PEACE: A Novel Privacy-Enhanced Yet Accountable Security Framework for Metropolitan Wireless Mesh Networks

Kui Ren, *Member*, *IEEE*, Shucheng Yu, *Student Member*, *IEEE*,
Wenjing Lou, *Senior Member*, *IEEE*, and Yanchao Zhang, *Member*, *IEEE*

Abstract—Recently, multihop wireless mesh networks (WMNs) have attracted increasing attention and deployment as a low-cost approach to provide broadband Internet access at metropolitan scale. Security and privacy issues are of most concern in pushing the success of WMNs for their wide deployment and for supporting service-oriented applications. Despite the necessity, limited security research has been conducted toward privacy preservation in WMNs. This motivates us to develop PEACE, a novel Privacy-Enhanced yet Accountable seCurity framEwork, tailored for WMNs. On one hand, PEACE enforces strict user access control to cope with both free riders and malicious users. On the other hand, PEACE offers sophisticated user privacy protection against both adversaries and various other network entities. PEACE is presented as a suite of authentication and key agreement protocols built upon our proposed short group signature variation. Our analysis shows that PEACE is resilient to a number of security and privacy related attacks. Additional techniques were also discussed to further enhance scheme efficiency.

Index Terms—Wireless mesh networks, privacy, authentication, security, anonymous communication.

1 Introduction

Wireless mesh networks (WMNs) have recently attracted increasing attention and deployment as a promising low-cost approach to provide last-mile high-speed Internet access at metropolitan scale [2], [3]. Typically, a WMN is a multihop layered wireless network as shown in Fig. 1 [4], [5]. The first layer consists of access points, which are high-speed wired Internet entry points. At the second layer, stationary mesh routers form a multihop backbone via long-range high-speed wireless techniques such as WiMAX [6]. The wireless backbone connects to wired access points at some mesh routers through high-speed wireless links. The third layer consists of a large number of mobile network users. These network users access the network either by a direct wireless link or through a chain of other peer users to a nearby mesh router. WMNs represent *a unique marriage of the ubiquitous coverage of wide area cellular networks with the ease and the speed of local area Wi-Fi networks* [4]. The advantages of WMNs also include low deployment costs, self-configuration and self-maintenance, good scalability, high robustness, etc. [2].

Security and privacy issues are of most concern in pushing the success of WMNs for their wide deployment and for supporting service-oriented applications. Due to the intrinsically open and distributed nature of WMNs, it is essential to enforce network access control to cope with both free riders and malicious attackers. Dynamic access to WMNs should be subject to successful user authentication based on the properly preestablished trust between users and the network operator; otherwise, network access should be prohibited. On the other hand, it is also critical to provide adequate provisioning over user privacy as WMN communications usually contain a vast amount of sensitive user information. The wireless medium, open network architecture, and lack of physical protection over mesh routers render WMNs highly vulnerable to various privacy-oriented attacks. These attacks range from passive eavesdropping to active message phishing, interception, and alteration, which could easily lead to the leakage of user information. Obviously, the wide deployment of WMNs can succeed only after users are assured for their ability to manage privacy risks and maintain their desired level of anonymity.

The necessity of security and privacy in WMNs can be well illustrated through the following example. In a metro-scale community mesh network, the citizens access WMNs from everywhere within the community such as offices, homes, restaurants, hospitals, hotels, shopping malls, and even vehicles. Through WMNs, they access the public Internet in different roles and contexts for services like e-mails, e-banking, e-commerce, and Web surfing, and also interact with their local peers for file sharing, teleconferencing, online gaming, instant chatting, etc. Integrated with sensors and cameras, the WMN may also be used to collect information of interest. In fact, at Boston suburb area, the

- K. Ren is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, 3301 Dearborn St, Chicago, IL 60616. E-mail: kren@ece.iit.edu.
- S. Yu and W. Lou are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609. E-mail: {yscheng, wjlou}@ece.wpi.edu.
- Y. Zhang is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, University Heights, Newark, NJ 07102. E-mail: yczhang@njit.edu.

Manuscript received 30 Oct. 2008; revised 26 Feb. 2009; accepted 18 Mar. 2009; published online 26 Mar. 2009.
Recommended for acceptance by P. Mohapatra.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2008-10-0399.
Digital Object Identifier no. 10.1109/TPDS.2009.59.

1045-9219/10/$26.00 © 2010 IEEE Published by the IEEE Computer Society

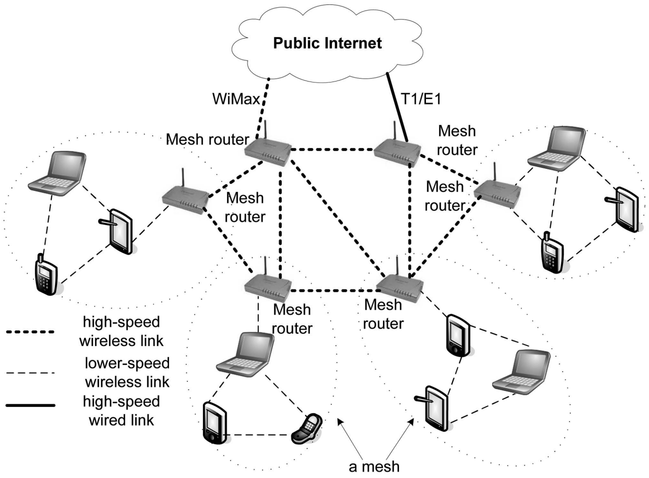

Fig. 1. WMN network architecture [4], [5].

City of Malden [7], the police department will use the WMN "to stream video footage from local areas directly to the police station, making it easier for police officers to monitor and respond to crimes at those locations" [7]. Obviously, all these communications contain various kinds of sensitive user information like personal identities, activities, location information, financial information, transaction profiles, social/business connections, and so on. Once disclosed to the attackers, these information could compromise any user's privacy, and when further correlated together, can cause even more devastating consequences. Hence, securing user privacy is of paramount practical importance in WMNs. Moreover, for both billing purpose and avoiding abuse of network resources, it is also essential to prohibit free riders and let only legitimate residents access WMNs.

Despite the necessity and importance, limited research has been conducted to address privacy-enhanced security mechanisms in WMNs. This motivates us to propose PEACE, a novel Privacy-Enhanced yet Accountable seCurity framEwork for WMNs. Our contributions are fourfold as follows:

Security: It achieves explicit mutual authentication and key establishment between users and mesh routers and between users themselves. It, thus, prohibits both illegal network access from free riders and malicious users and phishing attacks due to rogue mesh routers.

Anonymity: It simultaneously enables unilateral anonymous authentication between users and mesh routers and bilateral anonymous authentication between any two users. It, thus, ensures user anonymity and privacy.

Accountability: It enables user accountability, at regulating user behaviors and protecting WMNs from being abused and attacked. Network communications can always be audited in the cases of disputes and frauds. It further allows dynamic user revocation so that malicious users can be evicted.

Sophisticated user privacy: It allows users to disclose minimum information possible while maintaining accountability. In PEACE, the user identity is a multifaceted information as network users as society members always interact with WMNs in different roles and contexts. Therefore, a dispute regarding a given communication session should only be attributed according to the role/context information of the user without disclosing his full identity information (unless necessary).

To the best of our knowledge, PEACE is the first attempt to establish an accountable security framework with a sophisticated privacy protection model tailored for WMNs. PEACE also lays a solid background for designing other upper layer security and privacy solutions, e.g., anonymous communication.

The rest of the paper is organized as follows: Section 2 is the introduction of the cryptographic knowledge entailed by PEACE. Section 3 describes the problem formulation. Then, in Section 4, the details of PEACE are described. We further analyze in Section 5 the security and privacy properties of PEACE, as well as its performance. Section 6 is about related work. Finally, we conclude the paper in Section 7.

2 The Cryptographic Background

2.1 Bilinear Groups

We first introduce a few concepts related to bilinear maps as they are important to the design of PEACE. Let $\mathbb{G}_1, \mathbb{G}_2$ be multiplicative cyclic groups generated by g_1 and g_2, respectively, whose orders are a prime p, and $\mathbb{G}_T$ be a cyclic multiplicative group with the same order p. Suppose that there is an efficient and computable isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ such that $\psi(g_2) = g_1$. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be a bilinear pairing with the following properties [8]:

- **Bilinearity:** $e(aP, bQ) = e(P, Q)^{ab}$ for all $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2, a, b \in \mathbb{Z}_p$.
- **Nondegeneracy:** $e(g_1, g_2) \neq 1$.
- **Computability:** There is an efficient algorithm to compute $e(P, Q)$ for all $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$.

In this paper, we only use the fact that $\mathbb{G}_1, \mathbb{G}_2$ can be of size approximately 2^{170}, elements in $\mathbb{G}_1$ are 171-bit strings, and discrete log in $\mathbb{G}_1, \mathbb{G}_2$ is as hard as discrete log in $\mathbb{Z}_q^*$, where q is a 1,020-bit prime number.

2.2 Group Signature

Group signature schemes are a relatively recent cryptographic concept introduced by Chaum and van Heyst in 1991 [9]. A group signature scheme is a method for allowing a member of a group to sign a message on behalf of the group. In contrast to ordinary signatures, it provides anonymity to the signer, i.e., a verifier can only tell that a member of some group signed. However, in exceptional cases, such as a legal dispute, any group signature can be "opened" by a designated group manager to reveal unambiguously the identity of the signature's originator. Some group signature schemes support revocation, where group membership can be disabled. One of the most recent group signature schemes is the one proposed by Boneh and Shacham [8], which has a very short signature size that is comparable to that of an RSA-1024 signature [10]. This scheme is based on the following two problems that are believed to be hard. Let $\mathbb{G}_1, \mathbb{G}_2, g_1, g_2$ as defined above.

q-Strong Diffie-Hellman problem: The q-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is defined as follows: given a $(q+2)$-tuple $(g_1, g_2, g_2^{\gamma}, g_2^{(\gamma^2)}, \ldots, g_2^{(\gamma^q)})$ as input, output a pair $(g_1^{1/(\gamma+x)}, x)$, where $x \in \mathbb{Z}_p^*$.

Decision linear on $\mathbb{G}_1$: Given arbitrary generators u, v, h of $\mathbb{G}_1$ and $u^a, v^b, h^c \in \mathbb{G}1$ as input, output yes if $a + b = c$, and no, otherwise.

3 Problem Formulation and the Scheme Overview

3.1 Network Architecture and System Assumptions

The three-layer architecture in Fig. 1 considers a metropolitan-scale WMN under the control of a network operator (*NO*). The network operator deploys a number of APs and mesh routers and forms a well-connected WMN that covers the whole area of a city and provides network services to network users, i.e., the citizens. Network users, on the other hand, subscribe to the network operator for the services and utilize their mobile clients to freely access the network from anywhere within the city. The membership of network users may be 1) terminated/renewed according to user-operator agreement in a periodic manner or 2) dynamically revoked by *NO* in case of dispute/attack.

Similar to [4], [11], we assume that the downlink from a mesh router to all users within its coverage is one hop. However, the uplink from a user to a mesh router may be one or multiple hops. That is, a network user needs to transmit packets in multiple hops to a mesh router beyond his direct transmission range. In this case, network users cooperate with each other on relaying the packets to mesh routers. We further assume that all the network traffic has to go through a mesh router except the communication between two direct neighboring users. We assume so as it is expected that communications to and from a mesh router will constitute the majority of traffic in a WMN [12]. Moreover, this assumption would significantly reduce the routing complexity from the users' point of view as mesh routers will take the responsibility.

We assume that *NO* can always communicate with mesh routers through preestablished secure channels, and so are mesh themselves. The WMN is assumed to be deployed with redundancy in mind so that revocation of individual mesh routers will not affect network connection. We assume the existence of an offline trusted third party (*TTP*), which is trusted for not disclosing the information it stores. *TTP* is required only during the system setup. We further assume that there is a secure channel between *TTP* and each network user.

3.2 Threat Model and Security Requirements

Due to the open medium and spatially distributed nature, WMNs are vulnerable to both passive and active attacks. The passive attacks include eavesdropping, while active attacks range from message replaying, bogus message injection, phishing, active impersonation to mesh router compromise. Hence, for a practical threat model, we consider an adversary that is able to eavesdrop all network communications, as well as inject arbitrary bogus messages. In addition, the adversary can compromise and control a small number of users and mesh routers subject to his choice; it may also set up rogue mesh routers to phish user accesses. The purposes of the adversary include 1) illegal and unaccountable network access, 2) the privacy of legitimate network users, and 3) denial-of-service (DoS) attacks against service availability.

In light of the above threat model, the following security requirements are essential to ensure that a WMN functions correctly and securely as purposed.

- *User-router mutual authentication and key agreement*: A mesh router and a user should mutually authenticate each other to prevent both unauthorized network access and phishing attacks. The user and the mesh router should also establish a shared pairwise symmetric key for session authentication and message encryption.
- *User-user mutual authentication and key agreement*: Users should also authenticate each other before cooperation in regard to message relaying and routing. Moreover, symmetric keys should be established and effectively maintained to provide session authentication and message encryption over the corresponding traffic.
- *Sophisticated user privacy protection*: The privacy of users should be well protected, and we differentiate user privacy against different entities such as the adversary, *NO*, and the law authority, as will be elaborated in the next section.
- *User accountability*: In the cases of attacks and disputes, the responsible users and/or user groups should be able to be audited and pinpointed. On the other hand, no innocent users can be framed for disputes/attacks they are not involved in.
- *Membership maintenance*: The network should be able to handle membership dynamics including membership revocation, renewing, and addition.
- *DoS resilience*: The WMN should maintain service availability despite of DoS attacks.

3.3 Privacy Model

In a metropolitan WMN, city residents as network users access the WMN for services related to every aspect of their personal and professional lives. Inevitably, these network communications will contain a large amount of personal, business, and organization information that are highly sensitive and interested by different parties for different purposes. The malicious adversaries are interested as they could gain economic benefits by stealing the identity and other information. In fact, identity theft has been an infamous type of the Internet crimes. Furthermore, network communications accumulated over time and space may intentionally be collected and used for establishing user profiles by certain parties, including *NO*. These parties are not necessarily malicious, but such actions certainly violate user privacy. Obviously, the success deployment of WMNs is subject to users' assurance of their ability to manage privacy risks and maintain their desired level of anonymity.

The above observation leads to the establishment of a practical user privacy model, which provides sophisticated user privacy management and addresses user accountability simultaneously. We observe that a user usually accesses the WMN in different roles and under different contexts. For example, a user as an engineer may access the WMN in his office as an employee of a company. The same user may also

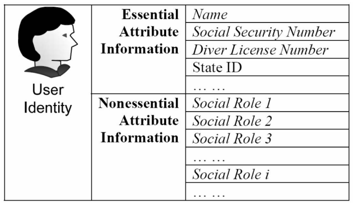

Fig. 2. The format of user identity information.

access the WMN from a university campus as a student, from a rented apartment as a tenant, and from a golf club as a paid member, and so on. In our privacy model, we, hence, refer to the *user identity* as a user's collective attribute information according to his different roles in the society. In the above example, the user identity may include

{*name, ssn, engineer of company X, tenant of apartment Y, student of university Z, member of golf club V*, …}.

Informally, we can divide the user identity information into two different categories, that is, *essential attribute information* and *nonessential attribute information* as shown in Fig. 2. The *essential attribute information* includes all the information that can be used uniquely to identify a specific user such as user's name, social security number, driver license number, passport number, etc. On the other hand, the *nonessential attribute information* of a user may include his different social roles as indicated in the above example. We note that if *essential attribute information* of a user is disclosed, this user is fully exposed and all his attribute information will also be disclosed. On the other hand, disclosing nonessential attribute information does not lead to the full exposure of the user's identity. That is, a user can still maintain a certain level of anonymity, when only his nonessential attribute information is disclosed. It is further observed that the nonessential attribute information of users are still sufficient for accountability purpose from *NO*'s perspective. This is because *NO* can still enforce network access control and audit network communications as it makes no difference to *NO* whether or not a responsible entity is a person, a company, or an organization, etc.

To protect the user privacy, the user identity information should be well protected from network communications against the adversary and even *NO*. Therefore, it should be required that

1. *no communication sessions should reveal any user identity information except that the user is a legitimate network user*;
2. *no entity including the adversary and NO could link two different communication sessions to the same particular user*.

Furthermore, in the cases of disputes and attacks, user privacy should be protected against *NO* in such a way that

3. *A given communication session under audit by NO can only be linked according to the attribute information of the user without disclosing his full identity information*. That is, only minimum necessary identity information is disclosed for the security purpose so that user privacy can be best protected.

Our privacy model further considers the extreme cases such as severe attacks in which the law authority has to track the particular responsible attacker. For this purpose, we introduce the concept of *user group* and try to utilize the natural society hierarchy among network users. A *user group* refers to any society entity, which, through a *user group manager*, manages a certain number of network users, i.e., its staffs and/or employees, and subscribes network services on behalf of its users. A *user group* can be any company, organization, university, or government agency, etc. A network user, on the other hand, usually belongs to multiple different *user groups* according to his roles in the society. In our privacy model, we further require that

4. *only by joint effort from both a user group manager and NO can a user's full identity be disclosed; and neither of them can do so alone*.

Note that the capability of a *user group manager* itself is strictly restricted, that is, it has no more ability than an ordinary network user. *User group managers* cannot link any communication session to a specific user by only them selves.

In summary, our privacy model is aimed at the following privacy guarantees under the threat model discussed above.

- Against the adversary, the *user group manager*s, and other entities (except *NO* and the law authority): At no circumstances, the adversary could tell that two different communication sessions are from the same network user or link a communication session to a specific user.
- Against *NO*: Given any communication session, *NO* can only tell which user group the corresponding user is from, but cannot recover user's full identity. That is, *NO* can only recover the corresponding nonessential user attribute information for the accountability purpose.
- For the law authority: With the help from both *NO* and *user group managers*, the law authority could link any communication session to the corresponding network user that is responsible.

3.4 Trust and Key Management Model

Given the security and privacy requirements discussed above, PEACE bases its design on the following trust and key management model. Fig. 3 is the high-level illustration of PEACE trust model, which consists of four kinds of network entities: the network operator, user group managers, users organized in groups, and a TTP. Each user group is a collection of users according to certain aspects of their nonessential attributes. For instance, a company is a user group consisting of all its employees, and all the tenants of an apartment is another user group maintained by the corresponding apartment management office. Each user group has one group manager responsible for adding and removing users. Before accessing the WMN, each user has to enroll in at least one user group whose manager, thus, knows both the essential and nonessential attributes of the user. In PEACE, users no longer directly register with the network operator; instead, each group manager subscribes to the network operator on behalf of its group members. Upon registration from a group manager, the network operator allocates a set of group secret keys to this

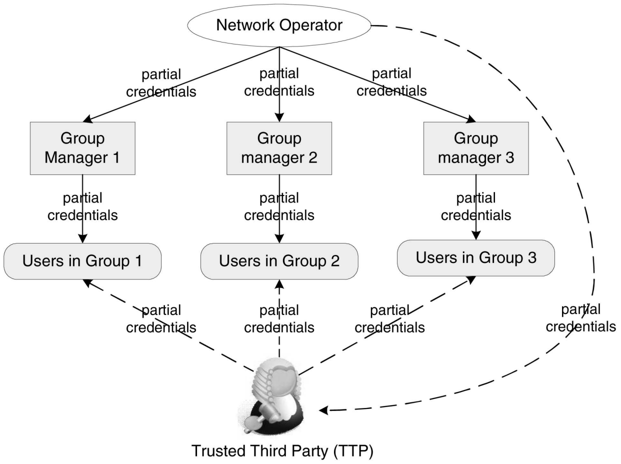

Fig. 3. PEACE trust model.

user group. Then, the network operator divides each group secret key into two parts, one part sent to the requesting group manager and the other part to the TTP. To access the WMN, each user requests one part of the group secret key from his group manager and the other part from the TTP to recover a complete group secret key. The user also needs to return signed acknowledgments to both the group manager and the TTP.

The above key management scheme is based on the principle of "separation of powers" and possesses a number of salient features. First, from network access control point of view, every legitimate user with a valid group secret key can generate a valid access credential, i.e., the signature of the authentication challenge—typically a nonce, upon request. The validity of this access credential can be verified by the network operator. Therefore, access security is guaranteed. Second, PEACE divides user identity information and their corresponding secret key information among three autonomous entities: the network operator, the group manager, and the TTP. In particular, the network operator knows the complete user secret key information, but not the mapping of the keys to the essential attributes of the users; the group manager or the TTP knows the essential attributes of the users, but not the complete secret key information. The system is designed in such a way that given an access credential submitted by a user, none of the network operator, the group manager, and the TTP can determine the user's essential attribute or compromise his privacy unless any two of them collude. User privacy is enhanced in this way. Finally, in case of service disputes or frauds, an authorized entity, such as a law enforcement authority, can collect information from the network operator, the user group manager, and the TTP to precisely identify the responsible user and hold him accountable. Therefore, user accountability can be attained as well.

4 PEACE: The Scheme

When designing PEACE, we find that none of the existing privacy-aware crytographic primitives, such as blind signature, ring signature, and group signature schemes, suits our purpose given the security and privacy requirements discussed above. Blind signature and ring signature schemes can only provide irrevocable anonymity, while PEACE demands user accountability, and hence, revocable anonymity. Existing group signature schemes do provide revocable anonymity, but cannot support sophisticated user privacy. This motivates us to tailor a group signature scheme to meet all the requirements. We, hence, develop a variation of the short group signature scheme proposed in [8] by modifying its key generation algorithm for our purpose. PEACE is then built on this new group signature variation by further integrating it into the authentication and key agreement protocol design.

4.1 Scheme Setup: Key Management

The following setup operations are performed in an offline manner by all the entities in PEACE, namely *NO*, a *TTP*, mesh routers, network users, and user group managers. PEACE works under bilinear groups $(\mathbb{G}_1, \mathbb{G}_2)$ with isomorphism ψ and respective generators g_1 and g_2, as in Section 2.1. PEACE also employs hash functions H_0 and H, with respective ranges $\mathbb{G}_2^2$ and $\mathbb{Z}_p$. The notation below mainly follows [8].

NO is responsible for key generation operation. Specifically, *NO* proceeds as follows:

1. Select a generator g_2 in $\mathbb{G}_2$ uniformly at random and set $g_1 \longleftarrow \psi(g_2)$. Select $\gamma \overset{R}{\longleftarrow} \mathbb{Z}_p^*$ and set $w = g_2^{\gamma}$.
2. Select
$$\mathrm{grp}_i \overset{R}{\longleftarrow} \mathbb{Z}_p^*$$
for a registered user group i.
3. Using γ, generate an SDH tuple $(\mathrm{A}_{i,j}, \mathrm{grp}_i, \mathrm{x}_j)$ by selecting $\mathrm{x}_j \overset{R}{\longleftarrow} \mathbb{Z}_p^*$ such that $\gamma + \mathrm{grp}_i + \mathrm{x}_j \neq 0$, and setting $\mathrm{A}_{i,j} \longleftarrow g_1^{1/(\gamma+\mathrm{grp}_i+\mathrm{x}_j)}$.
4. Repeat Step 3 for a predetermined number of times that are mutually agreed by *NO* and the user group manager GM_i.
5. Send $GM_i\{[i,j], \mathrm{grp}_i, \mathrm{x}_j \mid \forall j\}$ via a secure channel.
6. Repeat Steps 2, 3, and 4 for every user group.
7. Send *TTP*: $\{[i,j], \mathrm{A}_{i,j} \oplus \mathrm{x}_j \mid \forall i, j\}$ via a secure channel, where $\oplus$ denotes bitwise *exclusive OR* operation.[1]

The above operation generates the group public key *gpk* and a number of private keys *gsk*:

$$\begin{cases} gpk = (g_1, g_2, w), \\ \{gsk[i,j] = (\mathrm{A}_{i,j}, \mathrm{grp}_i, \mathrm{x}_j) \mid \forall\ i,j\}. \end{cases}$$

Furthermore, *NO* obtains a set of revocation tokens, *grt*, with $grt[i,j] = \mathrm{A}_{i,j}$ and also keeps the mapping between group id i and grp_i for all user groups. Note that γ is the system secret only known to *NO*. For the purpose of nonrepudiation, *NO* signs on Steps 5 and 7 under a standard digital signature scheme, such as ECDSA [13]. In PEACE, we assume that ECDSA-160 is used. For the same purpose, GM_i and TTP also sign on these messages upon receipt and send the resulted signature back to *NO*.

Additionally, *NO* prepares each mesh router MR_k a public/private key pair, denoted by $(\mathrm{RPK}_k, \mathrm{RSK}_k)$. Each mesh router also obtains an accompanied public key

[1] x_j might have a larger bit length as compared to $\mathrm{A}_{i,j}$, which is a point on the chosen elliptic curve. In this case, we simply ignore the unnecessary bits of x_j.

certificate signed by NO to prove key authenticity. The signing key pair of NO is denoted by (NPK, NSK). The certificate contains the following fields at the minimum:

$$Cert_k = \{MR_k, \mathrm{RPK}_k, ExpT, Sig_{\mathrm{NSK}}\},$$

where $ExpT$ is the expiration time and $Sig_\bullet$ denotes an ECDSA-160 signature signed on a given message using a private key $\bullet$.

Before accessing the WMN, a network user has to authenticate himself to his belonging user groups.[2] From each such user group i, a network user uid_j is assigned a random group private key as follows:

1. GM_i sends $\mathrm{uid}_j([i,j], \mathrm{grp}_i, \mathrm{x}_j)$ as well as the related system parameters.
2. GM_i requests TTP to send $\mathrm{uid}_j([i,j], \mathrm{A}_{i,j} \oplus \mathrm{x}_j)$ by providing the index $[i,j]$.
3. uid_j assembles his group private key as $gsk[i,j] = (\mathrm{A}_{i,j}, \mathrm{grp}_i, \mathrm{x}_j)$.

Note that in our setting,

- GM_i only keeps the mapping of $(\mathrm{uid}_j, (\mathrm{grp}_i, \mathrm{x}_j))$ but has no knowledge of the corresponding $\mathrm{A}_{i,j}$.
- NO only knows the mapping of $(GM_i, gsk[i,j])$ but has no knowledge regarding to whom $gsk[i,j]$ is assigned.
- TTP has the mapping of $(\mathrm{uid}_j, (\mathrm{A}_{i,j} \oplus \mathrm{x}_j, \mathrm{grp}_i))$ as it sends uid_j this information through a secure channel between the two upon the request from GM_i. But TTP has no knowledge of the corresponding x_j or $\mathrm{A}_{i,j}$.

Here, we use uid_j the user's essential attribute information. For the purpose of nonrepudiation, uid_j signs on the messages it receives from GM_i and TTP under ECDSA-160, and sends back GM_i the corresponding signature.

4.2 User-Router Mutual Authentication and Key Agreement

To access the WMN, a network user follows the user-router mutual authentication and key agreement protocol as specified below, when a mesh router is within his direct communication range.[3]

1. The mesh router MR_k first picks a random nonce $r_R \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and a random generator g in $\mathbb{G}_1$ and then computes g^{r_R}. MR_k further signs on g, g^{r_R}, and current time stamp ts_1, using ECDSA-160. MR_k then broadcasts

   $$g, g^{r_R}, \mathrm{ts}_1, Sig_{\mathrm{RSK}_k}, Cert_k, CRL, URL \qquad \text{(M.1)}$$

   as part of *beacon messages* that are periodically broadcasted to declare service existence. Here, CRL and URL denote the mesh router certificate revocation list and the user revocation list, respectively. Specifically, URL contains a set of revocation tokens that corresponds to the revoked group private keys, which is a subset of grt. Both CRL and URL are signed by NO.
2. Upon receipt of (M.1), a network user uid_j proceeds as follows:

   a. Check the time stamp ts_1 to prevent replay attack. Examine $Cert_k$ to verify public key authenticity and the certificate expiration time; examine CRL and see if $Cert_k$ has been revoked by applying NPK. Further verify the authenticity of Sig_{RSK_k} by applying RPK_k.

   b. Upon positive check results, uid_j believes that MR_k is legitimate and does the following:

      i. Pick two random nonce $r, r_j \overset{R}{\leftarrow} \mathbb{Z}_p$, compute g^{r_j}, and prepare the current time stamp ts_2. Further obtain two generators $(\hat{u}, \hat{v})$ in $\mathbb{G}_2$ from H_0 as

      $$(\hat{u}, \hat{v}) \leftarrow H_0(gpk, g^{r_j}, g^{r_R}, \mathrm{ts}_2, r) \in \mathbb{G}_2^2, \qquad (1)$$

      and compute their images in $\mathbb{G}_1$: $u \leftarrow \psi(\hat{u})$ and $v \leftarrow \psi(\hat{v})$.

      ii. Compute $T_1 \leftarrow u^\alpha$ and $T_2 \leftarrow \mathrm{A}_{i,j} v^\alpha$ by selecting an exponent $\alpha \overset{R}{\leftarrow} \mathbb{Z}_p$. Set $\delta \leftarrow (\mathrm{grp}_i + \mathrm{x}_j)\alpha \in \mathbb{Z}_p$. Pick blinding values r_α, r_x, and $r_\delta \overset{R}{\leftarrow} \mathbb{Z}_p$.

      iii. Compute helper values R_1, R_2, and R_3 : $R_1 \leftarrow u^{r_\alpha}, R_2 \leftarrow e(T_2,\ g_2)^{r_x} \cdot e(v,\ w)^{-r_\alpha} \cdot e(v,\ g_2)^{-r_\delta}$, and $R_3 \leftarrow T_1^{r_x} \cdot u^{-r_\delta}$. Compute a challenge value $c \in \mathbb{Z}_p$ using H:

      $$c \leftarrow H(gpk, g^{r_j}, g^{r_R}, \mathrm{ts}_2, r, T_1, T_2, R_1, R_2, R_3) \in \mathbb{Z}_p.$$

      iv. Compute $s_\alpha = r_\alpha + c\alpha$, $s_x = r_x + c(\mathrm{grp}_i + \mathrm{x}_j)$ and $s_\delta = r_\delta + c\delta \in \mathbb{Z}_p$. Obtain the group signature on $\{g^{r_j}, g^{r_R}, \mathrm{ts}_2\}$ as

      $$\widehat{SIG}_{gsk[i,j]} \leftarrow (r, T_1, T_2, c, s_\alpha, s_x, s_\delta).$$

      v. Compute the shared symmetric key with MR_k:

      $$K_{k,j} = (g^{r_R})^{r_j}.$$

   c. Unicast back to MR_k

      $$g^{r_j}, g^{r_R}, \mathrm{ts}_2, \widehat{SIG}_{gsk[i,j]}. \qquad \text{(M.2)}$$
3. Upon receipt of (M.2), MR_k carries out the following to authenticate uid_j:

   a. Check g^{r_R} and ts_2 to make sure the freshness of (M.2).

   b. Check that $\widehat{SIG}_{gsk[i,j]}$ is a valid signature by applying the group public key gpk as follows:

      i. Compute $\hat{u}$ and $\hat{v}$ using (1), and their images u and v in $\mathbb{G}_1$: $u \leftarrow \psi(\hat{u})$ and $v \leftarrow \psi(\hat{v})$.

      ii. Retrieve R_1, R_2, and R_3 as:

[2] Such an authentication is based on the preestablished trust relationship between the user and the user group and may be done through in-person contact.

[3] If direct communication is not possible due to lack of mobility, a user can increase transmit power to reach the mesh router during this phase. After this phase, the user should reduce transmit power back to the normal level to help increase spatial concurrency and frequency reuse [4].

$$\tilde{R}_1 \leftarrow u^{s_\alpha}/T_1^c,$$
$$\tilde{R}_2 \leftarrow e(T_2, g_2)^{s_x} \cdot e(v, w)^{-s_\alpha} \cdot e(v, g_2)^{-s_\delta} \cdot (e(T_2, w)/e(g_1, g_2))^c,$$

and $\tilde{R}_3 \leftarrow T_1^{s_x} \cdot u^{-s_\delta}$.

iii. Check that the challenge c is correct:

$$c \stackrel{?}{=} H(gpk, g^{r_j}, g^{r_R}, \mathrm{ts}_2, r, T_1, T_2, \tilde{R}_1, \tilde{R}_2, \tilde{R}_3). \tag{2}$$

c. For each revocation token $\mathrm{A} \in URL$, check whether A is encoded in (T_1, T_2) by checking if

$$e(T_2/\mathrm{A}, \hat{u}) \stackrel{?}{=} e(T_1, \hat{v}). \tag{3}$$

If no revocation token of URL is encoded in (T_1, T_2), then the signer of $\widehat{SIG}_{gsk[i,j]}$ has not been revoked.

If all the above checks succeed, MR_k is now assured that the current user is a legitimate network user, although MR_k does not know which particular user this is. Note that uid_j is never disclosed or transmitted during protocol execution.

d. MR_k further computes the shared symmetric key as $K_{k,j} = (g^{r_j})^{r_R}$ and sends back uid_j:

$$g^{r_j}, g^{r_R}, E_{K_{k,j}}(MR_k, g^{r_j}, g^{r_R}), \tag{M.3}$$

where $E_\bullet()$ denotes symmetric encryption of the given message within the brackets using key $\bullet$.

The above protocol enables explicit mutual authentication between a mesh router and a legitimate network user; it also enables unilateral anonymous authentication for the network user. Upon successful completion of the protocol, the mesh router and the user also establish a shared symmetric key used for the subsequent communication session. And this session is uniquely identified through (g^{r_R}, g^{r_j}).

Remarks.

1. Equation (2) holds because

   a. $\tilde{R}_1 = u^{s_\alpha}/T_1^c = u^{r_\alpha + c\alpha}/(u^\alpha)^c = u^\alpha = R_1$.

   b.

$$\begin{aligned}
\tilde{R}_2 &= e(T_2, g_2)^{s_x} \cdot e(v, w)^{-s_\alpha} \cdot e(v, g_2)^{-s_\delta} \cdot \left(\frac{e(T_2, w)}{e(g_1, g_2)}\right)^c \\
&= (e(T_2, g_2)^{r_x} \cdot e(v, w)^{-r_\alpha} \cdot e(v, g_2)^{-r_\delta}) \\
&\cdot (e(T_2, g_2)^{\mathrm{grp}_i + \mathrm{x}_j} \cdot e(v, w)^{-\alpha} \\
&\cdot e(v, g_2)^{-(\mathrm{grp}_i + \mathrm{x}_j)\alpha} \cdot \frac{e(T_2, w)}{e(g_1, g_2)})^c \\
&= R_2 \cdot \left(\frac{e(T_2 v^{-\alpha}, w g_2^{\mathrm{grp}_i + \mathrm{x}_j})}{e(g_1, g_2)}\right)^c \\
&= R_2 \cdot \left(\frac{e(\mathrm{A}_{i,j}, w g_2^{\mathrm{grp}_i + \mathrm{x}_j})}{e(g_1, g_2)}\right)^c = R_2 \cdot \left(\frac{e(g_1, g_2)}{e(g_1, g_2)}\right)^c = R_2.
\end{aligned}$$

   c.

$$\begin{aligned}
\tilde{R}_3 &= T_1^{s_x} u^{-s_\delta} \\
&= (u^\alpha)^{r_x + c(\mathrm{grp}_i + \mathrm{x}_j)} \cdot u^{-r_\delta - c\alpha(\mathrm{grp}_i + \mathrm{x}_j)} \\
&= (u^\alpha)^{r_x} \cdot u^{-r_\delta} = T_1^{r_x} \cdot u^{-r_\delta} = R_3.
\end{aligned}$$

2. Equation (3) holds when there is an element A of URL encoded in (T_1, T_2) because of the following.

   We know that $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ is an isomorphism such that $\psi(g_2) = g_1$. According to the definition of isomorphism, we have $\psi(PQ) = \psi(P)\psi(Q)$ for any $P, Q \in \mathbb{G}_2$. Using this property and mathematical induction, it is easy to know the following fact: For any natural number $m \in N, \psi(g_2^m) = g_1^m$.

   Hence, if a group private key $(\mathrm{A}_{i,j}, \mathrm{grp}_i, x_j)$ with $\mathrm{A}_{i,j} \in URL$ signed the group signature σ. For simplicity, let $\hat{u} = g_2^a$ and $\hat{v} = g_2^b$ for some integers a and b.[4] On one hand,

$$\begin{aligned}
e(T_2/\mathrm{A}_{i,j}, \hat{u}) &= e(\mathrm{A}_{i,j} v^\alpha/\mathrm{A}_{i,j}, \hat{u}) = e(v^\alpha, \hat{u}) = e((\psi(\hat{v}))^\alpha, \hat{u}) \\
&= e((\psi(g_2^b))^\alpha, \hat{u}) = e((g_1^b)^\alpha, g_2^a) = e(g_1, g_2)^{ab\alpha}.
\end{aligned}$$

   On the other hand,

$$\begin{aligned}
e(T_1, \hat{v}) &= e(u^\alpha, \hat{v})) = e((\psi(\hat{u}))^\alpha, \hat{v}) = e((\psi(g_2^a))^\alpha, \hat{v}) \\
&= e((g_1^a)^\alpha, g_2^b) = e(g_1, g_2)^{ab\alpha}.
\end{aligned}$$

   Therefore, $e(T_2/\mathrm{A}_{i,j}, \hat{u}) = e(T_1, \hat{v})$.

4.3 User-User Mutual Authentication and Key Agreement

In PEACE, neighboring legitimate network users may help to relay each other's traffic. To this end, two network users within each other's direct communication range first authenticate each other and establish shared secret pairwise key as follows:

1. uid_j picks a random nonce $r_j \xleftarrow{R} \mathbb{Z}_p^*$ and computes g^{r_j}, where g is obtained from the *beacon messages* broadcasted by the current service mesh router. uid_j further signs on g, g^{r_j}, and current time stamp ts_1, using his group private key $gsk[i,j]$ following Steps 2b(i) to 2b(iv), as in Section 4.2. uid_j then locally broadcasts

$$g, g^{r_j}, \mathrm{ts}_1, \widehat{SIG}_{gsk[i,j]}. \tag{$\widetilde{\mathrm{M}}.1$}$$

2. Upon receipt of $(\widetilde{M}.1)$, uid_l checks the time stamp and verifies the authenticity of $\widehat{SIG}_{gsk[i,j]}$ by applying the group key gpk following Step 3b, as in Section 4.2. uid_l further checks if the signature is generated from a revoked group private key following Step 3c, as in Section 4.2. Note that URL can always be obtained from the *beacon messages*.

   If all checks succeed, uid_l is assured that the current user it communicates with is legitimate. uid_l proceeds to pick a random nonce $r_l \xleftarrow{R} \mathbb{Z}_p^*$ and computes g^{r_l}. uid_l further signs on g^{r_j}, g^{r_l}, and current time stamp ts_2, using an appropriate group private key $gsk[t,l]$ of his. uid_l also computes the shared pairwise session key as $K_{r_j,r_l} = (g^{r_j})^{r_l}$. uid_l then replies uid_j

$$g^{r_j}, g^{r_l}, \mathrm{ts}_2, \widehat{SIG}_{gsk[t,l]}. \tag{$\widetilde{\mathrm{M}}.2$}$$

3. Upon receipt of $(\widetilde{M}.2)$, uid_j first checks whether $\mathrm{ts}_2 - \mathrm{ts}_1$ is within the acceptable delay window. uid_j also

[4] Note that we do not know the exact value of a and b, but they indeed exist due to the fact that g_2 is a generator of $\mathbb{G}_2$.

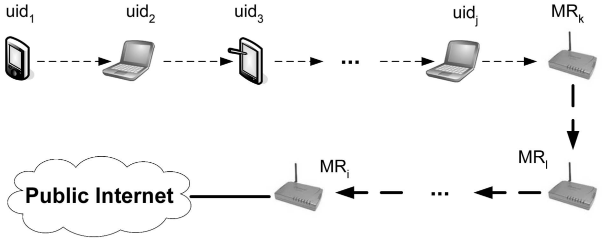

Fig. 4. A sample date session initiated by a network user.

examines $\widehat{SIG}_{gsk[i,j]}$ and URL as uid_l did above. If all checks succeed, uid_j is also assured that its communicating counterpart is legitimate. uid_j computes the shared pairwise session key as $K_{r_j,r_l} = (g^{r_l})^{r_j}$. uid_j finally replies uid_l

$$g^{r_j}, g^{r_l}, E_{K_{r_j,r_l}}(g^{r_j}, g^{r_l}, \mathrm{ts}_1, \mathrm{ts}_2). \qquad (\widetilde{\mathrm{M}}.3)$$

Upon receipt of ($\widetilde{M}$.3) and successful decryption of $E_{K_{r_j,r_l}}(g^{r_j}, g^{r_l}, \mathrm{ts}_1, \mathrm{ts}_2), \mathrm{uid}_l$ is assured that uid_j has successfully completed the authentication protocol and established the shared key for their subsequent communication session, which is uniquely identified through (g^{r_j}, g^{r_l}).

4.4 Data Traffic Authentication

Fig. 4 denotes a typical scenario, where the message sent by a network user has to travel multihops before reaching the nearest service mesh router. The following protocol describes how such a message sent by uid_1 is forwarded and efficiently authenticated in a hop-by-hop manner. Note that only symmetric cryptographic operations are required for data traffic authentication.

Assume that all the involving network users and mesh routers have mutually authenticated each other and established respective corresponding symmetric keys as summarized in Table 1, following the protocols described in the previous section. We also note that secure channels already exist among mesh routers themselves as the consequence of preconfiguration.

1. uid_1 first prepares the message M to be sent to a destination $dest$ and calculates $K_1 = h(K_{r_k^1,r_1^2}, 0)$ and $K_2 = h(K_{r_k^1,r_1^2}, 1)$ shared with and MR_k. uid_1 further encrypts $M, dest, g^{r_k^1}, g^{r_1^2}$ using K_1 and obtains $C = E_{K_1}(M, dest, g^{r_k^1}, g^{r_1^2})$. uid_1 also computes two message authentication codes (MAC) using K_2 and $K_{r_1^1,r_2^1}$, respectively:

$$MAC_1 = MAC_{K_2}(C),$$
$$MAC_2 = MAC_{K_{r_1^1,r_2^1}}(g^{r_1}, g^{r_2}, g^{r_k^1}, g^{r'_1}, \mathrm{MR}_k, C, MAC_1).$$

Finally, uid_1 sends uid_2:

$$g^{r_1^1}, g^{r_2^1}, g^{r_k^1}, g^{r_1^2}, \mathrm{MR}_k, C, MAC_1, MAC_2. \qquad (\overline{\mathrm{M}}.1)$$

TABLE 1
Link Summarization

Link	Session Identifier	Session Key
$(\mathsf{uid}_1, \mathsf{uid}_2)$	$(g^{r_1^1}, g^{r_2^1})$	$K_{r_1^1,r_2^1}$
$(\mathsf{uid}_1, \mathsf{MR}_k)$	$(g^{r_k^1}, g^{r_1^2})$	$K_{r_k^1,r_1^2}$
$(\mathsf{uid}_2, \mathsf{uid}_3)$	$(g^{r_2^2}, g^{r_3^1})$	$K_{r_2^2,r_3^1}$
$(\mathsf{uid}_2, \mathsf{MR}_k)$	$(g^{r_k^2}, g^{r_2^3})$	$K_{r_k^2,r_2^3}$
...	...	...
$(\mathsf{uid}_j, \mathsf{MR}_k)$	$(g^{r_k^j}, g^{r_j^3})$	$K_{r_k^j,r_j^3}$
...	...	...

2. Upon receipt of $(\overline{M}.1), \mathrm{uid}_2$ checks $(g^{r_1^1}, g^{r_2^1})$, fetches $K_{r_1^1,r_2^1}$ from the memory, and further verifies MAC_2 using $K_{r_1^1,r_2^1}$. If the verification succeeds, uid_2 proceeds to update MAC_2 using $K_{r_2^2,r_3^1}$ shared with uid_3, that is,

$$MAC_2 = MAC_{K_{r_2^2,r_3^1}}(g^{r_1^1}, g^{r_2^1}, g^{r_k^1}, g^{r_1^2}, \mathrm{MR}_k, C, MAC_1).$$

uid_2 next sends uid_3:

$$g^{r_2^2}, g^{r_3^1}, g^{r_k^1}, g^{r_1^2}, \mathrm{MR}_k, C, MAC_1, MAC_2. \qquad (\overline{\mathrm{M}}.2)$$

If the verification fails, the message is bogus and will be immediately dropped.

3. Upon receipt of $(\overline{M}.2), \mathrm{uid}_3$ processes it the same way as uid_2 does, and so are all the intermediate users.
4. When the message arrives at MR_k from $\mathrm{uid}_j, \mathrm{MR}_k$ further verifies it in three steps:

   a. verify MAC_2 using

   $$K_{r_k^j,r_j^3};$$

   b. verify MAC_1 using K_1 calculated from $K_{r_k^1,r_1^2}$; and
   c. check whether C can be properly decrypted using K_1.

   If all checks succeed, MR_k now forwards M to its destination $dest$ probably through more intermediate mesh routers.

In PEACE, $dest$ may be either a remote destination outside the WMN belonging to the public Internet, which can be indicated by its IP address, or another network user of the WMN. That is, two peer WMN users may also communicate with each other. Obviously, for the purpose of privacy protection, $dest$ cannot use IP address of the destination user or put user's ID in plain text in this latter case. This is because both approaches violate user privacy. The solution to this is to encrypt the destination user's ID in $dest$ so that no other network users or mesh routers, except the purposed receiver, are able to recover its content. The straightforward adoption of this solution in the above protocol will require all mesh routers to broadcast the message so that the real destination user is guaranteed to receive it. That is, the straightforward solution demands network-wide flooding for message delivery, which is highly inefficient. To deal with this problem, anonymous routing techniques [14], [15], [16], [17] are required, which usually make use of network-wide flooding only at the routing discovery phase but utilize unicast approach for the subsequent data transmission. Many anonymous routing approaches [14], [15], [16], [17] can be almost directly applied here, the detail of which, however, is beyond the scope of this paper and is a part of our ongoing work.

4.5 Privacy-Enhanced User Accountability

This design of PEACE protects user privacy in a sophisticated manner, while still maintaining user accountability.

4.5.1 User Anonymity against the Adversary, the User Groups, and TTP

In PEACE, a user only authenticates himself as a legitimate service subscriber without disclosing any of his identity information by utilizing the group signature technique. Neither the adversary nor the user group managers can tell which particular user generates a given signature. The adversary, even by compromising mesh routers and other network users, that is, knowing a number of group private keys in addition to the group public key, still cannot deduce any information regarding the particular group private key used for signature generation. This is due to the hardness of the underlying q-SDH problem, where q is a 1,020-bit prime number. Due to the same reason, neither a user group manager can distinguish whether or not one of his group members has signed a particular signature as he has no knowledge of the corresponding $\mathrm{A}_{i,j}$s nor can he compute them. The same conclusion also holds for TTP as TTP can compute neither x_j nor $\mathrm{A}_{i,j}$ given $\mathrm{A}_{i,j} \oplus \mathrm{x}_j$. Furthermore, every data session in PEACE is identified only through pairs of fresh random numbers, which again discloses nothing regarding user identity information. In addition, PEACE requires a network user to refresh session identifiers and the shared symmetric keys for each different session. This further eliminates the linkability between any two sessions initiated by the same network user. We note that even with the help of compromised mesh routers and other network users, the adversary still cannot judge whether two communication sessions are from the same user. This is because, fundamentally, none of them can tell whether two signatures are from the same user, given q-SDH problem and decision linear on $\mathbb{G}_1$ problem are hard.

4.5.2 User Privacy against NO and User Accountability

Since NO knows grt, it can always tell which $gsk[i,j]$ produces a given signature. However, NO has no knowledge regarding to whom $gsk[i,j]$ is assigned as PEACE allows a late binding between group private keys and network users. Furthermore, it is user group managers' sole responsibility to assign group private keys to each network user without any involvement of NO. Therefore, NO could only map $gsk[i,j]$ to the user group i based on grp_i. Because no other entities except NO and the key holder himself has the knowledge of the corresponding $\mathrm{A}_{i,j}$, and can therefore, generate the given signature, the key holder must be a member of the user group i. This audit result serves our both requirements. On one hand, the result only reveals partial nonessential attribute information of the user and still protects user privacy to an extent. On the other hand, the result is sufficient for user accountability purposes for NO.

When NO (on behalf of mesh routers) finds certain communication session disputable or suspicious, it conducts the following protocol to audit the responsible entity:

1. Given the link and the session identifier, find the corresponding authentication session message $(M.2) = g^{r_j}, g^{r_R}, \mathrm{ts}_2, \widehat{SIG}_{gsk[i,j]}$ from the network log file.
2. For each revocation token $\mathrm{A}_{i,j} \in grt$, check whether $e(T_2/\mathrm{A}_{i,j}, \hat{u}) \overset{?}{=} e(T_1, \hat{v})$. Output the first element $\mathrm{A}_{i,j} \in grt$ such that $e(T_2/\mathrm{A}_{i,j}, \hat{u}) = e(T_1, \hat{v})$.
3. For the found revocation token $\mathrm{A}_{i,j}$, output the corresponding mapping between $\mathrm{A}_{i,j}$ and grp_i. Since grp_i maps to a particular user group i, now a responsible entity is found from the perspective of NO.

From the user's perspective, only part of his nonessential attribute information is disclosed from the audit. But such nonessential attribute information will not reveal his essential attribute information. For example, the above audit may find that the responsible user is a member of Company XYZ but cannot reveal any other information regarding the user. Yet NO still has sufficient evidence to prove to Company XYZ that one of his members violates certain network access rule so that Company XYZ should take the corresponding responsibility specified in their service subscription agreement.

4.5.3 Revocable User Anonymity against Law Authority

When law authority decides to track the particular attacker that is responsible for a certain communication session, the following procedure is taken: NO reports to the law authority $(\mathrm{A}_{i,j}, \mathrm{grp}_i)$ by executing the above protocol against the session in audit. $(\mathrm{A}_{i,j}, \mathrm{grp}_i)$ is then further forwarded to GM_i. GM_i checks its local record, finds out the mapping between $(\mathrm{grp}_i$ and $\mathrm{x}_i)$, and hence, the corresponding user identity information uid_j, to whom $gsk[i,j]$ is assigned during the system setup. GM_i then replies uid_j to the law authority. At this point, law authority and only law authority gets to know about which particular user is responsible for the communication session in audit. We point out that this tracing procedure has the nonrepudiation property because 1) GM_i signed on all *gsk*s that are assigned from NO as the proof of receipt; 2) uid_j also signed on the messages when obtaining $gsk[i,j]$ from GM_i and TTP as the proof of receipt. PEACE also has nonframeability property because no one else knows $gsk[i,j]$ except NO and uid_j or is able to forge a signature on behalf of uid_j.

4.6 System Maintenance

PEACE supports both member addition and revocation in a dynamic manner. In PEACE, a group private key can be revoked, and no network users holding the same key are able to access the WMN afterward. Specifically, to revoke a group private key $gsk[i,j]$, NO simply adds the corresponding $\mathrm{A}_{i,j}$ to URL and sends the updated URL to mesh routers via secure channels. We note that the size of URL is linear to the accumulated number of group private keys being revoked, which can potentially grow fairly large as time elapses. To deal with this problem, we observe that revocations of group privacy keys are mainly due to two reasons: 1) expiration of service subscription and 2) violation of network access policy. According to the nature of the network access service, key revocations due to the former reason usually happen periodically and are prescheduled; and this is the major reason causing the size growth of URL. At the same time, key revocations due to the latter is often random and sporadic. Based on this observation, PEACE adopts a hybrid membership maintenance approach to keep the size of URL to the minimum.

Assume that the minimum subscription period of the network service is τ time unit, which can be set, for example, as one month. For the duration of each minimum subscription period, *NO* prepares a new group public key and a sufficient number of corresponding private keys. *NO* also arranges the usage of these group public keys in a sequential manner. That is, *NO* will attach the current *gpk* in use in every $(M.1)$, which is part of the beacon messages being periodically broadcasted by each mesh router. Then, a network user that subscribes the network service for $x\tau$ time units through user group i will obtain x group private keys from GM_i and TTP. Each of these group private keys will only be valid for τ time unit and expires automatically afterward. Next, within each minimum subscription period, if a group private key has to be revoked on the fly, *NO* simply follows the procedure described above to update URL. Now the size of URL will not grow very large as URL is always periodically emptied.

PEACE also supports the dynamic addition and revocation of mesh routers. To add a new mesh router, *NO* only needs to assign the router a new certificate and establish secure channels between the new router and the existing ones. To revoke a mesh router, *NO* simply revokes its certificate and updates CRL. In PEACE, CRL is constantly updated by *NO* in a prescheduled frequency known as a system parameter to every network user. That is, CRL is updated periodically such as once per hour, even if there is no mesh router being revoked. Moreover, an additional CRL update is always immediately issued once a mesh router is revoked. Every user in PEACE also keeps a most up-to-date version of CRL when interacting with different mesh routers and checks CRL against its current service mesh router whenever receiving a newer version. With this certification revocation approach, network users can easily judge whether or not a currently received CRL is up-to-date with a guaranteed delay upper bound: *min*{`inverse of the update frequency, (current time—the update time of the locally stored` CRL)}. We note that the size of CRL is usually much smaller than that of URL as we consider that the compromise of mesh routers is not very often. At the same time, the size of CRL can be easily controlled by setting a shorter valid period.

5 Scheme Analysis

5.1 System Security Analysis

As its fundamental security functionality, PEACE enforces network access control. Hence, we are most concerned with the following three different types of attacks, i.e., bogus data injection attacks, data phishing attacks, and DoS attacks.

Bogus data injection attacks: In such attacks, the adversary wants to inject bogus data to the WMN aimed at utilizing the network service for free. The sources of the bogus data could be outsiders, revoked users, or revoked mesh routers. However, such bogus data traffic will be all immediately filtered in PEACE. First, with respect to outsiders, they do not know any group private keys. Thus, they cannot produce correct message signatures, when attempting to initialize a communication session with *NO* and/or other network users. They also cannot bypass the authentication procedure and directly send out bogus data to others as they do not possess any shared symmetric session keys with them, and thus, cannot produce correct MACs. Then, regarding revoked users, there are two situations: 1) they do not have any group private key currently in use due to group public key update or 2) the corresponding group private keys owned by them are already revoked and are published in URL in beacon messages. Obviously, the revoked users cannot gain network access in neither cases. Finally, for revoked mesh routers, they are no longer valid members of the WMN. By checking CRL, no legitimate mesh routers will accept/relay data traffic from revoked mesh routers. Also, since the downlink from a mesh router to its service range is only one hop, network users never need to and will not relay data traffic for mesh routers in PEACE.

Data phishing attacks: In such attacks, the adversary may set up bogus mesh routers and try to phish user connections to such routers. In this way, the adversary could control network connection and analyze users' data traffic for their benefits. The phishing mesh routers can be either completely new mesh routers or revoked mesh routers both at the adversary's control. In the former case, the mesh router will not be able to authenticate itself to the network user. Therefore, no network user will establish any session with such a mesh router. Even if the mesh router could intercept the network traffic between a network user and a legitimate mesh router, it will not be able to decrypt the message and obtain any useful information. In the latter case, a newly revoked mesh router, however, will possibly be able to authenticate itself to a network user, if such a user does not possess the latest version of CRL. The network user may be cheated in this case but only for up to (`inverse of the update frequency—(current time—last periodical update time)`) time period. This is because the revoked mesh router will not be able to provide a legal CRL update at the next periodical CRL update time point.

DoS attacks: In such attacks, the adversary may flood a large number of illegal access request messages to mesh routers. The purpose is to exhaust their resources and render them less capable of serving legitimate users. In PEACE, for every access request message $(M.2)$, the corresponding mesh router has to verify a group signature and check the validity of the signer. Both operations involve expensive pairing operations, which, hence, can easily be exploited by the adversary. To deal with this issue, we adopt the same client-puzzle approach as adopted in [18]. The idea of this approach is as follows: When there is no evidence of attack, a mesh router processes $(M.2)$ normally. But, when under a suspected DoS attack, the mesh router will attach a cryptographic puzzle to every $(M.1)$ and require the solution to the puzzle be attached to each $(M.2)$. The mesh router commits resources to process $(M.2)$ only when the solution is correct. Typically, solving a client puzzle requires a brute-force search in the solution space, while solution verification is trivial [18].

Therefore, the adversary must have abundant resources to be able to promptly compute a large enough number of puzzle solutions in line with his sending rate of bogus access request $(M.2)$. In contrast, although puzzles slightly increase legitimate users' computational load when the mesh router is under attack, they are still able to obtain network accesses regardless the existence of the attack. We refer the readers to [18] for the complete design.

5.2 User Privacy and Accountability Analysis

PEACE protects user privacy in a sophisticated manner, while still maintaining user accountability. First, PEACE enables user anonymity against the adversary, the user group managers, and TTP. In PEACE, a network user only authenticates himself as a legitimate service subscriber without disclosing any of his identity information by utilizing the group signature technique. Neither the adversary nor the user group managers can tell which particular user generates a given signature. The adversary, even by compromising mesh routers and other network users, that is, knowing a number of group private keys in addition to the group public key, still cannot deduce any information regarding the particular group private key used for signature generation. This is due to the hardness of the underlying q-SDH problem, where q is a 1,020-bit prime number. Due to the same reason, a user group manager also cannot distinguish whether or not one of his group members has signed a particular signature as he has no knowledge of the corresponding $\mathrm{A}_{i,j}$s nor can he compute them. The same conclusion also holds for TTP as TTP can compute neither x_j nor $\mathrm{A}_{i,j}$ given $\mathrm{A}_{i,j} \oplus \mathrm{x}_j$. Furthermore, every data session in PEACE is identified only through pairs of fresh random numbers, which again discloses nothing regarding user identity information. In addition, PEACE requires a network user to refresh session identifiers and the shared symmetric keys for each different session. This further eliminates the linkage between any two sessions originated from the same network user. We note that even with the help of compromised mesh routers and other network users, the adversary still cannot judge whether two communication sessions are from the same user. This is because, fundamentally, none of them can tell whether two signatures are from the same user, given q-SDH problem and decision linear problem on $\mathbb{G}_1$ are hard.

Second, PEACE provides sufficient user privacy protection against NO while maintaining user accountability. Since NO knows grt, it can always tell which $gsk[i,j]$ produces a given signature. However, NO has no knowledge regarding to whom $gsk[i,j]$ is assigned as PEACE allows a late binding between group private keys and network users. Furthermore, it is the user group managers' sole responsibility to assign group private keys to each network user without any involvement of NO. Therefore, NO could only map $gsk[i,j]$ to the user group i based on grp_i. Because no other entities except NO and the key holder himself has the knowledge of the corresponding $\mathrm{A}_{i,j}$, and can therefore, generate the given signature, the key holder has to be a member of the user group i. This audit result serves our both requirements. On one hand, the result only reveals partial nonessential attribute information of the user and still protects user privacy to an extent. On the other hand, the result is sufficient for user accountability purposes for NO.

Finally, PEACE provides revocable user anonymity against the law authority. As discussed in Section 4.5, the law authority could track any particular user through the cooperation from both NO and the corresponding user group manager.

5.3 Performance Analysis

Communication overhead: In PEACE, Both authentication and key agreement protocols require only three-way communication between mesh routers and network users and between network users. This is the minimal communication rounds necessary to achieve mutual authentication, and therefore, PEACE incurs a reduced authentication delay. Furthermore, by design, PEACE poses minimum additional communication overhead on network users as they may carry their mobile clients such as PDAs and smart phones other than laptops to access the WMN. These mobile clients are much less powerful as compared to mesh routers with regard to their communication capability. In messages $(M.1), (\widetilde{M}.1)$, and $(\widetilde{M}.2)$, a network user only needs to transmit a group signature to fulfill the authentication function. As we base our group signature variation on the scheme proposed in [8], the signature comprises two elements of $\mathbb{G}_1$ and five elements of $\mathbb{Z}_p$. When using the curves described in [19], one can take p to be a 170-bit prime and use a group $\mathbb{G}_1$, where each element is 171 bits. Thus, the total group signature length is 1,192 bits or 149 bytes. With these parameters, security is approximately the same as a standard 1,024-bit RSA signature, which is 128 bytes [8]. That is, the length of the group signature is almost the same as that of a standard RSA-1024 signature.

Computational overhead: In PEACE, the most computationally expensive operations are the signature generation and verification. Signature generation requires two applications of the isomorphism ψ. Computing the isomorphism takes roughly the same time as an exponentiation in $\mathbb{G}_1$ (using fast computations of the trace map) [8]. Thus, signature generation requires about eight exponentiations (or multiexponentiations) and two bilinear map computations. Signature verification takes six exponentiations and $3 + 2|URL|$ computations of the bilinear map. By design, PEACE adopts an asymmetric-symmetric hybrid approach for session authentication to reduce computational cost. Network entities (both mesh routers and network users) execute expensive group signature operation to authenticate each other only when establishing a new session; all subsequent data exchanging of the same session is authenticated through highly efficient MAC-based approach.

More specifically, PEACE requires a network user executing exactly one signature generation and one signature verification when performing mutual authentication for establishing a new session. It can be seen that the actually computational cost of signature verification depends on the size of URL, while signature generation cost is fixed. PEACE can proactively control the size of URL. Moreover, a far more efficient revocation check algorithm, whose running time is independent of $|URL|$ can be adopted as described in [8] with a little bit sacrifice on user privacy. This technique could further bring the total cost of signature verification to six exponentiations and five bilinear map computations. On the other hand, PEACE requires a mesh router to perform mutual authentication with every network user within its coverage for each different session and sign on every beacon message being periodically broadcasted.

Storage overhead: In PEACE, network users may carry resource-constrained pervasive devices such as PDAs and smart phones to access the WMN. Therefore, storage overhead for each network user should be affordable to modern pervasive devices. As is shown in our scheme description, each network user in PEACE needs to store two pieces of information: his group private key and the related system parameters. The group private key for each user just contains 1 group element of $\mathbb{G}_1$ and 2 elements of $\mathbb{Z}_p^*$. If we choose p to be a 170-bit prime and use a group $\mathbb{G}_1$ with each group element of 171 bits, the group private key for each user just consumes 511-bit memory, which is negligible for modern pervasive devices. The most memory-consuming parts are the system parameters, which may include codes to describe the bilinear groups ($\mathbb{G}_1$ and $\mathbb{G}_2$), the bilinear pairing function (e), the isomorphism ψ, the hash functions (H_0 and H_1), and the signing function ECDSA-160. Fortunately, the required code size for each part could be in the magnitude of kilobytes as is studied in previous work such as [20]. Therefore, it should be affordable to most of the modern pervasive devices.

6 Related Work

Security research in WMNs is still in its early stage, especially with respect to user privacy protection. Ben Salem and Hubaux [21] discussed specifics of WMNs and identified fundamental network operations that need to be secured. Siddiqui and Hong [22] surveyed the threats and vulnerabilities faced by WMNs and also identified a number of security goals. Cheikhrouhou and Chaouchi [23] discussed a security architecture for WMNs based on IEEE 802.1X. [5] and Zhang and Fang [4] discussed how to support secure user roaming in a number of WMNs belonging to different domains. Wu and Li [24] presented an anonymous routing scheme for static WMNs. Wan et al. [25] proposed two privacy-preserving routing schemes to provide anonymity, unlinkability, and security for WMNs. The authors of [26], [27] presented an authentication scheme for WMNs, which is resilient against mesh router compromise. Other general privacy-aware authentication techniques are described in [28], [29], [30].

7 Conclusion

In this paper, we proposed PEACE, which, to the best of our knowledge, is the first attempt to establish an accountable security framework with a sophisticated user privacy protection model tailored for metropolitan scale WMNs. We developed a variation of the short group signature scheme [8]. We then built PEACE on this new signature variation by further integrating it into the authentication and key agreement protocol design. On one hand, PEACE enforces strict user access control to cope with both free riders and malicious users. On the other hand, PEACE offers sophisticated user privacy protection against both adversaries and various other network entities. Our analysis showed that PEACE is resilient to a number of security and privacy related attacks. Additional techniques were also discussed to further enhance the scheme efficiency.

Acknowledgments

This work was supported in part by the US National Science Foundation under grants CNS-0831963, CNS-0626601, CNS-0716306, CNS-0831628, and CNS-0716302. A preliminary version of this paper appeared in [1].

References

[1] K. Ren and W. Lou, “A Sophisticated Privacy-Enhanced Yet Accountable Security Framework for Wireless Mesh Networks,” *Proc. 28th Int'l Conf. Distributed Computing Systems (ICDCS '08),* June 2008.

[2] I.F. Akyildiz, X. Wang, and W. Wang, “Wireless Mesh Networks: A Survey,” *Computer Networks,* vol. 47, no. 4, pp. 445-487, Mar. 2005.

[3] “Self Organizing Neighborhood Wireless Mesh Networks,” http://www.research.microsoft.com/mesh/, 2009.

[4] Y. Zhang and Y. Fang, “A Secure Authentication and Billing Architecture for Wireless Mesh Networks,” *ACM Wireless Networks,* to be published.

[5] Y. Zhang and Y. Fang, “ARSA: An Attack-Resilient Security Architecture for Multi-Hop Wireless Mesh Networks,” *IEEE J. Selected Areas in Comm.,* vol. 24, no. 10, pp. 1916-1928, Oct. 2006.

[6] “The Wimax Forum,”http://www.wimaxforum.org. 2009.

[7] “Boston Suburb Secures Metro-Scale Wireless Mesh Network with Bluesocket,” http://www.tmcnet.com/usubmit/2006/09/27/1936581.htm, Sept. 2006.

[8] D. Boneh and H. Shacham, “Group Signatures with Verifier-Local Revocation,” *Proc. ACM Conf. Computer and Comm. Security (CCS),* pp. 168-177, 2004.

[9] D. Chaum and E. van Heyst, “Group Signatures,” *Proc. Conf. Eurocrypt,* pp. 257-265, 1991.

[10] R. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” *Comm. ACM,* vol. 21, no. 2, pp. 120-126, 1978.

[11] M. Jakobsson, J. Hubaux, and L. Buttyan, “A Charging and Rewarding Scheme for Packet Forwarding in Multi-Hop Cellular Networks,” *Proc. Seventh Int'l Conf. Financial Cryptography (FC),* 2003.

[12] N. Salem, L. Buttyan, J. Hubaux, and M. Jakobsson, “A Micro-Payment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks,” *Proc. ACM MobiHoc,* 2003.

[13] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography.* Springer-Verlag, 2004.

[14] Y. Zhang, W. Liu, and W. Lou, “Anonymous Communications in Mobile Ad Hoc Networks,” *Proc. IEEE INFOCOM,* Mar. 2005.

[15] Y. Zhang, W. Liu, W. Lou, and Y. Fang, “MASK: Anonymous On-Demand Routing in Mobile Ad Hoc Networks,” *IEEE Trans. Wireless Comm.,* vol. 5, no. 9, pp. 2376-2385, Sept. 2006.

[16] J. Kong, X. Hong, and M. Gerla, “An Identity-Free and On-Demand Routing Scheme against Anonymity Threats in Mobile Ad Hoc Networks,” *IEEE Trans. Mobile Computing,* vol. 6, no. 8, pp. 888-902, Aug. 2007.

[17] B. Zhu, Z. Wan, M.S. Kankanhalli, F. Bao, and R.H. Deng, “Anonymous Secure Routing in Mobile Ad-Hoc Networks,” *Proc. 29th Ann. IEEE Int'l Conf. Local Computer Networks (LCN '04),* pp. 102-108, Nov. 2004.

[18] A. Juels and J. Brainard, “Client Puzzles: A Cryptographic Countermeasure against Connection Depletion Attacks,” *Proc. Sixth Network and Distributed System Security Symp. (NDSS),* 1999.

[19] D. Boneh, H. Shacham, and B. Lynn, “Short Signatures from the Weil Pairing,” *J. Cryptology,* vol. 17, no. 4, pp. 297-319, 2004.

[20] TinyECC Library, http://discovery.csc.ncsu.edu/software/TinyECC/index.html, 2009.

[21] N. Ben Salem and J.-P. Hubaux, “Securing Wireless Mesh Networks,” *IEEE Wireless Comm.,* vol. 13, no. 2, pp. 50-55, Apr. 2006.

[22] M. Siddiqui and C. Hong, "Security Issues in Wireless Mesh Networks," *Proc. IEEE Int'l Conf. Multimedia and Ubiquitous Eng.,* 2007.

[23] A. Cheikhrouhou and H. Chaouchi, "Security Architecture in a Multi-Hop Mesh Network," *Proc. Fifth Conf. Security Architecture Research,* 2006.

[24] X. Wu and N. Li, "Achieving Privacy in Mesh Networks," *Proc. ACM Workshop Security of Ad Hoc and Sensor Networks (SASN),* 2006.

[25] Z. Wan, K. Ren, B. Zhu, B. Preneel, and M. Gu, "Anonymous User Communication for Privacy Protection in Wireless Metropolitan Mesh Networks," *Proc. ACM Symp. Information, Computer and Comm. Security (AsiaCCS),* 2009.

[26] X. Lin, R. Lu, P.-H. Ho, X. Shen, and Z. Cao, "Tua: A Novel Compromise-Resilient Authentication Architecture for Wireless Mesh Networks," *IEEE Wireless Comm.,* vol. 7, no. 4, pp. 1389-1399, Apr. 2008.

[27] X. Lin, X. Ling, H. Zhu, P.-H. Ho, and X. Shen, "A Novel Localized Authentication Scheme in ieee 802.11 Based Wireless Mesh Networks," *Int'l J. Security and Networks,* vol. 3, no. 2, pp. 122-132, 2008.

[28] K. Ren, W. Lou, K. Kim, and R. Deng, "A Novel Privacy Preserving Authentication and Access Control Scheme for Pervasive Computing Environment" *IEEE Trans. Vehicular Technology,* vol. 55, no. 4, pp. 1373-1384, July 2006.

[29] K. Ren and W. Lou, "Privacy-Enhanced, Attack-Resilient Access Control in Pervasive Computing Environments with Optional Context Authentication Capability," *ACM Mobile Networks and Applications (MONET)* (special issue on wireless broadband access), vol. 12, pp. 79-92, 2007.

[30] Y. Zhang and K. Ren, "On Address Privacy in Mobile Ad Hoc Networks," *ACM/Springer Mobile Networks and Applications (MONET),* vol. 14, no. 2, pp. 188-197, Apr. 2009.

Kui Ren received the BEng and MEng degrees from Zhejiang University in 1998 and 2001, respectively, and the PhD degree in electrical and computer engineering from Worcester Polytechnic Institute in 2007. He is an assistant professor in the Electrical and Computer Engineering Department at Illinois Institute of Technology. In the past, he has worked as a research assistant at Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences; Institute for Infocomm Research, Singapore; and Information and Communications University, South Korea. His research interests include network security and privacy and applied cryptography with current focus on security and privacy in cloud computing, lower layer attack and defense mechanisms for wireless networks, and sensor network security. His research is sponsored by the US National Science Foundation (NSF). He is a member of the IEEE, the IEEE Computer Society, and the ACM.

Shucheng Yu received the BE and ME degrees in computer science and engineering, respectively, from Nanjing University of Posts and Telecommunications, China, in 1999, and Tsinghua University, China, in 2004. He is currently working toward the PhD degree in the Electrical and Computer Engineering Department at Worcester Polytechnic Institute. He worked as a senior software engineer at Beijing R&D Center of Cadence Design Systems from July 2006 to December 2006 and Chinese National Source Coding Center from July 2004 to June 2006. From July 1999 to August 2001, he worked as a software engineer at Bright Oceans Corporation, Beijing, China. His research interests include network security and applied cryptography. His current research area focuses on privacy-preserving access control. He is a student member of the IEEE.

Wenjing Lou received the BE and ME degrees in computer science and engineering from Xi'an Jiaotong University in China, the MASc degree in computer communications from the Nanyang Technological University in Singapore, and the PhD degree in electrical and computer engineering from the University of Florida. From December 1997 to July 1999, she worked as a research engineer at Network Technology Research Center, Nanyang Technological University. In 2003, she joined the Electrical and Computer Engineering Department at Worcester Polytechnic Institute as an assistant professor, where she is now an associate professor. Her current research interests are in the areas of ad hoc, sensor, and mesh networks, with emphases on network security and routing issues. She has been an editor of the *IEEE Transactions on Wireless Communications* since 2007. She was named Joseph Samuel Satin Distinguished fellow in 2006 by WPI. She is a recipient of the NSF Faculty Early Career Development (CAREER) award in 2008. She is a senior member of the IEEE.

Yanchao Zhang received the BE degree in computer communications from Nanjing University of Posts and Telecommunications, China, in 1999, the ME degree in computer applications from Beijing University of Posts and Telecommunications, China, in 2002, and the PhD degree in electrical and computer engineering from the University of Florida in July 2006. He subsequently joined as an assistant professor in the ECE Department at NJIT. His primary research interests are network and distributed system security, wireless networking, and mobile computing. He is currently an associate editor of the *IEEE Transactions on Vehicular Technology* and a feature editor of the *IEEE Wireless Communications*. He is also the TPC cochair for Communication and Information System Security Symposium, IEEE GLOBECOM 2010. He is a member of the IEEE.