Anonymous User Communication for Privacy Protection in Wireless Metropolitan Mesh Networks

Zhiguo Wan, Kui Ren, Bo Zhu, Bart Preneel, and Ming Gu

***Abstract*—As a combination of ad hoc networks and wireless local area network (WLAN), the wireless mesh network (WMN) provides a low-cost convenient solution to the last-mile network-connectivity problem. As such, existing route protocols designed to provide security and privacy protection for ad hoc networks are no longer applicable in WMNs. On the other hand, little research has focused on privacy-preserving routing for WMNs. In this paper, we propose two solutions for security and privacy protection in WMNs. The first scheme relies on group signatures, together with user credentials, to deliver security and privacy protection. By enforcing access control using user credentials, the user's identity has to be disclosed to mesh routers. To avoid this, our second scheme employs pairwise secrets between any two users to achieve stronger privacy protection. In the second scheme, the user is kept anonymous to mesh routers. Finally, we analyze these two schemes in terms of security, privacy, and performance.**

***Index Terms*—Security, wireless networks.**

I. 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 [1], [11]. A typical metropolitan WMN, as shown in Fig. 1, consists of a group of mesh routers that form a wireless backbone and a large number of mesh clients (i.e., network users[1]) directly or indirectly connected to these mesh routers. The wireless backbone network formed by the mesh routers provides high-bandwidth communication channels to mesh clients connected to it. On the other hand, the mesh clients themselves form multihop wireless ad hoc networks to furthermore extend the wireless connectivity. WMNs represent a unique marriage of the ubiquitous coverage of wide-area cellular networks with the ease of local-area Wi-Fi networks [23]. The advantages of WMNs also include low deployment costs, self-configuration and self-maintenance, good scalability, high robustness, etc. [1].

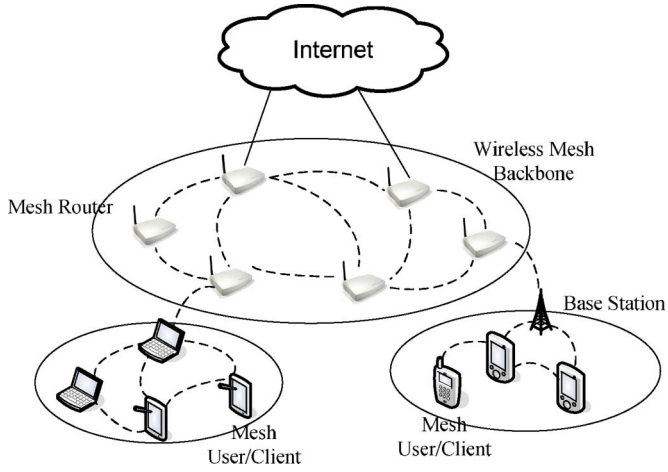


Fig. 1. Architecture of a typical mesh network.

Security and privacy issues are of utmost 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, WMNs are subject to various attacks. Anyone with an appropriate transceiver can eavesdrop, inject, or impersonate as others in a WMN. Rogue mesh routers can easily be set up to phish user connections and traffic. All these attacks pose a great threat on user privacy protection. In a metroscale community mesh network, the residents access the WMN from everywhere within the community, such as offices, homes, restaurants, hospitals, hotels, shopping malls, and even vehicles. Through the WMN, they access the public Internet in different roles and contexts for services like e-mails, e-banking, e-commerce, and web surfing, and also intensively interact with their local peers for file sharing, teleconferencing, online gaming, instant chatting, etc. Obviously, all these communications contain various kinds of sensitive user information like personal identities, activities, location information, movement patterns, financial information, transaction profiles, social/business connections, and so on. Once disclosed to the attackers, this information could compromise any user's privacy and, when further correlated together, can lead to even more devastating consequences. Hence, securing user privacy is of paramount practical importance in WMNs.

Manuscript received December 22, 2008; revised June 10, 2009. First published July 31, 2009; current version published February 19, 2010. This work was supported in part by the National Sciences and Engineering Research Council of Canada under Grant RGPIN/356059-2008, by the U.S. National Science Foundation under Grant CNS-0831963, by the Concerted Research Action Ambiorics 2005/11 of the Flemish Government, and by the Interuniversity Attraction Pole Program P6/26 Belgian Fundamental Research on Cryptology and Information Security of the Belgian State (Belgian Science Policy). The work of Z. Wan was supported in part by the Interdisciplinary Institute for BroadBand Technology, Flemish Government. The review of this paper was coordinated by Dr. L. Chen.

Z. Wan and M. Gu are with the Key Laboratory for Information System Security, Ministry of Education, Tsinghua National Laboratory for Information Science and Technology, School of Software, Tsinghua University, Beijing 100084, China.

K. Ren is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA.

B. Zhu is with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC H3G 2W1, Canada.

B. Preneel is with the Department of Electrical Engineering, Katholieke Universiteit Leuven, 3001 Leuven, Belgium.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2009.2028892

[1]In this paper, we do not distinguish mesh clients from network users.

Despite the necessity and importance, limited research has been conducted to address privacy-enhanced security mechanisms in WMNs. In this paper, we consider the problem of protecting local user communications in a metropolitan WMN. As previously mentioned, the WMN users are the city/community residents, and they may frequently communicate with each other for various purposes. Their communications, inevitably, contain large amount of user privacy information that should be protected from malicious attackers and other network entities. Addressing this issue in WMNs is both new and challenging for a number of reasons. First, a metropolitan WMN usually has a huge network size (on the order of thousands), which limits the applicability of traditional anonymous routing protocols designed for small-size mobile ad hoc networks (on the order of hundreds). This is because traditional anonymous routing protocols for ad hoc networks usually assume preestablished trust among all network nodes to find/establish secure routing paths [8]. However, in a metropolitan WMN, it is impossible to assume the preestablished trust relationship among all network users. Second, existing anonymous routing protocols for ad hoc networks do not enforce network access control (e.g., [8]), but in the WMN network, access control is essential for both security and billing purposes, as well as prohibiting network resource abuse. Last, but not least, most existing anonymous routing protocols rely on network-wide flooding for routing establishment [8]. While expensive, network-wide flooding is not a big concern in ad hoc networks because of their small size and light traffic load. However, flooding in the WMN will incur prohibitively high communication overhead and waste a large amount of precious network bandwidth resources, given its huge network size and heavy traffic load.

To address this challenge, in this paper, we propose two anonymous user communication protocol suites specifically tailored for WMNs: 1) the basic protocol suite and 2) the advanced protocol suite. Our basic scheme makes use of group signatures to anonymously establish session keys and enforce access control. In the first phase of the scheme, each mesh client anonymously constructs session keys with its neighboring nodes using group signatures. Then, these session keys are used for mesh clients to find routes to the nearest mesh router and have their identities registered. The registered identities are then used for route discovery within the mesh backbone. In this scheme, the user's identities are protected from eavesdroppers but known by mesh routers because of routing in the mesh backbone. In the advanced protocol, we make use of pairwise shared secrets along with group signatures to keep mesh clients anonymous from mesh routers. Hence, the advanced protocol suite achieves stronger privacy protection.

The rest of this paper is organized as follows: We discuss assumptions, threat model, and privacy requirements in Section II. Our basic anonymous communication protocol suite is described in Section III, followed by the advanced communication protocol suite. We analyze both schemes from the aspects of security, privacy, and performance. After that, a literature review on related work is given in Section VII. Finally, we draw the concluding remarks in Section VIII.

II. Assumptions, Threat Model, and Security/Privacy Requirements

A. Network Model

We consider a metropolitan-scale WMN under the control of a network operator (NO). The NO deploys a number of static 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. These mesh routers constantly exchange topology information with each other and, hence, maintain the global topology information of the whole WMN. Network users, on the other hand, subscribe to the NO for services and utilize their mobile clients to freely access the network and communicate with their peers 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 the NO in case of dispute/attack.

Both downlink (mesh router to user) and uplink (user to mesh router) can be multiple hop. In this case, network users cooperate with each other on relaying the packets. We further assume that all the network traffic has to go through a mesh router, except for the communication within the same subnet serviced by the same router. We assume that a reliable underlying transportation layer like Transmission Control Protocol has been implemented over mesh networks. Communication channels in WMNs are bidirectional, that is, if a node A can hear from node B at some time, then node B can also hear from A at the same time.

B. Threat Model

Due to the open medium and the spatially distributed nature, WMNs are vulnerable to both passive and active attacks. Hence, for a practical threat model, we consider a global passive attacker that is able to eavesdrop all network communications and an active attacker that can compromise and control a small number of users and mesh routers subject to his choices (but cannot control the NO) and thus can launch various active attacks based on this knowledge. The main purpose of the adversary is to compromise the following privacy features [12]:

1) Sender/receiver anonymity: Compromise the identity information of the source and destination nodes of any given communication session.
2) Relation unlinkability: Identify two communicating users even if they are not able to know their real identity.
3) Session unlinkability: Link different communication sessions of the same users.

In addition, the outside attacker may want to gain free access to WMN or impersonate as a valid entity in WMN. In addition, both inside and outside attackers may try to bring down the system by denial-of-service (DoS) attacks.

C. Security/Privacy Requirements

Regarding security requirements in WMN, we consider the following issues in our design: First, as WMN is used to

provide network services for its users, it is important to limit network access only to valid users and prevent unauthorized use. Without prior registration, users are not allowed to use resources of WMN. Second, WMN should be able to prevent invalid users from impersonating as valid entities, such as mesh routers and mesh clients. Finally, DoS attacks are serious threat for WMNs, and mechanisms should be designed to thwart such attacks aiming to deplete resources of WMNs.

As for privacy requirements, we follow the classification approach introduced in [15] to characterize different levels of user privacy. This approach defines user anonymity through two different dimensional parameters: 1) user identity information and 2) network entities, which are able to access this information. In this approach, a specific privacy requirement can always be expressed in terms of a 2-D matrix. In this privacy matrix, the columns represent different network entities, whereas the rows represent identity information. If a specific entity can identify the identity information from the messages exchanged in the WMN, then the corresponding entry in the matrix is marked $\surd$. Otherwise, the entry is marked $\times$.

In WMNs, network entities include network users, mesh routers, the NO, the eavesdropper (i.e., the global passive attacker), and the active attacker. Particularly, the NO controls the superset of all mesh routers, whereas the active attacker could take control of a superset of a small number of network users, mesh routers, and the eavesdropper. That is, the active attacker controls a small number of network users and mesh routers at its own choices in addition to the eavesdropper. Regarding a particular communication session between two network users, the identity information of interest are the following: 1) source node; 2) destination node; 3) source router, i.e., the current service mesh router of the source node; and 4) destination router, i.e., the current service mesh router of the destination node. We note that the identities of the source and destination routers themselves are not secret; however, they should never be able to be linked to a given communication session between two particular users. Different from ad hoc networks, in WMNs, the knowledge of the user's service mesh routers could directly lead to the disclosure of the user's location privacy.

Accordingly, the network entities can be grouped into ten different categories based on the accessibility of user identity information regarding a given communication session: 1) source node; 2) destination node; 3) all other nodes; 4) source router; 5) destination router; 6) routers on the communication path; 7) all other mesh routers; 8) the NO; 9) the eavesdropper; and 10) the active attacker.

We now define two different levels of user privacy-protection requirements using the privacy matrix.

1) C_1—User privacy against passive attackers and other network users (Table I): This level of protection hides the identity information of the source and destination nodes from the global passive eavesdropper and other (legitimate) network users. However, to the NO and the active attacker, the identity information of the two nodes is still available. Clearly, under this level of protection, a network user can always be traced by the NO and the active attacker.

TABLE I
PRIVACY REQUIREMENT CLASS C_1

Entity \ ID Info	Source Node	Source Router	Dest. Node	Dest. Router
Source Node	-	$\surd$	$\surd$	$\times$
Dest. Node	$\surd$	$\times$	-	$\surd$
Other Nodes	$\times$	$\surd$	$\times$	$\surd$
Source Router	$\surd$	-	$\surd$	$\surd$
Dest. Router	$\surd$	$\surd$	$\surd$	-
On-path Routers	$\times$	$\surd$	$\times$	$\surd$
Other Routers	$\times$	$\times$	$\times$	$\times$
Network Operator	$\surd$	$\surd$	$\surd$	$\surd$
Eavesdropper	$\times$	$\surd$	$\times$	$\surd$
Active Attacker	$\surd$	$\surd$	$\surd$	$\surd$

TABLE II
PRIVACY REQUIREMENT CLASS C_2

Entity \ ID Info	Source Node	Source Router	Dest. Node	Dest. Router
Source Node	-	$\surd$	$\surd$	$\times$
Dest. Node	$\surd$	$\times$	-	$\surd$
Other Nodes	$\times$	$\surd$	$\times$	$\surd$
Source Router	$\times$	-	$\times$	$\surd$
Dest. Router	$\times$	$\times$	$\times$	-
On-path Routers	$\times$	$\times$	$\times$	$\times$
Other Routers	$\times$	$\times$	$\times$	$\times$
Network Operator	$\times$	$\surd$	$\times$	$\surd$
Eavesdropper	$\times$	$\surd$	$\times$	$\surd$
Active Attacker	$\times$	$\surd$	$\times$	$\surd$

2) C_2—User privacy against both the NO and the active attacker (see Table II): Under this level of protection, the identity information of the source and destination nodes is always protected from all other network entities. C_2 also implies unlinkability, that is, different communication sessions between the same pair of users cannot be linked even by the active attacker. Furthermore, C_2 prohibits the active attacker from knowing the source and destination routers of a given communication session, as long as the source router itself is not compromised.

We further emphasize that this paper does not consider damage caused by global traffic analysis, which itself is of independent research interest. Under this type of attack, the attacker will possibly be able to identify an ongoing communication session between two network users by analyzing the network traffic pattern without knowing the user's identity information. This attack is effective in ad hoc networks with sparse data traffic patterns and usually deals with a dummy traffic-injection approach. However, WMNs usually present a dense traffic pattern by themselves and, thus, naturally cope with such attacks to a large extent. We leave a detailed treatment of this attack in our subsequent work.

III. ANONYMOUS USER COMMUNICATION: THE BASIC PROTOCOL SUITE

In this section, we present our basic protocol suite for anonymous user communication in WMNs, which achieves privacy requirement level C_1. This basic protocol suite consists of three parts: 1) local key establishment protocol; 2) node-to-router path finding and registration protocol; and 3) anonymous

TABLE III
NOTATION

q	A 160-bit prime number
$\mathbb{G}_1$	A group of order q
g	A generator of group G_1
$H(\circ)$	A secure one-way hash function $\{0,1\}^* \rightarrow \{0,1\}^m$
$Sig_R(\circ)$	A regular signature signed with R's private key
$SIG_S(\circ)$	A group signature signed with S's group private key
$E_k(\circ)$	Symmetric Encryption using key k
k_{S*}	Local broadcast key within S's one-hop neighborhood
k_{SX}	The pairwise session key shared between S and X
$\bar{k}_{SX}$	The pre-shared pairwise secret key between S and X
Nym_S	The route pseudonym noted by S
Nym_{SX}	The pseudonym used for verification between S and X

message delivery protocol. The notations are listed in Table III for easy reference.

In this paper, we utilize the group signature technique in our protocols to achieve anonymous user authentication and session key establishment. Group signature schemes are a relatively recent cryptographic concept introduced by Chaum and van Heyst [5], in which all members of a group share the same group public key but have different group private keys. A group signature scheme is a method to allow a group member 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 verify a signature but cannot decide who is the signer. However, in exceptional cases, such as a legal dispute, any group signature can be "opened" by a designated group manager to unambiguously reveal the identity of the signature's originator. Some group signature schemes support revocation, where group membership can be disabled. Various group-signature schemes based on different mathematical hard problems have been proposed in the literature. In this paper, we do not specify the group-signature scheme to be used as any scheme that provides membership revocation can be used.

A. System Setup

We assume that there is an offline trusted third party (TTP) responsible for generating/assigning user group private keys and the group public key. Each network user registers with TTP after subscribing to the network service from the NO and is assigned a unique group private key and the common group public key. Only TTP can identify the actual signer given a signature and has the ability to revoke the group members. TTP is trusted not to disclose the user group private key information to the NO, but TTP will cooperate with the NO to resolve disputes and attacks when necessary. Note that although we described a centralized approach here, distributed approaches that deal with semi-trusted TTP are always available, for example, in [10] and [14].

We also assume that each mesh router is also preconfigured with a public/private key pair under any regular signature scheme, such as elliptic curve digital signature algorithm or Rivest–Shamir–Adleman algorithm. The authenticity of such public keys is certified through certificates signed by the NO, and the public key of the NO is distributed to all network users when they subscribe to the service.

B. Anonymous Local Key Establishment Protocol

During this stage, every network user first needs to perform mutual authentication with his one-hop neighbors[2] and therefore establishes a set of session keys shared with each one of them, respectively. These keys are used for the subsequent route-finding protocol.

In this protocol, every user broadcasts a message within his neighborhood to initiate the local key establishment protocol. Each of his neighbors replies to the initiation message and derives session keys from the messages. For privacy protection, this protocol makes use of the group-signature technique to achieve anonymous authentication.

In proposed protocol, a mesh router R broadcasts

$$g, g^{r_R}, l, \mathtt{ts}, Sig_R, \mathtt{Cert}_R, \mathtt{CRL}, \mathtt{URL}$$

as part of the *beacon messages* that are periodically broadcast to declare service existence. Here, $r_R \in \mathbb{Z}_p^*$ denotes a random nonce, g denotes a random generator of $\mathbb{G}_1$, l is a system parameter specifying the packet size, and Sig_R is a regular signature over g, g^{r_R}, l, and current timestamp $\mathtt{ts}$, signed with R's private key. CRL and URL denote the mesh router certificate revocation list and the user revocation list, respectively. Both of them are signed by the NO. This beacon message reaches network users with one or multiple hops.

The proposed local key establishment protocol then goes as follows for a network user S:

1) Node S generates a random number $r_S \in \mathbb{Z}_q^*$, computes g^{r_S} and further obtains a group signature SIG_S of g^{r_S} using its group private key. Note that anyone can verify this signature using the group public key. S then broadcasts $\langle g^{r_S}, SIG_S \rangle$ within its neighborhood.
2) A neighbor X of S receives the message from S and verifies the signature in that message.[3] If the verification is successful, then X chooses a random number $r_X \in \mathbb{Z}_q^*$ and computes g^{r_X}. X also obtains a group signature SIG_X over g^{r_X} using its own group private key. X further computes the session key $k_{SX} = H(g^{r_S r_X})$ and replies to S with message $\langle g^{r_X}, SIG_X, E_{k_{SX}}(k_{X*}) \rangle$, where k_{X*} is X's local broadcast key.
3) Upon receiving the reply from X, S verifies the signature inside the message. If the signature is valid, then S proceeds to compute the session key between X and itself as $k_{SX} = H(g^{r_S r_X})$. S also generates a local broadcast key k_{S*} and sends $E_{k_{SX}}(k_{S*}|k_{X*})$ to its neighbor X to inform X about the established local broadcast key.
4) X receives the message from S and computes the same session key as $k_{SX} = H(g^{r_S r_X})$. It then decrypts the message to get the local broadcast key k_{S*}.

Fig. 2 illustrates how session keys can anonymously be established. As a result of this protocol, nodes X and S establish a pairwise session key k_{SX} without knowing each other's identity, but both are assured that the other party is a legitimate

[2]The neighbors possibly include the mesh router if it is within the direct communication range of the network user.

[3]Note that X checks the URL it obtained from the beacon message as well, and this also applies to S when checking X's signature below.

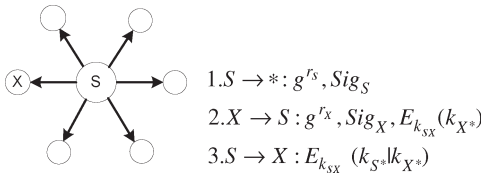


Fig. 2. Anonymous key agreement. S broadcasts the first message to its direct neighbors. Each of S's neighbors follows the same procedure as X to learn S's local broadcast key. The session key is $k_{SX} = H(g^{r_S r_X})$.

network user. This key is used to protect unicast messages between S and X, e.g., route reply packets. On top of this, node S also establishes a local broadcast key k_{S*} shared with all of its neighbors. It serves to secure broadcast messages in the subsequent routing-discovery process. Any broadcast message should be encrypted with this local broadcast key, e.g., the route request packets (RREQs).

Note that the number of session keys constructed in this phase is limited by the number of neighbors of a node. This is important when analyzing the computation complexity in the next route-discovery phase. The foregoing protocol is periodically reexecuted or when the network user detects new neighbors, and a network user only maintains the keys for the current active neighbors.

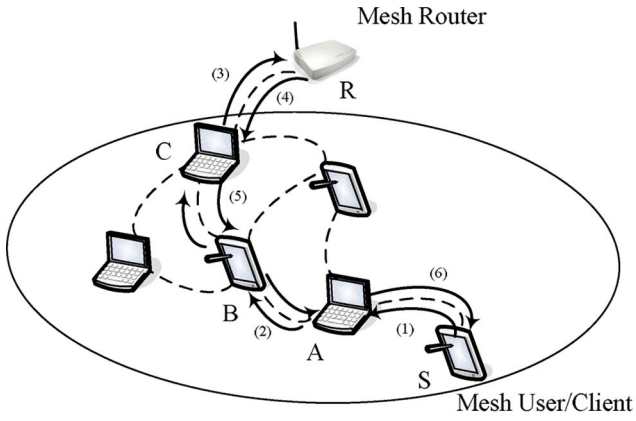
C. Node-to-Router Path Finding and Registration Protocol

This protocol makes use of the session keys obtained from the foregoing procedure to establish an anonymous route between a mesh client and its nearest mesh router. It also registers the client to the mesh router.

This protocol consists of three steps. In the first step, the source node broadcasts a route request throughout the subnet to which it belongs, and the request would reach the nearest mesh router under the protection of session keys. Then, the mesh router registers the node and puts it into its user list in the second step. This information is exchanged among mesh routers so that every mesh router knows how to reach a specific node. Next, the mesh router sends a reply to the source node, and the route is constructed when the reply successfully reaches the source node.

Suppose a source node S needs to find a route to the nearest mesh router R. Without loss of generality, we assume that there are three intermediate nodes A, B, and C between S and R, as illustrated in Fig. 3. The general case can easily be inferred from this example. The routing-discovery process executes as follows.

Route Request: From the beacon message, S obtains the current g^{r_R} and the mesh router's ID R. S then chooses a random number r_S and calculates g^{r_S} and $k_{SR} = H(g^{r_S r_X})$. k_{SR} will be the node-router pairwise session key between S and R. S then encrypts (S, R, g^{r_R}, g^{r_S}) with k_{SR} to obtain $\mathtt{Reg}_S = E_{k_{SR}}(S, R, g^{r_R}, g^{r_S})$ for registration purposes. Next, S signs $\mathtt{Reg}_S$ together with R, g^{r_S}, and a random picked sequence number $seqno$ with his group private key for the purpose of authentication and message integrity protection. This yields SIG_S. S then chooses a nonce $Nonce_S$ and calculates S's route pseudonym as $\mathrm{Nym}_S = H(k_{S*}|Nonce_S)$. Finally, S puts together $\langle RREQ, \mathrm{Nym}_S, R, \mathtt{Reg}_S, g^{r_S}, seqno, SIG_S\rangle$,



(1) $Nonce_S, E_{k_{S*}}(\mathtt{RREQ}, \mathrm{Nym}_S, R, \mathtt{Reg}_S, g^{r_S}, seqno, SIG_S, \mathtt{Pad})$,
(2) $Nonce_A, E_{k_{A*}}(\mathtt{RREQ}, \mathrm{Nym}_A, R, \mathtt{Reg}_S, g^{r_S}, seqno, SIG_S, \mathtt{Pad})$,
(3) $Nonce_C, E_{k_{CR}}(\mathtt{RREQ}, \mathrm{Nym}_C, R, \mathtt{Reg}_S, g^{r_S}, seqno, SIG_S, \mathtt{Pad})$,
(4) $R, E_{k_{CR}}(\mathtt{RREP}, R, \mathrm{Nym}_C, seqno, \mathtt{Rep}_R, \mathtt{pad})$,
(5) $\overline{Nonce}_C, E_{k_{BC}}(\mathtt{RREP}, \mathrm{Nym}_{BC}, \mathrm{Nym}_B, seqno, \mathtt{Rep}_R, \mathtt{pad})$,
(6) $\overline{Nonce}_A, E_{k_{SA}}(\mathtt{RREP}, \mathrm{Nym}_{SA}, \mathrm{Nym}_S, seqno, \mathtt{Rep}_R, \mathtt{pad})$

Fig. 3. Routing example with three intermediate nodes.

performs random padding according to l, and encrypts the whole information with its local broadcast key k_{S*}. S then broadcasts the following undistinguishable RREQ in its neighborhood:

$$Nonce_S, E_{k_{S*}}(\mathtt{RREQ}, \mathrm{Nym}_S, R, \mathtt{Reg}_S, g^{r_S}, seqno, SIG_S, \mathtt{Pad}). \quad (1)$$

Upon receiving the route request message from S, A tries all the possible session keys shared with current neighbors to decrypt the message and calculate $H(k_{X*}|Nonce_S)$ (or $H(k_{XA}|Nonce_S)$) to see which key matches the decrypted Nym_S. The incurred computation cost is decided by the number of neighbors of A and is negligible compared with public key operations.

Then, A would find out that k_{S*} satisfies $\mathrm{Nym}_S = H(k_{S*}|Nonce_S)$. At this point, if A already has a route to reach the mesh router R in his routing table, then A will use the existing route. Otherwise, A continues to look for a route to the mesh router. Either way, A generates a nonce $Nonce_A$ and calculates A's pseudonym $\mathrm{Nym}_A = H(k_{A*}|Nonce_A)$. A also records Nym_S and $seqno$ in his routing table. In the end, A sends the following message to its neighbors:

$$Nonce_A, E_{k_{A*}}(\mathtt{RREQ}, \mathrm{Nym}_A, R, \mathtt{Reg}_S, g^{r_S}, seqno, SIG_S, \mathtt{Pad}). \quad (2)$$

A's neighboring nodes that do not know how to reach a mesh router proceed the same as A does. Until an intermediate node like C knows where the router is, C just unicasts the message to the router. Note that the unicast message is protected with the pairwise session key between R and C as

$$Nonce_C, E_{k_{CR}}(\mathtt{RREQ}, \mathrm{Nym}_C, R, \mathtt{Reg}_S, g^{r_S}, seqno, SIG_S, \mathtt{Pad}). \quad (3)$$

TABLE IV
ROUTING INFORMATION RECORDED IN ROUTING TABLES OF S, A, B, C, AND R

	Src/Dest	Sequence No.	Downlink Enc/Dec Key	Uplink Enc/Dec Key	Downlink Route Pseudonym	Uplink Route Pseudonym
S	S/R	Seqno	-	k_{SA}	-	Nym_S
A	-/R	Seqno	k_{SA}	k_{AB}	Nym_S	Nym_A
B	-/R	Seqno	k_{AB}	k_{BC}	Nym_A	Nym_B
C	-/R	Seqno	k_{BC}	k_{CR}	Nym_B	Nym_C
R	S/R	Seqno	k_{CR}	-	Nym_C	-

As the RREQ finally reaches the mesh router R, R finds out the correct key k_{CR} according to the equation $\mathrm{Nym}_C = H(k_{C*}|Nonce_C)$. R further records Nym_C and $seqno$ into his route table. R also verifies the validity of the signature by applying the group public key and checking URL. Note that R may receive more than one route request message that originates from the same source and has the same sequence number, but he just replies to the first arriving message and drops others.

Node Registration: Upon successful receipt and verification of a routing request message, the mesh router R proceeds to register the requesting node. R first computes $k_{SR} = H(g^{r_S r_X})$ and then decrypts $\mathtt{Reg}_S$ to obtain S. R thus knows that S is currently within his service coverage and puts S into its current user list but only for a predetermined period of time. S has to periodically reregister itself to R to maintain the active status and to refresh k_{SR} for forward secrecy; otherwise, S's registration automatically expires in R's user list.

Route Reply: Route reply messages (RREPs) are unicast backward toward the source node, although radio media makes every one-hop transmission a broadcast node. R first uses the session key $k_{SR} = H(g^{r_S r_X})$ to encrypt $\langle S, R, seqno, g^{r_S}, g^{r_R}\rangle$ and obtains $\mathtt{Rep}_R = E_{k_{SR}}(S, R, seqno, g^{r_S}, g^{r_R})$. Then, R performs the padding operation as appropriate and replies the following undistinguishable route reply packet to C:

$$R, E_{k_{CR}}(\mathtt{RREP}, R, \mathrm{Nym}_C, seqno, \mathtt{Rep}_R, \mathtt{pad}). \tag{4}$$

When C receives the foregoing message from R, he uses k_{CR} to decrypt the ciphertext and finds out which route this RREP is related to based on C's pseudonym Nym_C and $seqno$ Nym_C. Next, C computes $\mathrm{Nym}_{BC} = H(k_{BC}|\overline{Nonce}_C)$ with a new nonce $\overline{Nonce}_C$, and this pseudonym is used as the route pseudonym and sends the following message to B:

$$\overline{Nonce}_C, E_{k_{BC}}(\mathtt{RREP}, \mathrm{Nym}_{BC}, \mathrm{Nym}_B, seqno, \mathtt{Rep}_R, \mathtt{pad}). \tag{5}$$

Other intermediate nodes perform the same operations as C does. Finally, the following route reply is sent back to the source node S by A as in our example:

$$\overline{Nonce}_A, E_{k_{SA}}(\mathtt{RREP}, \mathrm{Nym}_{SA}, \mathrm{Nym}_S, seqno, \mathtt{Rep}_R, \mathtt{pad}). \tag{6}$$

S now decrypts the outer layer ciphertext using k_{SA} and $\mathtt{Rep}_R$ in the inner layer using k_{SR}. If S correctly decrypts both messages, then he is assured that he has successfully registered with and established a route to the mesh router R. This protocol is periodically reexecuted by each network user when the network user moves to a new service mesh router. It is important to note that the mesh router has route information about all the nodes within its subnet.

The final routing information recorded in the routing tables of S, A, B, C, and R is shown in Table IV. As shown in this table, only the mesh router R knows that the route starts from the source node S, and all the other nodes only know that the route terminates at the mesh router R. All nodes have the same sequence number for a specific route in their routing tables. Each node, except the source node and the mesh router, has two keys for packet decryption or encryption for uplink and downlink, respectively. When an intermediate node receives a packet from uplink (respectively downlink), it decrypts the packet and encrypts the new packet using the corresponding downlink (respectively, uplink) key. Similarly, two route pseudonyms in the routing table of each intermediate node denote the route for uplink and downlink, respectively.

D. Anonymous Message Delivery Protocol

This section explains in detail how a message is anonymously delivered from a source node S to a destination node D. Essentially, the delivery of a message consists of three steps: 1) uplink routing; 2) router–router routing; and 3) downlink delivery. First, it is sent to the source mesh router, which is the nearest mesh router to the source node. The message is sent along the route constructed from the source node to the source mesh router, which is protected with local session keys. Next, the source router finds out the correct destination router and routes the packet to the destination router. Every mesh router knows how to reach a specific node since each node has registered with the nearest mesh router. In the third step, the destination mesh router dispatches the message to the destination node, and this process is the same as the process of sending a route reply.

Without loss of generality, S is assumed to be located within the service coverage of the mesh router R^S, whereas D is served by another mesh router R^D. Both S and D are multiple hops away from their service mesh router. With the path-finding and registration protocol, both D and S establish routes to their service routers, respectively. Suppose the route from S to R^S passes through A, B, and C, whereas the route from R^D and D goes through X and Y. The whole delivery procedure consists of three steps.

1) Uplink Routing (Node→Router): Essentially, sending a message Msg from the source node to the source mesh router can be viewed as the reverse process of sending an RREP from the mesh router to the source node. The packet traveling

through an intermediate node A from S takes the following form:

$$Nonce_S, E_{k_{SA}}\left(\mathrm{Nym}_{SA}, \mathrm{Nym}_S, E_{k_{SR^S}}(D, Msg)\right).$$

Before Msg is sent out, preprocessing is performed as appropriate, which either breaks Msg into smaller pieces or pads it to size l. The content of Msg can further be protected from the mesh router by encrypting it using the receiver node's public key or the secret key shared between the sender and the receiver, when necessary. The packets are routed from S to R^S following the already established route, i.e., they will go through A, B, and C to reach R^S.

2) Router-Router Routing (Router→Router): Since network users always register themselves with mesh routers, hence, by constantly exchanging such information, each mesh router will be able to establish the global topology information of the WMN.

At each mesh router, a dynamic list of registered users is maintained, and it is changing when a user moves out of the service coverage or a new user enters the service coverage. Periodically or triggered by user list changes, each router broadcasts changes on its registered user list so that every router can get the latest information on network topology. As it is assumed that the mesh routers are static and well connected, such information exchange can be accomplished with ease. Hence, all mesh routers know which destination mesh router to contact to reach a specific user.

Consequently, a mesh router will be able to reach any destination node as long as it is alive by first contacting its current service mesh router. With the topology information at hand, a mesh router could simply employ a source-routing technique to reach the destination mesh router R^D. Packets being transmitted between two neighboring mesh routers are always under the protection of preconfigured pairwise secure channels and kept the same length l.

3) Downlink Delivery (Router→Node): After receiving the message from R^S, router R^D retrieves plaintext from it and produces the following outgoing message destined for D. Since every user should register with the nearest service mesh router, R^D knows how to route the message to D—just the reverse of the uplink routing procedure. Therefore, the router R^D sends the following message toward destination D:

$$Nonce_R, E_{k_{RX}}\left(\mathrm{Nym}_{RX}, \mathrm{Nym}_R, E_{k_{DR^D}}(D, Msg)\right).$$

Node X would receive this message and forward it to Y, who will finally forward to the destination D. At this point, we complete our description of the basic protocol suite for anonymous user communication.

IV. Anonymous User Communication: the Advanced Protocol Suite

In this section, we present our advanced protocol suite, which is built upon the proposed basic protocol suite and shares the same assumptions. Additionally, each user is preinstalled with a public/private key pair, and any user knows the other user's public keys so that a secret key $\overline{k}_{SD}$ can be computed by a sender S and a receiver D from their private keys independently (i.e., by Diffie–Hellman exchange). To achieve enhanced user privacy, the proposed advanced protocol differs from the basic protocol with respect to two aspects: 1) node registration operation and 2) router–router routing operation.

The advanced protocol adopts an anonymous user registration process in which each user registers pseudonyms for different correspondent nodes with his servicing mesh router. Users update their pseudonyms registered on mesh routers for each session to remove linkability. When packets are transmitted between mesh routers, the advanced protocol employs the onion routing technique to protect information about the source mesh router and the destination router. Due to limited space, the following description only focuses on these differences.

A. Anonymous User Registration

In our basic protocol, the real identity of a network user is disclosed to his current service mesh router during registration for the purpose of routing. This, however, also enables the mesh router to track the user. To cope with this issue, it is essential to keep network users anonymous even to mesh routers while still maintaining the routing functionality. With this goal in mind, we propose a pseudonym-based registration approach that achieves not only complete anonymity for network users but routing efficiency as well. The proposed approach radically differs from previous approaches in that the previous pseudonym-based anonymous routing schemes all rely on expensive network-wide flooding for route establishment. Our approach, however, effectively avoids network-wide flooding by allowing each network user to register pseudonyms at mesh routers.

Specifically, in the proposed advanced protocol, a network node S registers to its current service mesh router using a pseudonym, which is denoted as $\overline{\mathrm{Nym}}_{SD}$, instead of its real identity S. S and D register $\overline{\mathrm{Nym}}_{SD}$ and $\overline{\mathrm{Nym}}_{DS}$ at their service mesh routers, respectively. Initially

$$\overline{\mathrm{Nym}}_{SD} = \overline{\mathrm{Nym}}_{SD}^{0} = H\left(\overline{k}_{SD}|S|D\right)$$
$$\overline{\mathrm{Nym}}_{DS} = \overline{\mathrm{Nym}}_{DS}^{0} = H\left(\overline{k}_{SD}|D|S\right).$$

Then, for each new session i $(i \geq 1)$, S and D update $\overline{\mathrm{Nym}}_{SD}$ as

$$\overline{\mathrm{Nym}}_{SD} = \overline{\mathrm{Nym}}_{SD}^{i} = H\left(H(\overline{k}_{SD})|\overline{\mathrm{Nym}}_{SD}^{i-1}\right)$$

and $\overline{\mathrm{Nym}}_{DS}$ is accordingly updated. It is crucial to update the pseudonym at each session as fixed pseudonyms provide linkability among different communication sessions of the same two nodes. S and D confirm their next shared $\overline{\mathrm{Nym}}_{SD}^{i+1}$ by the end of each session i. Note that since $\overline{\mathrm{Nym}}_{SD}$ are node-pair specific, a node might have to register multiple pseudonyms with the mesh router to maintain reachability by all of its peers. Node S stores the three-element tuple $\langle\overline{\mathrm{Nym}}_{SD}, \overline{\mathrm{Nym}}_{DS}, \overline{k}_{SD}\rangle$ for the efficient identification of message sender and key retrieval.

B. Onion-Routing-Based Router–Router Message Delivery

Due to anonymous user registration, mesh routers now have no access to the real identities of network nodes. However, they are still able to perform the routing functionality based on pseudonyms. In the basic protocol, a simple source routing approach is adopted for the mesh router to deliver packets to each other. This approach, however, allows mesh routers on the routing path to gain a lot of information with respect to the ongoing communication session. In other words, a mesh router on the path can determine that the session is 1) between an unknown node currently serviced by the source router R^S and the node D currently serviced by the destination router R^D in the basic protocol case or 2) between an unknown node currently serviced by R^S and a node $\overline{\mathrm{Nym}}_{DS}^{i}$ currently serviced by R^D in the case of anonymous user registration. Obviously, neither case is desirable according to the user privacy requirement class C_2. To address this issue, we propose to explore the onion routing technique for router–router message delivery, as described next.

Specifically, when a mesh router R^S receives a data packet, it first decrypts it and obtains $\overline{\mathrm{Nym}}_{DS}^{i}$ as the destination. It further identifies $\overline{\mathrm{Nym}}_{DS}^{i}$'s current service mesh router, for example, R^D, and chooses a path to R^D, if it does not have one already. The path selection can dynamically be based on, for example, traffic-balancing rules, and different paths may be chosen for the same destination so that no mesh router is overused, and randomness is introduced. Suppose a path chosen by a source router R^S for the destination $\overline{\mathrm{Nym}}_{DS}^{i}$ service by R^D to establish the path is as follows:

$$R^S \rightarrow R^i \rightarrow R^j \rightarrow R^t \rightarrow \cdots \rightarrow R^v \rightarrow R^D \rightarrow \overline{\mathrm{Nym}}_{DS}^{\,i}.$$

Then, the onion constructed by R^S is as follows:

$$R^S \rightarrow R^i : R^S, E_{\overline{k}_{R^S R^i}} \times \left(seqno, E_{pk_{R^i}}\left(R^i, sk_i, \ldots \left(E_{pk_{R^D}}(R^D, sk_D) \ldots\right), \mathtt{Pad}\right).$$

$\overline{k}_{R^S R^i}$ is a key shared between router R^S and R^i; pk_{R^i} is the public key of R^i; sk_i is a secret selected by source router R^S.

This onion is then processed by each router on the path as follows:

$$R^i \rightarrow R^j : R^i, E_{\overline{k}_{R^i R^j}} \times \left(seqno, E_{pk_{R^j}}\left(R^j, sk_j, \ldots \left(E_{pk_{R^D}}(R^D, sk_D) \ldots\right), \mathtt{Pad}\right).$$

Note that Pad is updated at each router to keep the same packet length. When the onion finally reaches R^D from some router R^v, it will have the following form:

$$R^v \rightarrow R^D : R^v, E_{\overline{k}_{R^v R^D}}\left(seqno, E_{pk_{R^D}}(R^D, sk_D), \mathtt{Pad}\right).$$

R^D finds itself as the destination after getting the inner part of the message by decryption and thus stops further packet propagation. Each router, for example, R^j, keeps a four-element tuple $\langle seqno, R^i, R^t, sk_j\rangle$ for both routing and message-processing purposes. This sets up a unique routing path indexed by $seqno$ between R^S and R^D. Now, R^S delivers the packets to $\overline{\mathrm{Nym}}_{DS}^{i}$ as follows:

$$R^S \rightarrow R^i : R^S, E_{\overline{k}_{R^S R^i}}\left(seqno, E_{sk_j}\left(E_{sk_t}\left(\ldots E_{sk_D} \times \left(\overline{\mathrm{Nym}}_{DS}^{\,i}, Msg\right)\ldots\right)\right), \mathtt{Pad}\right)$$

$$R^i \rightarrow R^j : R^i, E_{\overline{k}_{R^i R^j}}\left(seqno, E_{sk_t}\left(\ldots E_{sk_D} \times \left(\overline{\mathrm{Nym}}_{DS}^{\,i}, Msg\right)\ldots\right), \mathtt{Pad}\right)$$

$$\vdots$$

$$R^v \rightarrow R^D : R^v, E_{\overline{k}_{R^v R^D}}\left(seqno, E_{sk_D}\left(\overline{\mathrm{Nym}}_{DS}^{\,i}, Msg\right), \mathtt{Pad}\right).$$

At this point, R^D uses the same approach as in the basic protocol to deliver packets to $\overline{\mathrm{Nym}}_{DS}^{i}$. Note that R^S does not have to set up such a path to R^D every time for each different session; it has the flexibility to refresh the path either periodically or on demand. That is, the path setup is independent of individual sessions. This completes our description of the advanced scheme.

V. Security and Privacy Analysis

A. Security Analysis

The security of our two routing protocols is based on the keys constructed in the anonymous local session key establishment phase in which two session keys are generated. The key exchange in this phase makes use of group signatures to offer message authentication while keeping the participants' identities anonymous. On the other hand, the session keys are constructed based on the discrete logarithm problem (DLP) assumption. As we assume a computation-bounded adversary who is not able to solve the DLP problem, it can be claimed that the keys are established securely and anonymously.

Equipped with these session keys, the route-request and route-reply packets, as well as the registration packet, are well protected from outsiders, including both eavesdroppers and other unrelated legitimate nodes. Packets can only be recognized by legitimate forwarding nodes and opened by the expected destination node.

In the following, we will discuss the security features of our protocols, including access control, DoS attacks, and impersonation attacks.

Access Control: Our protocols ensure that only legitimate users can gain access to mesh networks. To be able to access the mesh network, a mobile user has to compute the correct session key k_{SD} and successfully register to the mesh router with it. Since the session key is computed by the Diffie–Hellman computation, only the user with a valid group signature signing key, generating r_S and Sig_S, can succeed in computing a correct session key. An adversary that cannot forge a valid signature over g^{r_S} of his choice would not be authenticated by the mesh router. If the attacker chooses to replay a previous message, then he has to solve the DLP problem, which is assumed hard in this paper.

Impersonation Attacks: Impersonation attacks are only possible for inside attackers. If a mesh client is compromised and its group private key is exposed to the attacker, then the attacker can use the group private key to compose valid route requests at will. As a result of the privacy feature of group signature, mesh routers or mesh clients cannot distinguish the identity of a specific mesh client by a group signature. Hence, the attacker can use the compromised group private key to masquerade as any other mesh client. However, if the attacker impersonating a mesh client tries to register the mesh client in a mesh router, then the conflicting registration information indicates that an impersonation attack is going on. With the help of the TTP, the mesh router can find out who is the imposter in order to revoke the compromised group private key.

DoS Attacks: DoS attacks aim to deplete resources, including computation capability, bandwidth, memory, energy, etc. DoS attacks can be divided into two groups: 1) DoS attacks by outside eavesdroppers and 2) DoS attacks by inside attackers. If the attacker is an outsider without knowing any key, then he is not able to forge correct packets that can pass packet verification. The packet verification process consists of symmetric crypto operations; hence, it does not lead to depletion of computation resources. Such DoS attacks have little effect on bandwidth, memory, or energy consumption. Therefore, DoS attacks by outside attackers are no threat to our protocols.

However, DoS attacks by insiders cause more damage than outside attackers. As the insider attacker knows a valid group signature key, he can use it to compose a large number of RREQs to deplete the mesh router's computation resource. Fortunately, the effectiveness of this DoS attack is largely limited in our protocols for the following reasons: First, DoS attacks aiming to exhaust network bandwidth are limited by the wireless signal range; their affect on nodes outside of the attacker's transmission range is reduced. Second, sending a large amount of route request can easily be detected by other nodes in the sender's neighborhood. Once such abnormal behaviors are detected, any other node can refuse to relay packets from the suspicious node. In the case in which an insider sends legitimate traffic without a valid destination into the WMN, the servicing source mesh router is not able to find a correct destination mesh router for both protocol suites. Hence, this traffic only happens within the subnet to which the attacker belongs, which limits the impact of the DoS attack. Moreover, the source mesh router can identify the node sending the traffic and exclude it from the network in the basic protocol suite. In the advanced protocol suite, the source mesh router needs the help of TTP to identify the sender of the malicious traffic, and then, the attacker can be exiled from the network. Furthermore, the TTP can revoke the group signature key of the suspicious node after detecting the suspicious node.

B. User Privacy Protection of Basic Protocol

We now analyze how the identity information of two communicating nodes is protected in the proposed basic protocol.

Against the Global Eavesdropper: In the basic protocol, all three different types of packets, i.e., route request, rout reply, and regular message packets, are indistinguishable from each other to the outsider. This is because they all take the same format and are of the same length. Moreover, none of these messages carry the real identity information of the source or destination nodes in plaintext. At the same time, the local key establishment protocol also reveals no information about the individual network node as the group signature technique is applied. Therefore, it is impossible for an eavesdropper to obtain the source or destination node identity information of any communication session. Such an eavesdropper cannot even tell if a routing protocol is being executed, even if he can monitor the global traffic.

Against Other Network Users: In the basic protocol, the real identity information of the source and destination nodes is always encrypted with either a node–router pairwise session key or a router–router preconfigured pairwise key in all three types of packets. Therefore, no other network users, including those on the routing path, are able to obtain the identity information. Moreover, no packet, regardless of its type, reveals the session originator or the session destination. When a network node on the routing path receives a packet from a previous-hop node, he can only know the packet type and the pseudonym of this previous-hop node after decryption. However, it has no ability to judge whether the packet is originated or just being relayed by this node. At the same time, the decrypted packet tells the user nothing about the destination information. The decrypted packet only tells the user the source router identity in case of a RREQ, which is the same of his own, or the pseudonym of the next-hop node in cases of route reply and regular message packets. This implies that other network users cannot link a communication session between two nodes, even to their current pseudonyms.

Against Mesh Routers and the NO: In the basic protocol, the mesh routers always have access to the network user's real identity information. Thus, the proposed protocol does not protect user privacy against mesh routers and the NO.

Against the Active Attacker: It is also clear that the proposed basic protocol does not protect user privacy against the active attacker either as the active attacker can always compromise and control a small number of mesh routers of his choice, according to our adversary model.

In summary, the proposed basic protocol successfully achieves the user privacy protection requirement C_1. That is, the protocol is perfect against other network users and the global passive attacker. As we will analyze later, this protocol is highly efficient compared with any other existing approaches in terms of communication overhead and bandwidth efficiency. However, it is also clear that the protocol 1) does not protect user privacy from mesh routers and, hence, the NO and 2) does not withstand active attacks if individual mesh routers can be compromised by the adversary.

C. User Privacy Protection of Advanced Protocol

We now further analyze how the identity information of two communicating nodes is protected in the proposed advanced protocol.

Against the Global Eavesdropper and Other Network Users: The analysis is the same as that earlier.

Against Mesh Routers and the NO: In the advanced protocol, none of the mesh routers now have access to the real identity information of the network users due to anonymous user registration. Therefore, none of the mesh routers, including those on the routing path, are able to obtain the identity information of the two communicating nodes. Next, the mesh routers on the routing path, in particular, the source router and the destination router, have access to the pseudonyms of the sender and receiver. However, the source router only knows that there is a session going on between two anonymous nodes with such respective pseudonyms as $\overline{\mathrm{Nym}}_{SD}^{i}$ and $\overline{\mathrm{Nym}}_{DS}^{i}$ that are currently serviced by itself and the destination router, respectively. However, the source router cannot link any two communication sessions to the same pair of users as the node-pair-specific pseudonyms are refreshed every session. With respect to the destination router, it, however, only knows that a message is being delivered to an anonymous node $\overline{\mathrm{Nym}}_{DS}^{i}$ within its coverage but has no knowledge about the source node or source mesh router due to the use of the onion routing technique. For the same reason, all of the remaining mesh routers on the routing path have no knowledge regarding the two communicating nodes and their respective service router. Therefore, to the NO (who owns the joint knowledge of all mesh routers), it knows that the session under inspection is between two anonymous nodes with such respective pseudonyms as $\overline{\mathrm{Nym}}_{SD}^{i}$ and $\overline{\mathrm{Nym}}_{DS}^{i}$ but nothing beyond this.

Against the Active Attacker: Based on the same preceding analysis, we can see that unless the active attacker compromises the source router of a given session, it cannot detect that there is a session going on between $\overline{\mathrm{Nym}}_{SD}^{i}$ and $\overline{\mathrm{Nym}}_{DS}^{i}$. Compromising any other mesh routers or network users will not give the attacker an advantage in this regard. In other words, the active attacker will not be able to tell that $\overline{\mathrm{Nym}}_{SD}^{i}$ and $\overline{\mathrm{Nym}}_{DS}^{i}$ are communicating by compromising any other mesh router, except for the source router. Furthermore, the active attacker cannot link two different sessions between the same two users together, even if it compromises the source router.

In summary, the proposed advanced protocol now satisfies the privacy requirement level C_2.

VI. Performance Evaluation

A typical WMN is characterized by low-power mobile devices and low-bandwidth wireless channels. Hence, the performance of routing protocols has great impact on their applicability and usability.

We implement our anonymous communication protocol for WMNs on simulator ns2 and evaluate its performance by comparing it with the Ad hoc On-Demand Distance Vector (AODV)-based (nonanonymous) mesh routing protocol. The network scenario parameters used in our simulation are listed in Table V. In the simulation scenario shown in Fig. 4, a mesh network of size 5000 $\times$ 5000 m consists of nine mesh routers with each router servicing 50 mobile nodes. Each mesh router and its servicing mobile nodes forms a subnet, and there are nine subnets in the network. The nine routers are aligned in a 3 $\times$ 3 matrix style and connected as shown in the figure.

TABLE V
Scenario Parameters

Simulation Time	600s
Scenario Dimension	5000m x 5000m
Wireless Radio Range	250m
Mobile Nodes Number	50x9
Mesh Routers Number	9
Node Speed	0-10m/s
Source-Destination Pairs	20 random pairs
Traffic Type	Bidirectional CBR 512-byte packet
Packet Frequency	1,2,4 Packet(s)/s
Wireless Bandwidth	2Mbps
Mesh Bandwidth	20Mbps
Key Update Interval	30s

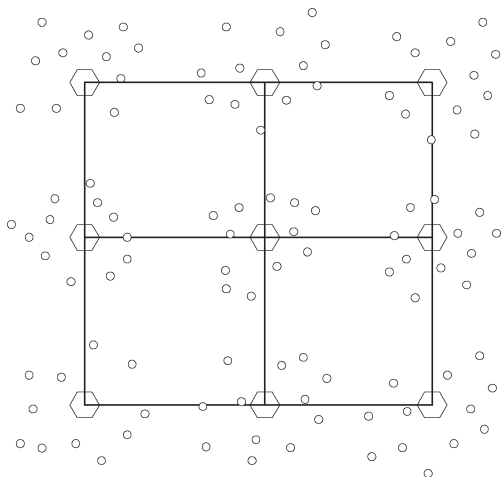


Fig. 4. Simulation model. Nine mesh routers, each connecting 50 mesh clients.

TABLE VI
Computation Time for Cryptographic Operations on a 1 GHZ Pentium III Platform

Group Signature Generation	24ms
Group Signature Verification	26ms
1024-bit DH Key Pair Generation	2.13ms
1024-bit DH Key Agreement	7.10ms

The mobile nodes are moving in the field according to the random waypoint model [7], and their average speeds range from 0 to 10 m/s. A bidirectional constant bit rate (CBR) traffic is generated for 20 random pairs to resemble point-to-point communication in the real world. The node pairs are selected from four diagonal subnets in the corners of the mesh network according to the following rules: a node from the left-bottom subnet communicating with another node from the right-top subnet or a node from the right-bottom subnet communicating with one node from the left-top subnet.

We choose the group signature scheme by Boneh *et al.* [2] for its constant size and efficiency. In addition, a 1024-bit prime number is used for Diffie–Hellman key exchange in our protocol. In the simulation, we follow the benchmarks on a 1-GHz Pentium III platform [6], [16], as shown in Table VI.

The status of network connection of the mesh network during our simulation is shown in Fig. 5(a). We count the number of node pairs having different hops between them within the four

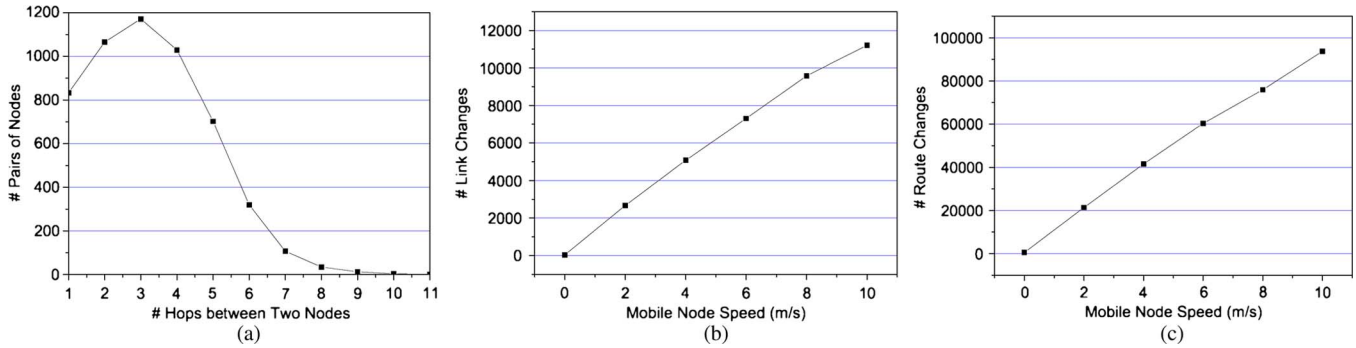


Fig. 5. Network simulation scenarios. (a) Number of hops between nodes. (b) Number of link changes at different speeds. (c) Number of route changes at different speeds.

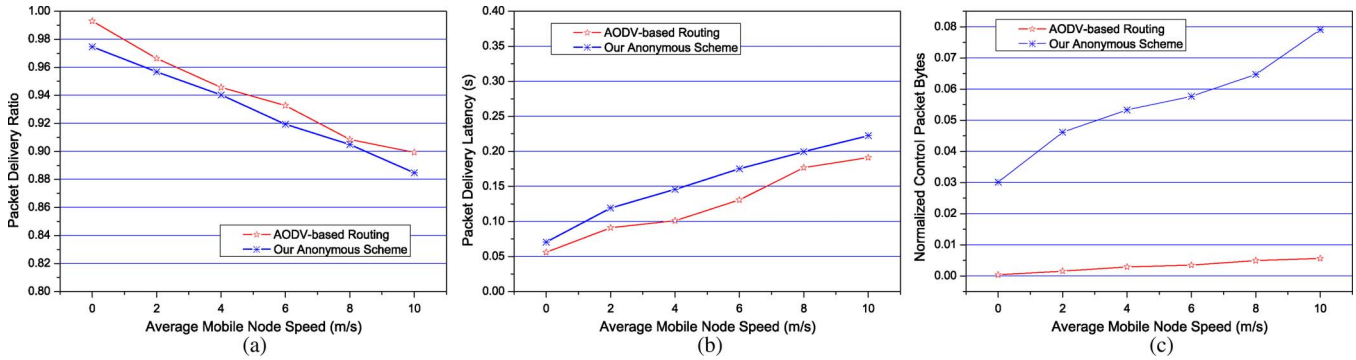


Fig. 6. Performance comparison between our protocols and AODV. (a) Packet delivery ratio comparison. (b) Packet delivery delay comparison. (c) Normalized control packet size comparison.

subnets. It conforms to the Poisson distribution according to the given curve.

Fig. 5(b) and (c) shows link changes and route changes for different node speeds in our simulation scenarios during the 600-s simulation time. It is clear that the number of link changes and the number of the route changes are perfectly positive proportional to node speed.

Fig. 6 compares performance of the basic protocol and the advanced protocol with nonanonymous AODV-based routing protocol under the same network settings. Specifically, we study the packet delivery ratio, the delivery latency, and the normalized control packet bytes of the three protocols. Due to dynamics and movements in the mesh network, all protocols exhibit lower performance at higher speeds. The nonanonymous AODV-based routing protocol achieves packet delivery ratio of about 90% and delivery latency of 200 ms, even at the average speed of 10 m/s. Although our basic protocol has worse performance than AODV, it still has a packet delivery ratio of more than 88% and a delivery latency of about 280 ms at the speed of 10 m/s. As the advanced protocol requires onion encryption and reregistration of mobile users for each session, it has a lower packer delivery ratio than AODV and the basic protocol. However, its packet delivery ratio is still larger than 86%, and its delivery latency is less than 300 ms. This is satisfactory for most applications. Fig. 6(c) shows the normalized control packet bytes on transmitting each data byte. It can be seen from the figure that the AODV-based scheme has very few normalized control packet bytes compared with our basic and advanced protocols. To send one data byte, the AODV-based scheme has to send less than 0.01 control packet bytes, whereas our basic and advanced protocols need to send 0.03–0.08 control packet bytes. Although our two protocols need more control bytes, the overhead due to control packets is still less than 10% of the data packet size.

Compared with AODV, the lower performance of the basic and advanced protocols is due to four facts: 1) In our basic and advanced protocols, only associated neighbors sharing local keys will forward route messages for each other; otherwise, route messages are simply dropped. 2) Local key expiration and node mobility lead to the disassociation of a node and its neighbors. Before neighboring nodes having shared local keys, no traffic can be passed between them. 3) Route repair is not applicable in both basic and advanced protocols for the sake of privacy protection, since route repair requires identity information about the destination. 4) Intermediate nodes cannot reply to a route request, even if they know the destination requested by the route request, as any intermediate node is supposed to know neither the source node nor the destination node. On the other hand, the advanced protocol's delivery ratio is even lower than the basic protocol because of the reregistration of one-time pseudonyms, which should be finished before a packet can successfully be delivered.

We also examine the impact of traffic load on performance of our basic protocol. Specifically, we study the performance of our basic protocol under three different traffic loads: 1) light, 2) medium, and 3) heavy traffic load with data rate of 1, 2, and 4 packet(s)/s, respectively. Fig. 7(a) shows the packet delivery ratio of the basic protocol under these three traffic loads. At

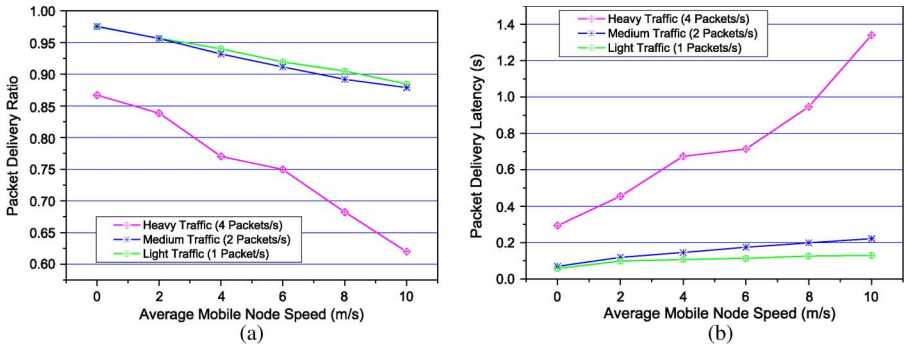


Fig. 7. Performance of protocol under different traffic loads. (a) Packet delivery ratio. (b) Packet delivery delay.

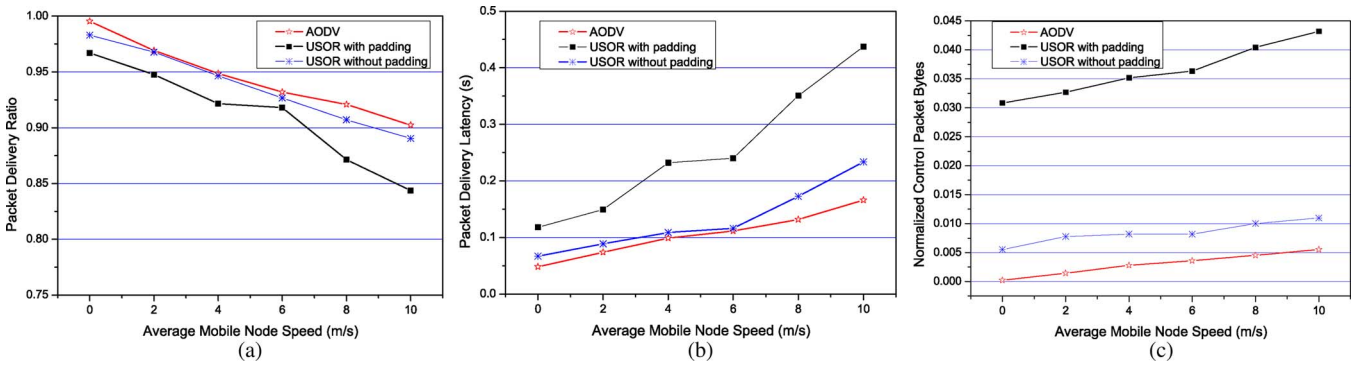


Fig. 8. Performance comparison of AODV. The basic protocol in case of packet padding. All packets are padded to 256 B, and traffic load is set to 2 packets/s. Bidirectional CBR traffic between 20 random pairs of nodes. (a) Packet delivery ratio comparison. (b) Packet delivery delay comparison. (c) Normalized control packet size comparison.

light or medium traffic load, the basic protocol can successfully deliver more than 87% packets. However, the success ratio dramatically decreases at heavy traffic load, and it is about 60% at a maximum average speed of 10 m/s. Fig. 7(b) illustrates the impact of traffic load on the packet delivery latency of our basic protocol. Under light or medium traffic load, the delivery latency is less than 220 ms, whereas under heavy traffic load, the delivery latency reaches more than 1.3 s at the speed of 10 m/s. Heavy traffic load results in more frequent and serious congestion and signal interference in the network. Packets get lost or dropped more easily, and they have to stay in the queue longer. More important, failure of local key establishment or reregistration of pseudonyms is more possible under heavier communication load in the network.

The impact of packet padding on routing performance is demonstrated in Fig. 8. We compare the performance of AODV, our protocol (basic) without padding, and our protocol (basic) with packet padding. The packet size is set to 256 B for all types of packets in the experiments, including RREQ and RREP packets. The traffic load is set to 2 packets/s with bidirectional CBR traffic between 20 random pairs of nodes. From the figures, it can be seen that the performance of our basic protocol downgrades a little due to packet padding. The delivery latency increases by about 150 ms on average, whereas the packet delivery ratio decreases by about 5%. As all control packets are padded to 256 B, the normalized control bytes increase to 0.045 from 0.015 after packet padding. Hence, we can say that the performance downgrade caused by packet padding is tolerable for most applications.

VII. Related Work

A. Privacy Protection in WMNs

WMNs present a promising way to provide wireless connectivity everywhere, but only modest research efforts have been put into the privacy-protection problem. Capkun *et al.* [4] have given a privacy-preserving scheme for the so-called hybrid ad hoc networks, which are actually WMNs. The main objective of their scheme is to provide anonymity and location privacy for mobile nodes in hybrid ad hoc networks (WMNs). Temporary public key pairs are used by each mobile node to anonymously establish pairwise secrets with its neighbors. These pairwise secrets in turn are used to secure routes to the access point. However, identifiers of mobile nodes have to be disclosed to access points so that some access point may be able to track a specific mobile user. On the other hand, an adversary is able to link messages by source and destination pseudonyms, which keep unchanged within a time slot. Moreover, the attacker is assumed to have only partial knowledge but not global knowledge of the network.

In [21], a structure called "Onion ring" is proposed to achieve routing privacy in WMNs. This scheme uses "Onion encryption" in a ring structure so that it is impossible for an adversary, even a global adversary, to distinguish the source node or the destination node. This scheme is able to identity the misbehaving nodes in order to evict them. However, it is not clear how to anonymously construct the ring in the first place, and furthermore, topology dynamics may make the scheme too inefficient.

The protocol proposed in [20] uses multiple paths for data delivery so that an attacker that is only able to observe a fraction of the traffic cannot obtain any meaningful information. This scheme, without cryptographic treatment, can only provide confidentiality or privacy with some probability, and it is not vulnerable to a globally passive attacker.

B. Privacy Protection in Ad Hoc Networks

Quite a few schemes have been proposed on anonymous routing in ad hoc networks. As WMNs share the same multihop property as ad hoc networks, we also review anonymous routing schemes for ad hoc networks here.

Recent years have witnessed a number of anonymous routing schemes proposed for ad hoc networks, including Anonymous On-Demand Routing (ANODR) [8], Anonymous Secure Routing (ASR) [24], Secure Distributed Anonymous Routing (SDAR) [3], Anonymous Routing Protocol for MANET (ARM) [17], Secure Anonymous Routing in Clustered MANET (SARC) [13], On-demand Anonymous Routing (ODAR) [19], Chain-based Anonymous Routing (CAR) [18], Discount-ANODR [22], and the Li and Ephremides (LE) scheme [9]. Essentially, these schemes exploit asymmetric cryptosystems to achieve privacy in routing discovery. These schemes fail to satisfy one or more privacy properties, as defined in [12].

The scheme ANODR proposed by Kong *et al.* [8] is the first to provide anonymity for routing in ad hoc networks. ANODR is based on onion routing for route discovery. It requires each node upon receiving a route request to generate a one-time public/private key pair, which is big computation burden. Furthermore, packets sent in ANODR are linkable, although ANODR is claimed to provide *anonymity in terms of unlinkability*. When two different packets sent using the same route are forwarded by the same intermediate node, they can easily be linked by their route pseudonyms, as route pseudonyms are computed by a public one-way function.

Following the work of ANODR, ASR [24], ARM [17], and SARC [13] use one-time public/private key pairs. SDAR [3], CAR [18], and ODAR [19] also use public key cryptosystems for secure anonymous routing, but they assume that long-term public/private key pairs have been set up on each node for anonymous communication. For the same reason, they are also vulnerable to such DoS attacks as ANODR, and the adversary can easily get to know the packet type by eavesdropping.

The LE scheme [9] adopts the pairing-based cryptographic approach to achieve anonymity, and it avoids exposing the destination node identity in the route requests by adding a long-term ID-based private key for each node.

The Discount-ANODR protocol [22], however, only uses symmetric cryptosystem for efficiency, but source anonymity and unlinkability are actually not achieved, as claimed by its authors.

VIII. Conclusion

In this paper, we have investigated the problem of privacy-preserving routing in WMNs and proposed two routing schemes to provide anonymity and unlinkability, as well as security, in WMNs. Both protocols have two stages: 1) the local session key establishment stage and 2) the anonymous routing-discovery stage. Relying on the privacy protection of group signature schemes, the first stage anonymously constructs session keys, which are used in the second stage to protect privacy in routing discovery. In the first protocol, what a mobile user needs is only a group signature signing key, and privacy against outsiders is protected in this protocol. As in our first protocol, where mesh routers are still able to identify mobile users and track them, we designed the second protocol to keep mobile users anonymous against mesh routers. Detailed security analysis and performance evaluation show that the proposed protocols are secure, privacy preserving, and efficient.

References

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: A survey," *Comput. Netw.*, vol. 47, no. 4, pp. 445–487, Mar. 2005.

[2] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proc. Adv. Cryptology–CRYPTO*, vol. 3152, *Lecture Notes in Computer Science*, 2004, pp. 41–55.

[3] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba, "SDAR: A secure distributed anonymous routing protocol for wireless and mobile ad hoc networks," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, Nov. 2004, pp. 618–624.

[4] S. Capkun, J. Hubaux, and M. Jakobsson, "Secure and privacy preserving communication in hybrid ad hoc networks," Swiss Fed. Inst. Technol.,-DI-ICA, Lausanne, Switzerland, 2004.

[5] D. Chaum and E. van Heyst, "Group signatures," in *Proc. Adv. Cryptology—EUROCRYPT*, vol. 547, *LNCS*, 1991, pp. 257–265.

[6] W. Dai, Crypto++ Benchmarks. [Online]. Available: http://www.cryptopp.com/benchmarks.html

[7] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, vol. 353. Norwell, MA: Kluwer, 1996, pp. 153–181.

[8] J. Kong and X. Hong, "ANODR: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks," in *Proc. 4th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2003, pp. 291–302.

[9] S. Li and A. Ephremides, "Anonymous routing: A cross-layer coupling between application and network layer," in *Proc. CISS*, Mar. 2006, pp. 22–24.

[10] W. Lou and K. Ren, "Security, privacy, and accountability in wireless access networks," *IEEE Wireless Commun. Mag.*, vol. 16, no. 4, Aug. 2009.

[11] Microsoft, Self Organizing Neighborhood Wireless Mesh Networks. [Online]. Available: http://www.research.microsoft.com/mesh/

[12] A. Pfitzmann and M. Hansen, Anonymity, Unobservability, and Pseudonymity: A Consolidated Proposal for Terminology, Draft, Jul. 2000.

[13] L. Qian, N. Song, and X. Li, "Secure anonymous routing in clustered multihop wireless ad hoc networks," in *Proc. CISS*, Mar. 2006, pp. 1629–1634.

[14] K. Ren and W. Lou, "A sophisticated privacy-enhanced yet accountable security framework for wireless mesh networks," in *Proc. 28th Int. Conf. Distrib. Comput. Syst.*, 2008, pp. 286–294.

[15] D. Samfat, R. Molva, and N. Asokan, "Untraceability in mobile networks," in *Proc 1st Annu. Int. Conf. Mobile Comput. Netw.*, 1995, pp. 26–36.

[16] M. Scott, MIRACL: Multiprecision Integer and Rational Arithmetic C/C++ Library. [Online]. Available: http://www.shamus.ie/index.php?page=Benchmarks

[17] S. Seys and B. Preneel, "ARM: Anonymous routing protocol for mobile ad hoc networks," in *Proc. 20th IEEE Int. Conf. AINA*, 2006, vol. 2, pp. 133–137.

[18] R. Shokri, N. Yazdani, and A. Khonsari, "Chain-based anonymous routing for wireless ad hoc networks," in *Proc. 4th IEEE CCNC*, 2007, pp. 297–302.

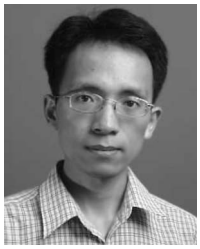
[19] D. Sy, R. Chen, and L. Bao, "ODAR: On-demand anonymous routing in ad hoc networks," in *Proc. IEEE Conf. MASS*, Oct. 2006, pp. 267–276.

[20] T. Wu, Y. Xue, and Y. Cui, "Preserving traffic privacy in wireless mesh networks," in *Proc. Int. Symp. WoWMoM*, 2006, pp. 459–461.

[21] X. Wu and N. Li, "Achieving privacy in mesh networks," in *Proc. 4th ACM Workshop Security Ad Hoc Sens. Netw.*, 2006, pp. 13–22.

[22] L. Yang, M. Jakobsson, and S. Wetzel, "Discount anonymous on demand routing for mobile ad hoc networks," in *Proc. 2nd Int. Conf. Security Privacy Commun. Netw.*, Aug. 2006, pp. 1–10.

[23] Y. Zhang and Y. Fang, "A secure authentication and billing architecture for wireless mesh networks," *Wirel. Netw.*, vol. 13, no. 5, pp. 663–678, Oct. 2007.
[24] B. Zhu, Z. Wan, F. Bao, R. H. Deng, and M. Kankanhalli, "Anonymous secure routing in mobile ad-hoc networks," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, 2004, pp. 102–108.



Zhiguo Wan received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 2002 and the Ph.D. degree in wireless network security from the National University of Singapore, Singapore, in 2007.

He is currently a Lecturer with the School of Software, Tsinghua University. His main research interests include cryptography and security in wireless networks.

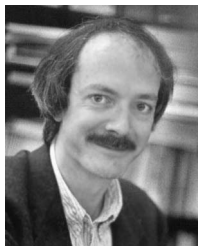
Kui Ren received the B.Eng. and M.Eng. degrees from Zhejiang University, Hangzhou, China, in 1998 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from Worcester Polytechnic Institute, Worcester, MA, in 2007.

He was a Research Assistant with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China, with the Institute for Infocomm Research, Singapore, and with the Information and Communications University, Daejeon, Korea. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago. His research is sponsored by the U.S. National Science Foundation. His research interests include network security and privacy; applied cryptography with current focus on security and privacy in cloud computing, lower layer attacks, and defense mechanisms for wireless networks; and sensor network security.

Dr. Ren is a member of the Association for Computing Machinery.

Bo Zhu received the B.Eng. and M.Eng. degrees from Wuhan University, Wuhan, China, in 1996 and 1999, respectively, and the M.Sc. and Ph.D. degrees from the National University of Singapore, Singapore, in 2002 and 2006, respectively.

He was a Postdoctoral Researcher with the Center for Secure Information Systems, George Mason University, Fairfax, VA, for two years. Since 2007, he has been an Assistant Professor with Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada. His research interests include security and privacy issues in various types of networks (including ad hoc/sensor/peer-to-peer/wireless networks, and Internet), data security, and software security.



Bart Preneel received the Master's degree in electrical engineering and the Doctorate degree in applied sciences (cryptology) from the Katholieke Universiteit Leuven (KUL), Leuven, Belgium, in 1987 and 1993, respectively.

He was a Visiting Professor with five universities in Europe and was a Research Fellow with the University of California at Berkeley. He is currently a Full Professor with KUL. He has authored or coauthored more than 200 reviewed scientific publications and is the holder of two patents. His main research interests are cryptography and information security.

Ming Gu received the B.S. degree from National University of Defense Technology, Changsa, China, in 1984 and the M.S. degree from Shenyang Institute of Computing Technology, Chinese Academy of Science, Shenyang, China, in 1986, both in computer science.

She is a Senior Researcher and the Vice Director of Key Laboratory for Information System Security, Ministry of Education, Tsinghua National Laboratory for Information Science and Technology, School of Software, Tsinghua University, Beijing, China. She has been in charge of more than ten national research projects and published more than 50 research papers in international conferences and journals. Her research interests include middleware techniques, formal methods in software, information system security.