AP is expected to readily support the targeted wireless applications.

TABLE VII
EVALUATION FACTORS' EXECUTION TIME

Execution Time	Conditions
$t_p = 0.071$ ms	commun. bandwidth 54 Mbps, $n_s = 500$ bytes
$t_p = 0.028$ ms	commun. bandwidth 54 Mbps, $n_s = 200$ bytes
$t_p = 0.011$ ms	commun. bandwidth 54 Mbps, $n_s = 80$ bytes
$t_p = 0.003$ ms	commun. bandwidth 54 Mbps, $n_s = 20$ bytes
$t_\pi = 29.934$ MBytes/sec	RC5 operation based on cryptographic library MIRACL [32]

B. Analysis on the Sending/Receiving Packet Phase

In this section, we evaluate the performance in the sending/receiving packet phase in terms of the overall packet delay. We compare the proposed protocol with the ordinary MAC address marked protocol (denoted as the Type-I protocol) and previous address encrypted protocol [14] (denoted as the Type-II protocol). The evaluation factors include several items: n_p denotes the total number of sending/receiving packets per node within a given period; n_s denotes the packet size; t_p is the latency of sending/receiving one packet; t_π is the process speed of the symmetric encryption π; and s_p and λ are the size of the PP and the maximal number of possible lost packets, respectively, as discussed in Section IV-D1. Table VII evaluates the execution time of t_p and t_π running on a 3.0-GHz Intel Pentium 4 machine with 1-GB RAM.

Next, we discuss the total delay of the three protocols.

1) In the Type-I protocol, because there is no encryption/decryption computation, sending/receiving n_p packets requires

$$T_1 = t_p \times n_p. \tag{14}$$

2) In the Type-II protocol, because each packet requires encryption/decryption computation, the evaluation factors n_s and t_π should be taken into consideration. Therefore, sending/receiving n_p packets requires

$$T_2 = \left(t_p + \frac{n_s}{t_\pi}\right) \times n_p. \tag{15}$$

3) In our protocol, since some encryption/decryption values have been offline computed and stored in the PP, when the number of sent/received packets is less than $s_p - \lambda$, the process is as fast as the Type-I protocol. On the other hand, if the number n_p is larger than $s_p - \lambda$, the delay due to subsequent $(n_p - s_p + \lambda)$ packets is almost the same as that in the Type-II protocol. Therefore, sending/receiving n_p packets in our protocol requires

$$T_{\text{our}} = \begin{cases} t_p \times n_p, & \text{if } n_p < s_p - \lambda \\ t_p \times (s_p - \lambda) + \left(t_p + \frac{n_s}{t_\pi}\right) & \\ \quad \times (n_p - s_p + \lambda), & \text{otherwise.} \end{cases} \tag{16}$$

Note that the time taken by hash operations and pseudorandom number generation for k-bit inputs are not taken into consideration in each invocation of PresetInIdle(), since they are negligibly compared with that of symmetric encryption on l-bit payload inputs.

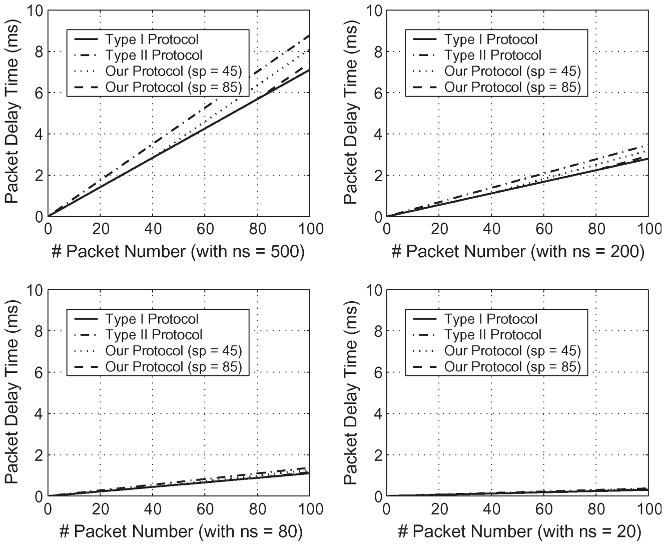


Fig. 6. Relation between the packet delay time and the packet number n_p.

With packet size $n_s = 500$, 200, 80, and 20 bytes, respectively, Fig. 6 shows the packet delay of the Type-I, Type-II, and proposed protocols. The packet delay of our protocol is almost the same as that of the Type-I protocol when the packet number of packets is less than $s_p - \lambda$. Furthermore, the larger the PP size s_p is set, the better efficiency our protocol achieves. In addition, Fig. 6 also shows the impact of the packet size on the efficiency of our protocol. We can see that for the same n_p and s_p, the larger the packet size n_s, the more salient the performance advantage of our protocol over the Type-II protocol. However, when the packet size is small, since the required time costs on encrypting packet are also reduced, the difference of these protocols on sending packets is not obvious.

Assume that n_p packets are broadcasted over the same link-layer domain, among which n_{pi} packets' destination is the MU denoted as N_i, where $0 \le n_{pi} \le n_p$. Then, we define the valid packet number as n_{pi}, and the valid packet ratio of N_i is defined as

$$\rho_i = \frac{n_{pi}}{n_p}. \tag{17}$$

In the Type-II protocol, because of the location privacy, an MU cannot tell that a packet is destined for *it* until the MU decrypts the packet. Therefore, when n_p packets come, the time cost is

$$T_2 = t_p \times n_p \tag{18}$$

which is irrelevant to the valid packet ratio ρ_i. However, in our protocol, the MU can predetermine his packet by looking up the PP, which yields the time cost as

$$T_{\text{our}} = \begin{cases} t_p \times n_p \times \rho_i, & \text{if } n_p \times \rho_i < s_p - \lambda \\ t_p \times (s_p - \lambda) + \left(t_p + \frac{n_s}{t_\pi}\right) & \\ \quad \times (n_p \times \rho_i - s_p + \lambda), & \text{otherwise.} \end{cases} \tag{19}$$

With $n_p = 100$, $s_p = 60$, $\lambda = 5$, and $t_\pi = 29.934$ MB/s, we compare the time costs of the Type-II protocol with that of our

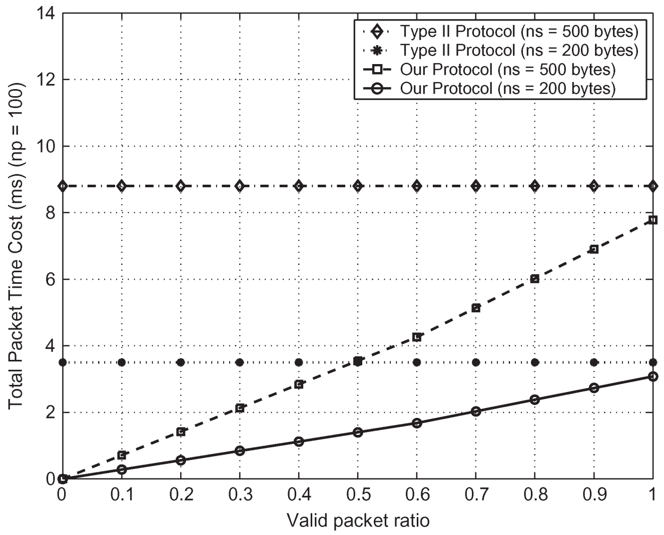


Fig. 7. Relation between the total packet time cost and the valid packet ratio ρ_i.

protocol for varying ρ_i. It is observed that when the valid packet $n_{pi} < s_p - \lambda$, our protocol is very efficient. However, the time cost of the Type-II protocol is high, particularly when ρ_i is low. Furthermore, the bigger the packet size, the higher the time cost in the Type-II protocol that is observed (Fig. 7).

VII. Conclusion

In this paper, we have introduced a formal security model on link-layer forward-secure location privacy that aims to achieve anonymous communication in wireless networks. Compared with previously reported counterparts, our model on location privacy is characterized by the fact that an attacker still cannot learn how long an MU has stayed at the current location, although he corrupts the MU's current location privacy. Based on the security model, a novel anonymous mutual authentication protocol between the AP and each MU has been proposed by considering the advantages of location- and time-aware keys. To the best of our knowledge, this is the first practical anonymous mutual authentication protocol for wireless communications. We have also developed a forward-secure location privacy protocol at the link layer and proved that the location privacy is tightly related to the symmetric encryption semantic security according to the provable security technique. With the help of the *Preset in Idle* technique, our protocol has been demonstrated to be efficient through extensive performance evaluation.

References

[1] D. R. Cheriton and M. Gritter, "Triad: A scalable deployable NAT-based Internet architecture," Comput. Sci. Dept., Stanford Univ., Stanford, CA, Jan. 2000. Tech. Rep.

[2] J. Girao, B. Lamparter, M. Liebsch, and T. Melia, "A practical approach to provide communication privacy," in *Proc. IEEE ICC*, Istanbul, Turkey, Jun. 2006, vol. 5, pp. 1965–1970.

[3] P. Nikander, J. Arkko, and B. Ohlman, "Host identity indirection infrastructure (Hi3)," in *Proc. 2nd SNCSW*, Nov. 2004.

[4] X. Lin, R. Lu, P.-H. Ho, X. Shen, and Z. Cao, "TUA: A novel compromise-resilient authentication architecture for wireless mesh networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1389–1399, Apr. 2008.

[5] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," in *Proc. ACM SIGCOMM Conf.*, Aug. 2002, pp. 73–88.

[6] M. Shi, H. Rutagemwa, X. Shen, J. W. Mark, and A. Saleh, "A service-agent-based roaming architecture for WLAN/cellular integrated networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 3168–3181, Sep. 2007.

[7] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE INFOCOM*, Phoenix, AZ, Apr. 15–17, 2008, pp. 246–250.

[8] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. IEEE INFOCOM*, Phoenix, AZ, Apr. 15–17, 2008, pp. 1229–1237.

[9] K. Ren, W. Lou, R. H. Deng, and K. Kim, "A novel privacy preserving authentication and access control scheme for pervasive computing environments," *IEEE Trans. Veh. Technol.*, vol. 55, no. 4, pp. 1373–1384, Jul. 2006.

[10] M. Gruteser and D. Grunwald, "Enhancing location privacy in wireless LAN through disposable interface identifiers: A quantitative analysis," *Mobile Netw. Appl.*, vol. 10, no. 3, pp. 315–325, Jun. 2005.

[11] L. Huang, K. Matsuura, H. Yamane, and K. Sezaki, "Enhancing wireless location privacy using silent period," in *Proc. IEEE WCNC*, New Orleans, LA, Mar. 2005, vol. 2, pp. 1187–1192.

[12] F. L. Wong and F. Stajano, "Location privacy in Bluetooth," in *Proc. 2nd ESAS*. New York: Springer-Verlag, 2005, vol. 3813, pp. 176–188.

[13] X. Lin, X. Sun, P. H. Ho, and X. Shen, "GSIS: A secure and privacy-preserving protocol for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3442–3456, Nov. 2007.

[14] F. Armknecht, J. Girao, A. Matos, and R.L. Aguiar, "Who said that? Privacy at link layer," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007, pp. 2521–2525.

[15] D. Chaum and E. van Heyst, "Group signatures," in *Proc. Advances Cryptology—EUROCRYPT*. New York: Springer-Verlag, 1991, vol. 547, pp. 257–265.

[16] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *Proc. Advances Cryptology—CRYPTO*. New York: Springer-Verlag, 1997, vol. 1294, pp. 410–424.

[17] J. Camenisch and M. Michels, "A group signature scheme with improved efficiency," in *Proc. Advances Cryptology—ASIACRYPT*. New York: Springer-Verlag, 1998, vol. 1514, pp. 160–174.

[18] M. Bellare, D. Micciancio, and B. Warinschi, "Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions," in *Proc. Advances Cryptology—EUROCRYPT*. New York: Springer-Verlag, 2003, vol. 2656, pp. 630–648.

[19] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Advances Cryptology—CRYPTO*. New York: Springer-Verlag, 2001, vol. 2139, pp. 213–229.

[20] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," in *Proc. Advances Cryptology—ASIACRYPT*. New York: Springer-Verlag, 2001, vol. 2248, pp. 514–532.

[21] D. H. Phan and D. Pointcheval, "About the security of ciphers," in *Proc. Workshop Sel. Areas Cryptography*. New York: Springer-Verlag, 2004, vol. 3357, pp. 185–200.

[22] W. Mao, *Modern Cryptography: Theory and Practice*. Upper Saddle River, NJ: Prentice–Hall PTR, 2003.

[23] Z. Chai, Z. Cao, and R. Lu, "Remote authentication with forward security," in *Proc. ATC*. New York: Springer-Verlag, 2006, vol. 4158, pp. 418–427.

[24] R. Lu, Z. Cao, and X. Dong, "Authenticated encryption protocol with perfect forward secrecy for mobile communication," *Wireless Commun. Mobile Comput.*, vol. 6, no. 3, pp. 273–280, May 2006.

[25] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proc. Advances Cryptology—CRYPTO*. New York: Springer-Verlag, 2004, vol. 3152, pp. 41–55.

[26] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction," *IEICE Trans. Fundam.*, vol. E84-A, no. 5, pp. 1234–1243, 2001.

[27] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Proc. Advances Cryptology—CRYPTO*. New York: Springer-Verlag, 1994, vol. 773, pp. 232–249.

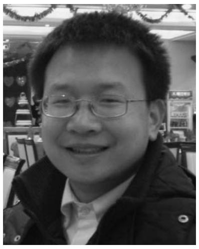
[28] C. G. Gunther, "An identity-based key-exchange protocol," in *Proc. Advances Cryptology—EUROCRYPT*. New York: Springer-Verlag, 1990, vol. 434, pp. 29–37.

[29] M. Bellare and S. K. Miner, "A forward-secure digital signature scheme," in *Proc. Advances Cryptology—CRYPTO*. New York: Springer-Verlag, 1999, vol. 1666, pp. 431–448.

[30] R. P. Brent, "Fast and reliable random number generators for scientific computing," in *Proc. PARA*. New York: Springer-Verlag, 2006, vol. 3732, pp. 1–10.

[31] M. Scott, *Efficient Implementation of Crytographic Pairings*. [Online]. Available: http://ecrypt-ss07.rhul.ac.uk/Slides/Thursday/mscottsamos07.pdf

[32] in *Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL)*. [Online]. Available: http://indigo.ie/~mscott/



Rongxing Lu received the B.Sc. and M.Sc. degrees in computer science from Tongji University, Shanghai, China, in 2000 and 2003, respectively, and the Ph.D. degree in computer science from Shanghai Jiao Tong University in 2006.

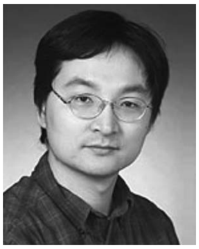
He is currently a Postdoctoral Fellow with the University of Waterloo, Waterloo, ON, Canada. His current research interests include wireless network security and cryptography.

Xiaodong Lin (S'07) is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

He is currently a Research Assistant with the Broadband Communications Research Group, Department of Electrical and Computer Engineering, University of Waterloo. His research interest includes wireless network security, applied cryptography, and anomaly-based intrusion detection.

Haojin Zhu received the B.Sc. degree in computer science from Wuhan University, Wuhan, China, in 2002 and the M.Sc. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2005. He is currently working toward the Ph.D. degree in electrical and computer engineering with the University of Waterloo, Waterloo, ON, Canada.

His current research interests include wireless network security and applied cryptography.



Pin-Han Ho (M'04) received the B.Sc. and M.Sc. degrees from the National Taiwan University, Taipei, Taiwan, in 1993 and 1995, respectively, and the Ph.D. degree from Queens University, Kingston, ON, Canada, in 2002, focusing on optical communications systems, survivable networking, and quality-of-service routing problems.

In 2002, he joined the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, as an Assistant Professor. He is the author or coauthor of more than 100 refereed technical papers and book chapters and the coauthor of a book on optical networking and survivability.

Prof. Ho was a recipient of the Distinguished Research Excellent Award from the Department of Electrical and Computer Engineering, University of Waterloo; the Early Researcher Award (Premier Research Excellence Award) in 2005; the Best Paper Award in the 2002 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'02), the 2005 IEEE International Conference on Communications (ICC'05) Optical Networking Symposium, the 2007 IEEE International Conference on Communications (ICC'07) Security and Wireless Communications Symposium; and the Outstanding Paper Award at the 2002 Workshop on High-Performance Switching and Routing (HPSR'02).

Xuemin (Sherman) Shen (M'97-SM'02) received the B.Sc. degree in electrical engineering from Dalian Maritime University, Dalian, China, in 1982 and the M.Sc. and Ph.D. degrees in electrical engineering from Rutgers University, Camden, NJ, in 1987 and 1990, respectively.

He is a Professor, the University Research Chair, and the Associate Chair for Graduate Studies with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on mobility and resource management in interconnected wireless/wireline networks, UWB wireless communications systems, wireless security, and ad hoc and sensor networks. He is the coauthor of three books and has published more than 300 papers and book chapters on wireless communications and networks, control, and filtering.

Dr. Shen serves as the Technical Program Committee Chair for the IEEE Globecom'07, General Co-Chair for Chinacom'07 and QShine'06, and Founding Chair for the IEEE Communications Society Technical Committee on P2P Communications and Networking. He also serves as a Founding Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the Editor-in-Chief for *Peer-to-Peer Networking and Application*, an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the *KICS/IEEE Journal of Communications and Networks*, *Computer Networks*, *ACM/Wireless Networks*, and *Wireless Communications and Mobile Computing (Wiley)*, etc. He has also served as a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, and *IEEE Communications Magazine*. He was a recipient of the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award in 2004 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, and the Distinguished Performance Award in 2002 from the Faculty of Engineering, University of Waterloo. He is a Registered Professional Engineer in the Province of Ontario.