This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

Lightweight Privacy-Preserving and Secure Communication Protocol for Hybrid Ad Hoc Wireless Networks

Mohamed M. E. A. Mahmoud[1], Sanaa Taha[2], Jelena Misic[3], Senior Member, IEEE, and Xuemin (Sherman) Shen[4], Fellow, IEEE

[1]*Department of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, Tennessee, 38505, USA*
[2]*Department of Information Technology, Faculty of Computers and Information, Cairo University, Cairo, Egypt*
[3]*Department of Computer Science, Ryerson University, Toronto, Ontario, M5B 2K3, Canada*
[4]*Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada*

***Abstract*—We propose lightweight protocol for securing communication and preserving users' anonymity and location privacy in hybrid ad hoc networks. Symmetric-key-cryptography operations and payment system are used to secure route discovery and data transmission. To reduce the overhead, the payment can be secured without submitting or processing payment proofs (receipts). To preserve users' anonymity with low overhead, we develop efficient pseudonym generation and trapdoor techniques that do not use the resource-consuming asymmetric-key cryptography. Pseudonyms do not require large storage area or frequently contacting a central unit for refilling. Our trapdoor technique uses only lightweight hashing operations. This is important because trapdoors may be processed by a large number of nodes. Developing low-overhead secure and privacy-preserving protocol is a real challenge due to the inherent contradictions: (1) securing the protocol requires each node to use one authenticated identity, but a permanent identity should not be used for privacy preservation; and (2) the low overhead requirement contradicts with the large overhead usually needed for preserving privacy and securing the communication. Our analysis and simulation results demonstrate that our protocol can preserve privacy and secure the communication with low overhead.**

***Index Terms*—**Anonymous and secure routing protocols, hybrid ad-hoc networks, privacy-preserving protocols, and payment systems.

1. INTRODUCTION

Hybrid ad hoc wireless network is a promising network architecture that incorporates ad hoc network with an infrastructure network including base stations [1]. The uplink mobile nodes may relay a source node's packets to the cell's base station, and the downlink mobile nodes may relay the packets to the destination node. This multihop packet relay can extend the base station's coverage area by enabling the nodes outside the coverage area to use the network. Multihop packet relay can increase throughput due to using the available bandwidth more efficiently. This is because the transmission interference area can be reduced by transmitting packets over shorter hops. However, involving autonomous and self-interested nodes in packet relay and the broadcast nature of radio transmission make the network highly vulnerable to serious security and privacy violation attacks.

Attackers may analyze the network transmissions to learn the users' communication activities, e.g., who communicates with whom, when, how long, etc, causing a severe threat for the users' privacy [2, 3]. The adversaries may try to trace the packets to learn the origin and/or the destination of the communications. They may also attempt to locate users in number of hops and track their movements. Revealing a user's location or the favorite locations he visits may lead to a physical attack. Attackers will exploit the fact that each node usually uses permanent identity and key to identify the node's transmissions and link them to a user. However, providing privacy preservation for hybrid ad hoc network poses many challenges.

Due to the open environment and the shared wireless medium, an attacker can intercept all the transmissions within the reception range of his radio receiver without the need to physically compromise a node. Moreover, multihop packet relay necessitates processing the packets by the mobile nodes to route them. This means that the packets' headers should not be encrypted to enable multihop routing. Unfortunately, attackers can inspect packets' headers to gain sensitive information. These attacks can be launched in an undetectable way by overhearing transmissions without disrupting the protocol.

Moreover, attackers may impersonate users or manipulate route establishment packets. For example, attackers may advertise false routing information to involve themselves in routes to collect sensitive information such as the pair of nodes that communicate and the nodes' locations in number of hops. Although the proper network operation requires the mobile nodes' cooperation in relaying others' packets, the selfish nodes will not cooperate without sufficient incentive to save their resources such as battery energy. This selfish behavior degrades the network performance significantly, which may cause the multihop communication to fail [4].

Developing low-overhead secure and privacy-preserving communication protocol is a real challenge due to the inherent contradictions. First, securing the protocol usually requires each node to use one authenticated identity, but a permanent identity should not be used to preserve the node's privacy. Second, reducing the protocol's overhead is necessary because the nodes are constrained by limited battery energy and computing power. However, the low-overhead requirement contradicts with the large overhead usually needed for preserving privacy and securing the communication, as we will discuss in Section 2.

In this paper, we propose a lightweight protocol for securing route establishment and data transmission, and preserving users' privacy in hybrid ad hoc wireless networks. To preserve users' anonymity, each node uses pseudonyms and one-time session key. Thus, if an adversary captures a packet, he cannot infer the real identities of the source, destination, or intermediate nodes. Our protocol enables the nodes to establish routes and send/relay packets without revealing their real identities or the identity of the destination node. A node's pseudonyms can authenticate it to the intended nodes without revealing its real identity. Packet tracing is prevented by changing the packet's appearance (bits) at each hop and using packet mixers. Therefore, even if an attacker eavesdrops on both the source and destination nodes, he cannot correlate their packets. To secure the protocol and preserve privacy, the intermediate nodes can ensure that the packets are sent by legitimate nodes without revealing the real identities of the source and destination nodes.

In order to secure the communication, we use hashing and symmetric-key-cryptography operations and a payment (or incentive) system. The system uses credits (or micropayment) to charge the nodes that send packets and reward those relaying them. The system can stimulate the nodes to relay others' packets to earn credits. Since the nodes pay for relaying their packets, the system can regulate packet transmission. Integrating privacy preservation with the payment system is essential to gain acceptance from the users to relay others' packets. Although the payment can make packet relay beneficial, most users will not sacrifice their privacy for earning credits.

To reduce the overhead, our protocol avoids the asymmetric-key cryptography because it consumes much resource, increases the packet delivery delay and degrades the packet delivery ratio [5]. We develop efficient pseudonym generation technique that uses hashing operations. The low overhead of the hashing operations will facilitate reducing the lifetime of each pseudonym and thus boosting the users' privacy. The end-to-end packet delay can be reduced because pseudonyms are fast to compute and can be pre-computed before receiving the packets. The pseudonyms are

Digital Object Indentifier 10.1109/TPDS.2013.298

authenticated and always synchronized and do not require large storage area or frequently contacting a central unit for refilling.

Trapdoor is a special token used to anonymously inform the destination node about the source node's call request. It is a key component in any anonymous communication protocol. The token (instead of the destination's identifier) is appended to the route request packet, where only the intended destination node can recognize it. A trapdoor may be broadcasted throughout the network and processed by a large number of nodes. The cost of creating and processing trapdoors should be minimized. We develop efficient trapdoor technique that does not require symmetric-key operations, but only lightweight hashing operations. Moreover, much overhead is usually consumed in submitting/processing payment proofs (or receipts) to secure the payment systems [6]. Our payment system can be secured without submitting/processing receipts. Our analysis and simulation results demonstrate that the proposed protocol can preserve the users' privacy and secure the communication with low overhead.

The remainder of this paper is organized as follows. Section 2 reviews the related works. Section 3 discusses the system models. We describe our protocol in Section 4. Security and privacy analyses and performance evaluation are given in Sections 5 and 6, respectively, followed by conclusion in Section 7.

2. RELATED WORKS

Lamparter et al. [7] propose a payment system for hybrid ad hoc network that is used to connect the nodes to the Internet via an Internet Service Provider. For each packet, the source node appends a signature to the full path identities, and the destination node signs a receipt and sends it to the last intermediate node in order to submit to the accounting center to update the nodes' credit account. Instead of generating a receipt per message or a group of messages, PIS [6] aims to reduce the receipts' submitting/processing overhead by generating a fixed-size receipt per session. ESIP [5] proposes a communication protocol that can be used for a payment system with limited use of asymmetric-key cryptography. The source and destination nodes generate signatures for only one packet and the efficient hashing operations are used in the other packets. Salem et al. [4] propose a payment system for hybrid ad hoc networks, where both the uplink and downlink packet relay can be multihop. When a route is broken, the nodes that receive the last packet should submit receipts to the base station to secure the payment.

Different from [1, 4 - 7], our protocol can preserve the users' privacy and secure the communication. It can also secure the payment without submitting receipts or using asymmetric-key cryptography to reduce the overhead.

Capkun et al. [8] proposed a privacy-preserving communication protocol for hybrid ad hoc network. Each node stores a set of public/private key pairs and certificates with different pseudonyms signed by a trusted party. The node uses a key pair to authenticate itself and to share symmetric keys with its neighbors. It periodically changes its public/private key pair and shares new symmetric keys with its neighbors to protect its anonymity. The nodes should contact the trusted party to refill their certified keys before they are exhausted. Each node also stores a routing table which contains the neighbors' pseudonyms and their distances to the base station in number of hops. Different from this protocol, our protocol is on-demand one that establishes routes only when needed. This can boost users' privacy because it does not send out unneeded routing advertisements.

In ANODR [9], the trapdoor is the encryption of the destination node's real identity and a random value by using the shared key with the destination node. However, the trapdoor technique is resource consuming because each node has to try to open the trapdoor with every key it shares with other nodes due to hiding the identities of the source and destination nodes to preserve their anonymity. Moreover, eavesdroppers can trace the packets along the route because their content does not change at each hop, and they can also know if a pair of nodes currently communicates.

In SDAR [10], the trapdoor is the encryption of the destination node's real identity and a one-time session key using the destination node's public key. Each node tries to open the trapdoor with its private key, and if it is not the destination, it uses the source node's one-time public key to add the encryption of its real identity, a one-time symmetric key, and a signature. However, the protocol is very resource consuming as it extensively use asymmetric-key cryptography operations. Moreover, the destination node learns the real identities and locations (in number of hops) of the intermediate nodes, and the location of the destination node is disclosed to the source node.

Unlike most communication protocols that are based on long-term identities, El-Defrawy et al. [11] argue that the location-centric communication paradigm is better-suited for privacy in mobile ad hoc networks. However, the location of the destination node and the distance between the source and the destination nodes are disclosed during route discovery.

Lin et al. [12] propose a privacy-preserving protocol based on group signature and identity-based signature techniques for vehicular ad hoc network. Ren et al. [13, 14] propose a protocol to enforce user access control and offer user privacy protection. The proposal is presented as a suite of authentication and key agreement protocols built upon a proposed short group signature variation. Mahmoud et al. [15] propose a scheme for protecting source nodes' location privacy in sensor networks. However, since these networks use different network and adversary models, they cannot be applicable for hybrid ad hoc networks effectively.

Zhang et al. [16] propose a secure communication protocol for ad hoc network using a combination of identity-based cryptography and threshold cryptography. In ARAN [17], the source node attaches its certificate, a signature, and the identity of the destination node to the route request (*RREQ*) packet. Each node verifies the signature, signs it, and forwards the packet to its neighbors. The destination node signs the route reply (*RREP*) packet and transmits it to the source node along the reverse path.

In Ariadne [18], the *RREQ* packet has the identities of the source and destination nodes, a randomly generated request identifier, and a message authentication code (*MAC*) computed over these elements with the key shared with the destination node. Each intermediate node attaches a *MAC* computed with the key shared with the destination node. The purpose of the per-hop *MAC* operations is to prevent the removal of identities from the packet. The destination node verifies the *MAC*, and sends *RREP* packet containing the list of identities obtained from the *RREQ* packet. In [19], Acs et al. introduce an attack against Ariadne protocol that takes advantage of the fact that the intermediate nodes cannot verify the *MAC*s of the previous nodes in the route because they do not know the keys used in the computation.

3. SYSTEM MODELS

3.1 Network Models

The considered hybrid ad hoc wireless network consists of mobile nodes, a trusted party (*Tp*), a set of base stations connected with each other and with *Tp*. The network is deployed for civilian applications, its lifetime is long, and the nodes have long relations with the network. *Tp* manages the nodes' credit accounts and maintains their symmetric keys. Each mobile node $\mathcal{N}_A$ should register with *Tp* to get a unique and long-term symmetric key K_A and identity ID_A. Without a valid key, the node cannot act as source, destination, or intermediate node.

A cell is the geographical area that is controlled by a base station. The transmission range of the base station is smaller than the radius of the cell. Thus, some mobile nodes will need to use the other nodes to relay their packets to communicate with the base station. The source base station (*Bs*) is the base station of the source node's cell, and the destination base station (*Bd*) is the base station of the destination node's cell. The source node ($\mathcal{N}_S$) sends packets to *Bs* (in multihops if necessary), *Bs* forwards the packets to *Bd* if the destination node ($\mathcal{N}_D$) resides in a different cell, and the packets are sent to $\mathcal{N}_D$, possibly in multiple hops.

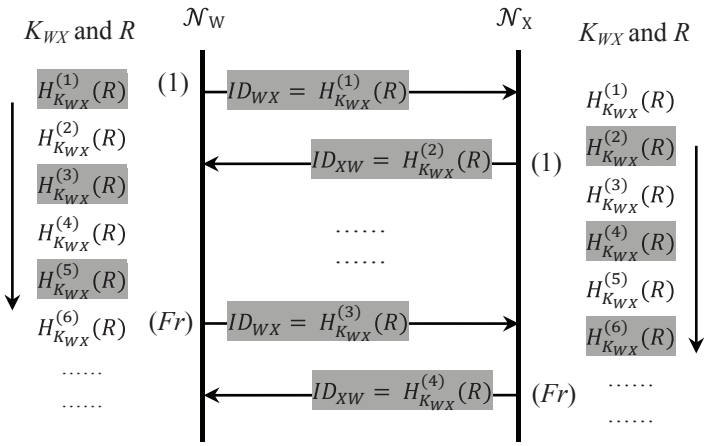


Fig. 1: Pseudonym generation technique.

The part of the route between $\mathcal{N}_S$ and *Bs* is called uplink, and the part of the route between *Bd* and $\mathcal{N}_D$ is called downlink.

Our payment model adopts a fair charging policy by supporting a cost sharing between the source and the destination nodes when both of them are interested in the communication. The payment-splitting ratio is adjustable and service-dependent, e.g., a DNS server should not pay for name resolution. The source and destination nodes are charged and the uplink intermediate nodes are rewarded when the source base station receives the source node's packets. The downlink intermediate nodes are rewarded when the destination base station receives acknowledgements of packet delivery. In Section 5, we will discuss that this payment model can stimulate packet relay and secure the payment without submitting receipts.

3.2 Adversary Model

The mobile nodes are potential attackers because they are autonomous, self-interested, and motivated to misbehave to increase their welfare. The network infrastructure including *Tp* and the base stations are secure. They are operated by a single operator that is interested to ensure the network security. The adversaries can be legitimate nodes which have valid keys to access the network, or external adversaries who are not members in the network. They may also work individually or collude with each other to launch sophisticated attacks.

We consider two different types of attackers. The first type of attackers would target the communication protocols including the payments system, the authentication protocol, and the route establishment and data transmission protocols. These attackers try to steal credits, pay less, and communicate freely. They can also attack the authentication protocol to impersonate other nodes and get an unauthorized use of the network, and manipulate/fabricate route establishment and data packets. The second type of attackers would target the users' privacy to know the users' communication activities, e.g., who is communicating to whom. They may also try to locate individual users in number of hops and track their movements to know the locations visited by them.

The base stations and *Tp* are trusted in performing auditing correctly and in preserving the nodes' location and identity privacy, but only *Tp* is trusted regarding the nodes' long-term keys. A node's identity and location should be known to the base stations to route the packets accordingly, but the long-term secret key is known only to *Tp*. The source and destination nodes do not know the location of each other or the real identities of the intermediate nodes. The intermediate nodes do not know the real identities or the locations of the source and destination nodes.

Our objective is to fully protect the payment system against colluding attackers. We also aim to protect the users' privacy against single and small-scale colluding attackers, and make the attacks launched by global eavesdroppers less effective. Global eavesdroppers can eavesdrop on every radio transmission on every communication link in the network at all time. In our protocol, global eavesdroppers may infer a communication route if there are few active sessions in the network, but they cannot link the nodes' pseudonyms to the real identities.

The symmetric-key cryptosystem and the keyed hash function are secure. Specifically, given a plaintext and its corresponding ciphertext, the attacker cannot infer the key used in the computation; computing the ciphertext without knowing the key is infeasible; and linking a plaintext to its ciphetext without knowing the key is infeasible. For the keyed hash function, computing the hash value without knowing the key is infeasible; the function is unidirectional in the sense that it is infeasible to compute the input value if the output hash value and the key are known; and given the input and the output hash values and without knowing the key, it is infeasible to link them or infer the key.

4. THE PROPOSED PROTOCOL

4.1 Pseudonym Generation Technique

The explicit use of a long-term identity or a permanent group of pseudonyms can violate users' privacy. Attackers can link the identity or the pseudonyms to the user, e.g., by analyzing the associated activities. To preserve users' anonymity, each pseudonym is used for short time in such a way that only the intended node can link the pseudonyms to each other. By this way, even if an attacker could link a pseudonym to the user in one occasion, he cannot violate the user's privacy for a long time and will not benefit from this conclusion in the future due to pseudonyms' periodic change and unlikability. Using a pseudonym for a long time enables attackers to collect much information about the visited locations by the anonymous user. Then, by analyzing this information, the attackers may identify the users and gain much information about their past visited locations.

From Fig. 1, if nodes $\mathcal{N}_W$ and $\mathcal{N}_X$ share a secret key K_{WX} (= K_{XW}) and a public random seed value R, they can generate shared pseudonyms by iteratively hashing R with K_{WX}. $H^{(n)}_{K_{WX}}(R)$ refers to the keyed hash value resulted from iteratively hashing R n times with K_{WX}. The pseudonyms generated from hashing R odd times ($ID_{WX} = H^{(2\times i-1)}_{K_{WX}}(R) = \{H^{(1)}_{K_{WX}}(R), H^{(3)}_{K_{WX}}(R), ..\}$) are used by $\mathcal{N}_W$, and those generated from hashing R even times ($ID_{XW} = H^{(2\times i)}_{K_{WX}}(R) = \{H^{(2)}_{K_{WX}}(R), H^{(4)}_{K_{WX}}(R), ..\}$) are used by $\mathcal{N}_X$, where i = 1, 2, .. etc. A one-way hash function, H, maps an input of any length to a fixed-length bit string. The function H is simple to compute yet computationally infeasible to invert. An example for a secure hash function is SHA-1 [20].

$\mathcal{N}_W$ can use a pseudonym, e.g., $H^{(1)}_{K_{WX}}(R)$, for multiple packets, and when the node uses the next pseudonym ($H^{(3)}_{K_{WX}}(R)$), $\mathcal{N}_X$ can know that $\mathcal{N}_W$ requests pseudonym change, so it uses the next pseudonym ($H^{(4)}_{K_{WX}}(R)$). In order to maintain pseudonym synchronization between $\mathcal{N}_W$ and $\mathcal{N}_X$, each node matches the other node's expected pseudonym with the current and next pseudonyms. For example, when $\mathcal{N}_W$ uses $H^{(1)}_{K_{WX}}(R)$, $\mathcal{N}_X$ matches $\mathcal{N}_W$'s pseudonym with $H^{(1)}_{K_{WX}}(R)$ and $H^{(3)}_{K_{WX}}(R)$. Moreover, pseudonyms are used in one direction, i.e., if $H^{(i)}_{K_{WX}}(R)$ is released, the pseudonyms $H^{(j)}_{K_{WX}}(R)$ for $j < i$ are no longer used. Each node also does not change its pseudonym more than once before the other node changes its pseudonym. By this way, if the packet containing a new pseudonym, e.g., $H^{(3)}_{K_{WX}}(R)$, is lost, the nodes do not lose synchronization because $\mathcal{N}_W$ will not use $H^{(5)}_{K_{WX}}(R)$ before $\mathcal{N}_X$ releases $H^{(4)}_{K_{WX}}(R)$ and shifts the window of expected pseudonyms from $\mathcal{N}_W$ to $H^{(3)}_{K_{WX}}(R)$ and $H^{(5)}_{K_{WX}}(R)$.

The requirement that a node should not change its pseudonym more than once before the other node changes its pseudonym, can work well if the two nodes exchange packets regularly. However, in some cases, such as route request packets, a node may send multiple packets before receiving a packet from the other node. This requirement can be relaxed if each node matches the other node's pseudonym against a window of L expected pseudonyms, where $L > 2$. The node should advance the window when it receives a pseudonym, where the last released pseudonym is always

on top of the window. Each node can release up to L pseudonyms before receiving a packet from the other node without losing synchronization.

Since privacy is a user-specific concept, our pseudonym generation technique allows users to trade off the privacy level and the computational overhead. Pseudonym change can be arbitrarily triggered by any of the two nodes without losing synchronization. The frequency of pseudonym change (Fr) is the number of packets that use one pseudonym. Higher privacy level is obtained when Fr decreases. The highest privacy level can be obtained when Fr = 1, i.e., a pseudonym is used for only one packet. Another advantage in our technique is that pseudonyms are computed by lightweight hashing operations and do not require large storage area or pseudonym refilling (unlike [8]). This means that Fr can be few (to boost nodes' privacy) with an acceptable overhead. Pseudonyms can also be computed before receiving a packet to avoid delaying the packet relay. Pseudonyms are not linkable to the real identity because the real identity is not used in computing them. An attacker cannot link the pseudonyms of a chain without knowing the secret key used in computations. Moreover, pseudonyms are authenticated because no one can compute them except the owner of the secret key.

4.2 Shared Keys and Authentication

In our protocol, each node uses three symmetric keys and pseudonym chains shared with Tp, base stations, and other nodes, as follows:-

1) Each node, e.g., $\mathcal{N}_X$, and Tp share a long-term key K_X. By using this key, they can generate a long-term pseudonym chain named ID_{XTp} and ID_{TpX}.

2) Each node, e.g., $\mathcal{N}_X$, shares a symmetric key and a pseudonym chain with its cell's base station. When the node handovers, the old base station sends the key and the pseudonyms to the new base station so that the key and pseudonym chain do not change and authentication process will not be needed. However, when $\mathcal{N}_X$ first joins the network or handover fails to keep the keep the keys and the pseudonyms, Tp mutually authenticates the node and the base station and distributes shared key to be used in generating pseudonyms. Tp should be involved because the base station does not know the node's long-term key. As shown in Fig. 2, $\mathcal{N}_X$ initiates the authentication process by sending an *Authentication Request* (*AREQ*) packet to the base station, probably through multi-hopping. *AREQ* packet has a fresh pseudonym shared with Tp (ID_{XTp}) and the encryption of ID_{XTp} and its real identity (ID_X), where $(ID_{XTp}, ID_X)K_X$ refers to the ciphertext resulted from encrypting "ID_{XTp}, ID_X" with K_X.

AREQ packet authenticates $\mathcal{N}_X$ to Tp because the secret key K_X is required to compose valid packet. Without knowing K_X, it is infeasible to compute valid $(ID_{XTp}, ID_X)K_X$ and fresh ID_{XTp}. The base station (Bs) forwards the request to Tp which checks whether the pseudonym is for a registered user and replies with the node's real identity, the shared key between $\mathcal{N}_X$ and Bs ($K_{XBs} = K_{BsX}$), and the seed of the pseudonym chain (R). With this packet, Tp authenticates $\mathcal{N}_X$ to the base station. R and K_{XBs} are used to generate pseudonyms shared between $\mathcal{N}_X$ and Bs. The base station sends *Authentication Reply* (*AREP*) packet to $\mathcal{N}_X$. $\mathcal{N}_X$ can ensure that the packet is sent from Tp because it is infeasible to compute ID_{TpX} and $(K_{XBs}, ID_{TpX}, R)K_X$ without knowing the secret key K_X. By this way, Tp mutually authenticates $\mathcal{N}_X$ and Bs without revealing the node's long-term secret key.

3) In route discovery phase, the base station mutually authenticates each two neighboring nodes, e.g., $\mathcal{N}_W$ and $\mathcal{N}_X$, and distributes a one-time/one-route shared key ($K_{WX} = K_{XW}$) to generate pseudonym chain ID_{WX} and ID_{XW}. If two nodes are neighbors in different active routes, they will have a different key and pseudonym chain per route, i.e., each key and pseudonym chain are unique for each route and two neighbors. By this way, routes can be identified by pseudonym chains, which is necessary for successful packet routing.

$\mathcal{N}_X \rightarrow Bs \rightarrow Tp$: $<AREQ, ID_{XTp}, (ID_{XTp}, ID_X)K_X>$

$Tp \rightarrow Bs$: $<(ID_X, K_{BsX}, R, ID_{TpX}, (K_{XBs}, ID_{TpX}, R)K_X)K_{TpBs}>$

$Bs \rightarrow \mathcal{N}_X$: $<AREP, ID_{TpX}, (K_{XBs}, ID_{TpX}, R)K_X>$

Fig. 2: Authentication phase.

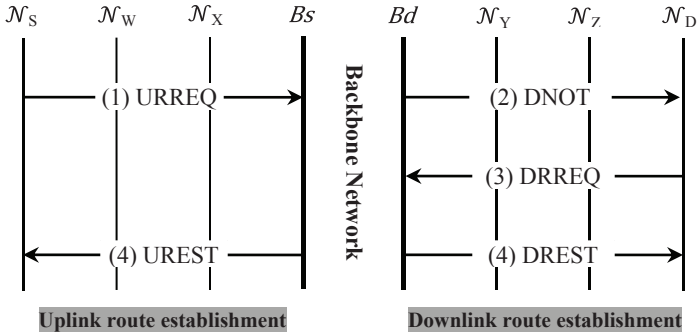
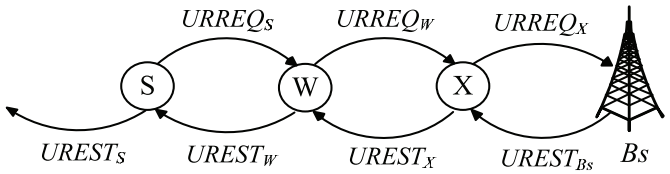


Fig. 3: Route discovery packets.



$\boldsymbol{URREQ_S}$: $<Un_i, TTL, (Un_i, ID_S, ID_D, P_L, Pad)K_{SBs}>$

$\boldsymbol{URREQ_W}$: $<Un_i, \boldsymbol{TTL\text{-}1}, \boldsymbol{ID_{WBs}}, ((Un_i, ID_S, ID_D, P_L, Pad)K_{SBs})\boldsymbol{K_{WBs}}>$

$\boldsymbol{URREQ_X}$: $<Un_i, \boldsymbol{TTL\text{-}2}, \boldsymbol{ID_{XBs}}, ((ID_{WBs}, ((Un_i, ID_S, ID_D, P_L, Pad)K_{SBs})K_{WBs}))\boldsymbol{K_{XBs}}>$

$\boldsymbol{UREST_{Bs}}$: $<ID_{BsX}, (Un_i, K_{XBs}, ID_{BsW}, (Un_i, K_{WX}, ID_{BsS}, (Un_i, K_{SW}, Pad)K_{BsS})K_{BsW})K_{BsX}>$

$\boldsymbol{UREST_X}$: $<ID_{BsW}, (Un_i, K_{WX}, ID_{BsS}, (Un_i, K_{SW}, Pad)K_{BsS})K_{BsW}>$

$\boldsymbol{UREST_W}$: $<ID_{BsS}, (Un_i, K_{SW}, Pad)K_{BsS}>$

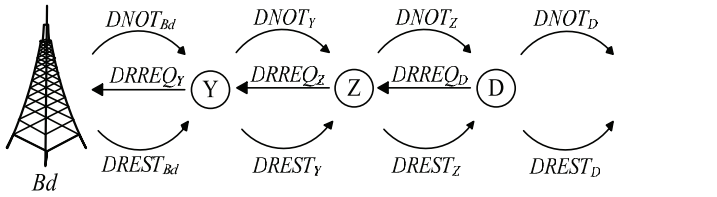
$\boldsymbol{UREST_S}$: $<Pad>$

Fig. 4: Anonymous uplink route establishment.

4.3 Anonymous Route Discovery

From Fig. 3, when a source node $\mathcal{N}_S$ wants to communicate with another node $\mathcal{N}_D$, two routes should be established: (1) uplink route between $\mathcal{N}$s and the source node's base station (Bs); and (2) downlink route between the destination node's base station (Bd) and $\mathcal{N}_D$. To establish end-to-end route, $\mathcal{N}_S$ broadcasts the *Uplink Route Request Packet* (*URREQ*) and Bs forwards a call request to the destination node's base station if $\mathcal{N}_D$ resides in a different cell. Bd broadcasts *Destination Notification Packet* (*DNOT*) if it does not know a route to $\mathcal{N}_D$ to inform the node about the call request. $\mathcal{N}_D$ replies with *Downlink Route Request Packet* (*DRREQ*) to enable Bd to know the identities of the intermediate nodes in the route. Finally, Bs and Bd send *Uplink Route Establishment Packet* (*UREST*) and *Downlink Route Establishment Packet* (*DREST*), respectively to establish the route.

***Uplink Route Request Packet* (*URREQ*):** As shown in Fig. 4, the source node initiates route discovery by broadcasting *URREQ* packet containing a unique request identifier (Un_i), time to live (*TTL*), and the encryption of Un_i, the source and the destination nodes' real identities, dummy bits called padding (*Pad*), and the padding length (P_L). Un_i is the pseudonym shared with Bs (ID_{SBs}) and time stamp. Each node and the base station process only the first received *URREQ* packet and discard all further packets having the identifier Un_i. Using this identifier is necessary to avoid routing loops and broadcast explosion that causes broadcasting

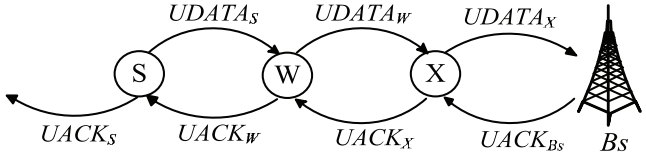


DNOT$_{Bd}$: <Dn_i, TTL, (Dn_i, ID_D, ID_S)K_{BdD}>
DNOT$_Y$: <Dn_i, **TTL-1**, (Dn_i, ID_D, ID_S)K_{BdD}>
DNOT$_Z$: <Dn_i, **TTL-2**, (Dn_i, ID_D, ID_S)K_{BdD}>
DNOT$_D$: <Dn_i, **TTL-3**, (Dn_i, ID_D, ID_S)K_{BdD}>

DRREQ$_D$: <Dn_i, TTL, (Dn_i, ID_S, ID_D, P_L, Pad)K_{DBd}>
DRREQ$_Z$: <Dn_i, **TTL-1**, **ID_{ZBd}**, ((Dn_i, ID_S, ID_D, P_L, Pad)K_{DBd})**K_{ZBd}**>
DRREQ$_Y$: <Dn_i, **TTL-2**, **ID_{YBd}**, ((ID_{ZBd}, ((Dn_i, ID_S, ID_D, P_L, Pad)K_{DBd})K_{ZBd}))**K_{YBd}**>

DREST$_{Bd}$: <ID_{BdY}, (Dn_i, K_{YBd}, ID_{BdZ}, (Dn_i, K_{ZY}, ID_{BdD}, (Dn_i, K_{DZ}, Pad)K_{BdD}) K_{BdZ})K_{BdY}>
DREST$_Y$: <ID_{BdZ}, (Dn_i, K_{ZY}, ID_{BdD}, (Dn_i, K_{DZ}, Pad)K_{BdD})K_{BdZ}>
DREST$_Z$: <ID_{BdD}, (Dn_i, K_{DZ}, Pad)K_{BdD}>
DREST$_D$: <Pad>

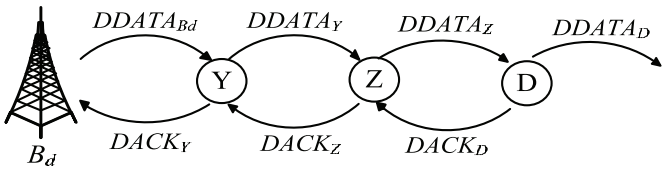
Fig. 5: Anonymous downlink route establishment.



UDATA$_S$: <ID_{SW}, (Un_i, C, M_C, $H(M_C)$)K_{SBs}>
UDATA$_W$: <**ID_{WX}**, ((Un_i, C, M_C, $H(M_C)$)K_{SBs})**K_{WBs}**>
UDATA$_X$: <**ID_{XBs}**, (((Un_i, C, M_C, $H(M_C)$)K_{SBs})K_{WBs})**K_{XBs}**>

UACK$_{Bs}$: <ID_{BsX}, (((Un_i, C, $H(M_C)$)K_{BsS})K_{BsW})K_{BsX}>
UACK$_X$: <**ID_{XW}**, ((Un_i, C, $H(M_C)$)K_{BsS})K_{BsW}>
UACK$_W$: <**ID_{WS}**, (Un_i, C, $H(M_C)$)K_{BsS}>
UACK$_S$: <Random>

Fig. 6: Anonymous uplink data transmission.



DDATA$_{Bd}$: <ID_{BdY}, (((Dn_i, C, M_C, $H(M_C)$)K_{BdD})K_{BdZ})K_{BdY}>
DDATA$_Y$: <**ID_{YZ}**, ((Dn_i, C, M_C, $H(M_C)$)K_{BdD})K_{BdZ}>
DDATA$_Z$: <**ID_{ZD}**, (Dn_i, C, M_C, $H(M_C)$)K_{BdD}>
DDATA$_D$: <Random>

DACK$_D$: <ID_{DZ}, (Dn_i, C, $H(M_C)$)K_{DBd}>
DACK$_Z$: <**ID_{ZY}**, ((Dn_i, C, $H(M_C)$)K_{DBd})**K_{ZBd}**>
DACK$_Y$: <**ID_{YBd}**, (((Dn_i, C, $H(M_C)$)K_{DBd})K_{ZBd})**K_{YBd}**>

Fig. 7: Anonymous downlink data transmission.

the same packet each time it is received from a neighbor. This identifier does not reveal much information because the packets are broadcasted. ID_{SBs} and the encrypted part authenticate $_S$ to *Bs*, which is necessary for authorizing the network access and securing the payment. *TTL* is used to bind the request propagation area. Each node decrements *TTL*, and once it is zero the request is no longer broadcasted.

Each node adds the pseudonym shared with Bs, encrypts the previous node's pseudonym and the encrypted part with the shared key with Bs, and broadcasts the request. As the packet moves towards the base station, it stores the pseudonyms of the nodes in the route. For the first received *URREQ* packet, Bs decrypts the encryption layers to tell the identities of the source, intermediate, and destination nodes. Then, it sends call request to *Bd* if $\mathcal{N}_D$ resides in a different cell. Since the packet length grows with fixed amount of data as it is relayed, the attackers may try to locate the source node's location either from *TTL* or the packet size. To protect the location privacy of $\mathcal{N}_S$ and to confuse its neighbors whether the packet is originated from or relayed by $\mathcal{N}_S$, a random-length padding is added and the initial *TTL* is variable value. Since Un_i varies over time, each time a node sends *URREQ* packet to the same destination, the packet looks different in spite of using the same key. This can thwart fingerprint recording attack as will be discussed in Section 5.

***Destination Notification Packet* (*DNOT*):** From Fig. 5, after the destination base station (*Bd*) receives a call request for a node in its cell, it notifies the node by broadcasting *Destination Notification Packet* (*DNOT*). The packet contains a unique identifier (Dn_i) that has the pseudonym shared with $\mathcal{N}_D$ and time stamp. The packet also contains Time-to-Live (*TTL*), and the encryption of Dn_i and the destination and source nodes' identities with using the shared key with $\mathcal{N}_D$. Padding is not needed because preserving the base station's location privacy is not important.

After receiving the packet, each node first checks whether it is the intended destination by checking if the attached pseudonym is in the list of expected pseudonyms. If so, the node decrypts the encryption to tell the identity of the source node, and sends *DRREQ* packet. If it is not the destination and *TTL* is greater than zero, the node decrements *TTL* and broadcasts the packet. Each node processes each notification once and drops any further packets with the same identifier. The destination node broadcasts the *DNOT* packet as well to deprive its neighbors from inferring that the destination is a one hop neighbor. Thus, all *DNOT* packets are transmitted for *TTL* hops regardless of the location of the destination node to preserve its location privacy.

***Downlink Route Request Packet* (*DRREQ*):** Fig. 5 shows that the destination node composes and broadcasts the *DRREQ* packet. Processing the packet is similar to that of the *URREQ* packet.

***Uplink Route Establishment Packet* (*UREST*):** The objective of the *UREST* packet is to inform the uplink intermediate nodes to act as relays and to distribute the session keys shared between each two neighboring nodes. From Fig. 4, each intermediate node removes one encryption layer by using the key shared with *Bs*, stores the session key shared with the previous neighbor in the route, and relays the packet after removing Un_i and its pseudonym and key. The node hashes this key to compute the key shared with the other neighbor, e.g., node $\mathcal{N}_W$ uses K_{WX} to communicate with $\mathcal{N}_X$ and $K_{WS} = H_{K_{WBs}}(K_{WX}, 1)$, $H_{K_{WBs}}(K_{WX}, 2)$ .. etc, to communicate with $\mathcal{N}_S$. Obviously, K_{WS} should be similar to K_{SW} distributed by *Bs*. By this way, the number of distributed keys can be nearly halved to reduce the packet overhead. Padding is added to make it infeasible to infer the source node's location from the packet size. The source node relays the packet as well to protect its location privacy from its neighbors.

***Downlink Route Establishment Packet* (*DREST*):** This packet informs the downlink intermediate nodes to act as relays and distributes the session keys shared between each two neighboring nodes. The packet's format is similar to that of *UREST* packet.

By the route discovery packets, the base station and the nodes mutually authenticate each other, and each two neighboring nodes mutually authenticate each other with the assistance of the base station. These authentication processes are necessary to secure the routing protocol and the payment.

4.4 Data Transmission

After receiving the *UREST* packet, $\mathcal{N}_S$ starts transmitting data to the destination through the established route. As shown in Fig. 6, the data packet at the source node has the shared pseudonym

with the next node in the route (ID_{SW}), and the encryption of Un_i, the message's number (C), and the message (M_C) and its hash value ($H(M_C)$). If a node simultaneously participates in different routes, it stores each route's pseudonyms and keys in memory, so that it can quickly verify whether a packet is targeted at it or not and which pseudonym/key it has to use. From Fig. 6, each intermediate node replaces the incoming pseudonym with the outgoing one shared with the next node, and encrypts the iteratively-encrypted part with the key shared with base station. Thus, when the packet reaches the source base station, it should have a layered-encrypted ciphertext that is computed by all the nodes in the uplink route. The source base station removes the encryption layers by iteratively decrypting the packet with the keys shared with the nodes in the route. It also verifies the attached hash value to make sure that the message has not been modified during transmission. If this verification fails, the base station sends a negative acknowledgement to the source node to retransmit the message, otherwise, it forwards the message to the destination base station if the destination node resides in a different cell.

As shown in Fig. 7, the destination base station iteratively encrypts the message with the keys shared with the nodes in the route, and sends the packet to the first node in the route ($\mathcal{N}_Y$). Each intermediate node removes one encryption layer and replaces the pseudonym with the one shared with the next node. The destination node decrypts the packet and verifies the hash value to ensure the message's integrity and authenticity. For reliable communication, the destination node sends back an acknowledgement packet when it receives a correct message.

Note that the session keys are used only for generating one-time pseudonyms, but the keys shared with the base station are used in encryption to prevent manipulating the messages and secure the payment by thwarting free riding attack, as will be discussed in Section 5. Moreover, the time element in Un_i can guarantee that the packets look different if the same message is sent at different times. As will be discussed in Section 5, this can protect the nodes' anonymity against fingerprint recording attack. To reduce the overhead on the mobile nodes, each node performs one encryption/decryption operation, but the base station performs more operations. To simplify our description, we focus on unidirectional data transmission, but the protocol can also be used for bidirectional communication.

A route is broken when two neighboring nodes in the route cannot communicate, e.g., because they are no longer in transmission range due to node mobility. When a node forwards a packet to its neighbor, it can confirm that the neighbor received the packet by link-layer acknowledgment. A route is considered broken if a node does not receive an acknowledgment after a limited number of packet retransmissions. In this case, the node should send an error packet to the base station to reestablish the route. Moreover, the base station can determine route breakage by re-starting a timer each time it receives a data or acknowledgement packet, and the route is considered broken if the timer expires. To reduce the overhead of reconnecting the broken routes, the base station can cache the routing information when it receives route discovery packets and uses this information when it needs to establish a route by unicasting a *DREST* packet.

4.5 Accounting and Auditing

When the source base station receives a data packet, the source and destination nodes are charged and the uplink intermediate nodes are rewarded. The downlink intermediate nodes are rewarded when the destination base station receives acknowledgement for packet delivery. Unlike [4] that uses receipts to make packet relay rational action for the nodes, our payment model can do that without using receipts as will be discussed in Section 5. To manage the payment without instantaneously contacting *Tp* in each session, the base stations can manage the payment of the nodes in their cells and update the nodes' accounts stored in *Tp*. The base stations can also enforce access control by rejecting a node's call request if it does not have sufficient credits.

5. SECURITY AND PRIVACY ANALYSES

5.1 Communication Security

The per-hop encryption/decryption operations can thwart several attacks. Removing the encryptions and verifying the correctness of the message implicitly authenticates the intermediate nodes, verifies the hop count, and ensures that the packet is relayed through the route it was supposed to take. For *URREQ* and *DRREQ* packets, the per-hop encryption operations can secure the routing by preventing manipulating the routing information including the identities of the nodes in the route. Moreover, the hop-by-hop encryption/decryption operations make the packets look different as they are relayed, which can boost privacy preservation, as will be discussed in subsection 5.2.

In *free-riding* attack, two colluding nodes, e.g., $\mathcal{N}_{C1}$ and $\mathcal{N}_{C2}$, in a legitimate session manipulate the packets to piggyback their data to communicate freely. The proposed payment systems in [1, 6, 7] use asymmetric-key cryptography to thwart this attack by signing the messages and verifying the signatures by intermediate nodes, so that manipulated packets can be detected and dropped. However, the asymmetric-key cryptography is resource consuming and usually inefficient in preserving users' privacy. In our protocol, the per-hop encryption/decryption operations can thwart this attack because the data sent by $\mathcal{N}_{C1}$ cannot be interpreted by $\mathcal{N}_{C2}$ due to encrypting (or decrypting) it by at least one intermediate node. The nodes should use the keys shared with the base station in the encryption/decryption operations because using the session keys cannot thwart the attack if there is only one intermediate node between colluders: $\mathcal{N}_{C1}$ can piggyback data and encrypt the packet with the session key K_{CIV} shared with the victim node $\mathcal{N}_V$; $\mathcal{N}_V$ encrypts the packet with the key (K_{VC2}) shared with the next node $\mathcal{N}_{C2}$; the colluding nodes can retrieve the data because they know K_{CIV} and K_{VC2}.

The existing payment systems can guarantee the rationality of packet relaying by rewarding the nodes for every relayed packet even if it does not reach the destination. However, this requires submitting payment receipts when a route is broken to identify the last node that relayed the packet. The nodes can collude to earn credits with consuming low resource by relaying only the security token (e.g., signature) to compose valid receipt instead of relaying the whole packet. Our payment system can guarantee the rationality of packet relaying, encourage the nodes' cooperation, and counteract rational cheating actions without the overhead of storing, submitting, and processing receipts, as follows:-

1) The uplink and downlink intermediate nodes are motivated to relay the data packets because they are rewarded only when the source base station and destination node receive the packets, and thus packet dropping is an irrational action.
2) Relaying the route discovery packets is beneficial for the nodes to participate in routes and thus earn credits. Relaying *UACK* packets can trigger the source node to generate more packets, and thus the nodes can earn more credits. Relaying *DACK* packets is beneficial for the downlink nodes because they are rewarded when the packets reach the base station.
3) If the source and destination nodes are charged only for delivered packets, they can communicate freely if the destination node denies receiving the packets or a colluding intermediate node claims route breakage. To prevent this, the source and destination nodes are charged for all sent packets.

For *credit-overspending* attack, the nodes may spend more than the amount of credits they have at the communication time. Most of the existing payment systems [1, 5 - 7] are vulnerable to this attack because they use post-paid payment policy, where the nodes communicate first and pay later. In our payment system, the base stations can thwart this attack because they can know the nodes' total credits at the communication time.

For *man-in-the-middle* attack, an attacker residing between a victim node and the base station (or *Tp*) may attempt to obtain the key shared between the node and the base station. The attacker can use the key to establish sessions that are payable by the victim

node or launch attacks under its name. Our protocol is not vulnerable to this attack because the shared key between a node and a base station is encrypted with the node's long-term key, and thus no one can obtain this key except the intended node. For *impersonation* attack, attackers attempt to impersonate *Tp*, base stations, or other nodes, e.g., to unfairly obtain free service or implicate victim nodes in malicious actions. This attack is infeasible in our protocol because the nodes have to authenticate themselves using the long-term keys shared with *Tp* to share a key with a base station. Without knowing this secret key, attackers cannot send valid packets under the name of others.

For *fabrication of route discovery packets*, an attacker tries to fabricate route discovery packets to impersonate a source or a destination node or a base station. This is infeasible in our protocol because the nodes' secret keys should be used to compose valid packets. For *packet-replay* attack, attackers may record valid packets and replay them in different locations or time to establish sessions under the name of others to communicate freely or violate users' privacy. In our protocol, the attackers cannot compose *URREQ* packet with valid timestamp and fresh pseudonym without knowing the secret keys of the victim nodes.

For *packet modification* attack, if an attacker manipulates a packet in our protocol, the packet integrity check fails at the base station and destination node. The attackers cannot manipulate the route request packets successfully, e.g., by adding or removing nodes' identities, because they do not know the nodes' secret keys. In *session-hijacking* attack, attackers try to hijack a session after it is established by legitimate nodes to communicate for free. Since the source node's encryption is required in each data packet, the attacker cannot compose valid packets without knowing the node's secret key and thus invalid packets can be detected and dropped.

For *access control*, our protocol ensures that only legitimate users can access the network to prevent unauthorized use. Only legitimate nodes can share keys with base stations and the nodes cannot communicate without these keys. For *authenticated packet forwarding*, although an intermediate node should not know the identity of the other nodes in a route, it should ensure that it relays packets for legitimate nodes to prevent unauthorized use of the network and to ensure that it will be rewarded for relaying packets. In our protocol, *Tp* mutually authenticates the nodes and base stations, and a base station authenticates each node to its neighbors in the route. With these authentications, each node can ensure that it relays packets sent from legitimate nodes.

5.2 Privacy Preservation

For *packet correlation*, attackers try to correlate the packets sent in one route at different hops by finding information that indicate that the packets belong to the same traffic flow. Attackers will try to correlate packets as follows:-

a) *Packet-content correlation*

In our protocol, the encryption/decryption operations and changing pseudonyms at each intermediate node guarantee that a packet looks quite different as it is relayed from the source to the destination node. Actually, we make use of the diffusion property of the encryption scheme, i.e., encrypting a message $\mathcal{M}$ with different keys produces different ciphetexts, e.g., although the ciphertexts $E_{KA}(\mathcal{M})$ and $E_{KB}(\mathcal{M})$ are for the same message, they look completely different. Moreover, with using secure symmetric-key cryptosystem such as AES [20], it is computationally infeasible to correlate the ciphertexts $E_{KA}(\mathcal{M})$ and $E_{KB}(\mathcal{M})$ without knowing the secret keys K_A and K_B.

b) Packet-length correlation

The packets of a flow can be correlated if they have distinguishable length. One of the following two techniques can be used to prevent this correlation: (1) *fixed-length packets*: all packets have the same length and random padding is appended if a packet's length is short; or (2) *random-length packets*: a random-length padding is added by a node and replaced by the next node so that a packet's length is variable at each hop.

c) *Packet-transmission-time correlation*

Attackers may try to correlate a packet as it is relayed by observing the transmission time at a node and its neighbors. The attackers make use of the fact that the nodes usually relay packets after a short processing delay and based on first-received-first-relayed basis. Changing the packets' appearance at each hop cannot prevent this correlation because it depends on the packets' sending time and not the content. A common approach to obfuscate the temporal relationship between the incoming and outgoing packets is to use mixing technique. A mixer buffers a sequence of incoming packets and shuffles them before transmission such that correlating the incoming and outgoing packets is difficult. It can also add dummy packets to the buffer if necessary. The base stations and some mobile nodes can act as mixers.

We use information-theoretic metric, called entropy [21], to quantify the privacy protection provided by mixers. The entropy of the probability that an attacker can correlate an ingoing packet of interest with the corresponding outgoing packet is given in Eq. 1. P_i is the probability assigned by the attacker for the outgoing packet number i to be the corresponding for the ingoing packet of interest. $\sum_{i=1}^{n_b-n_k} P_i = 1$, where n_b and n_k are the buffer size of the mixer and the number of ingoing packets the attacker sent to ease correlating packets, respectively. If the attacker can know that n_k packets are uncorrelated to the packet of interest, he can shrink the anonymity set from n_b to n_b - n_k. The maximum entropy (or the maximum privacy protection) can be achieved when the probabilities P_i (for $1 \le i \le n_b$ - n_k) pursue uniform distribution or $P_i = 1/(n_b - n_k)$. In this case, the attacker believes that all the outgoing packets have the same probability to be the correspondent of the packet of interest, and thus the input packet is perfectly hidden in the buffer's packets. The maximum entropy (H'_{max}) is given in Eq. 2, and the anonymity degree (D) is given in Eq. 3.

$$H(X) = -\sum_{i=1}^{n_b-n_k} P_i \cdot log_2(P_i) \quad (1)$$

$$H'_{max} = -\sum_{i=1}^{n_b-n_k} \frac{1}{n_b-n_k} \cdot log_2\left(\frac{1}{n_b-n_k}\right) = log_2(n_b-n_k) \quad (2)$$

$$D = 1-\frac{H_{max} - H'_{max}}{H_{max}} = \frac{H'_{max}}{H_{max}} = \frac{log_2(n_b-n_k)}{log_2 n_b} \quad (3)$$

$$Where:\ H_{max} = -\sum_{i=1}^{n_b} \frac{1}{n_b} \cdot log_2\left(\frac{1}{n_b}\right) = log_2(n_b)$$

Fig. 8 shows the degree of anonymity versus n_b at different values of n_k. It can be seen that the increase of n_b increases the degree of anonymity. For $n_k = 5$, increasing n_b above 20 has little impact on the degree of anonymity, but certainly increases the packet relaying delay. It can also be seen that the increase of n_k decreases the degree of anonymity for the same buffer size, however, this can be alleviated by increasing the buffer size.

Privacy is defined as the protection of data from unauthorized parties. While encryption can protect the content of the messages, traffic analysis may reveal valuable information about the users' relationships, communication activities, and locations.

Location privacy is defined as the ability to prevent attackers from deducing a user's current or past locations whether the exact physical locations or the relative locations in number of hops. Attackers should not be able to deduce the distance to either the anonymous source or destination node in number of hops, e.g., by analyzing the packets' length or content. In our protocol, the nodes' exact locations are not used, and the length and content of the route request packets do not reveal the location of the source nodes due to using random-length padding and random-value *TTL*. This can confuse the source nodes' neighbors whether the packets are originated from or relayed by them. The nodes also relay the packets destined to them to protect their location privacy. Moreover, the source and destination nodes cannot know the locations of each other even if they are one-hop away. They also cannot know whether they are in the same cell or not. To inform a destination node about a call with preserving its location and identity privacy, a trapdoor that only the destination node can recognize is used. In our protocol, the trapdoor is a fresh pseudonym shared between the base station and the destination node.

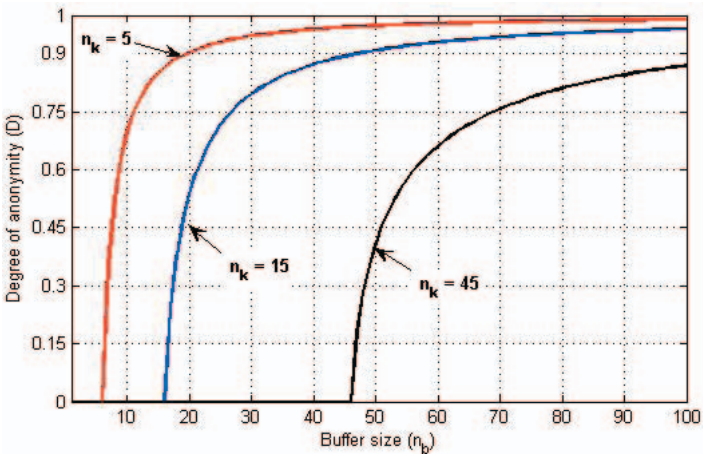


Fig. 8: The degree of anonymity versus n_b.

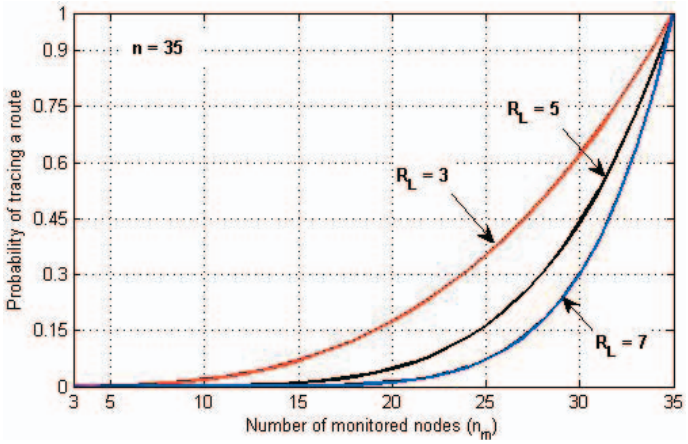


Fig. 9: The probability of tracing a route versus n_m.

Unlinkability of two or more items within a defined system means that these items are no more and no less related than they are related concerning the apriori knowledge. For *Source-destination pair unlinkability*, although attackers may know that a pair of nodes participates in communication activity, they cannot ensure that the pair communicates with each other. In our protocol, every time a source and destination pair communicates, the route discovery packets look different, so linking a packet to a source-destination pair is infeasible. Moreover, if an attacker eavesdrops on the source and destination nodes and their base stations, he cannot make sure that they currently communicate. For *source node and base station unlinkability*, if an adversary eavesdrops on a source node and its base station, linking the packets is infeasible. The packets exchanged between a source node and a base station pair at different times/sessions are uncorrelated because pseudonyms are changeable and unlinkable. This means that even if an adversary could correlate the pair in one occasion, he cannot benefit from this conclusion in the future.

For *a transmission and source node unlinkability*, an adversary cannot link a transmission to its source node because the packets sent in different times have no common information or any information that can be linked to a real identity. Moreover, identifying the source or the destination node does not necessarily lead to identifying the other party.

Anonymity of a subject means that the subject is not uniquely characterized within a set of subjects that is called the anonymity set. *Identity anonymity* means that the real identity of a node that participates in a route either as source, intermediate or destination cannot be identified by attackers. Identity anonymity enables users to avoid being identified by the locations they visit or the traffic they generate. In our protocol, the real identity is always kept confidential and never disclosed. The base stations and *Tp* need to know the nodes' identities for routing, payment clearance, and enforcing accountability and access control, but no one else can infer the nodes' real identities. The source and destination nodes cannot know the real identities of the intermediate nodes, and an intermediate node cannot know the real identities of the other intermediate nodes or the source and destination nodes. *Sender anonymity* means that an adversary cannot identify the source node in a particular communication session. In our protocol, a particular transmission is unlinkable to a source node, and any transmission is unlinkable to a particular source node. Similarly, *recipient anonymity* means that an adversary cannot identify the destination node of a particular session.

Route anonymity means that attackers cannot infer the nodes participating in the route. This property is important to make identifying the anonymous source/destination nodes' locations difficult. In our protocol, adversaries cannot correlate packets at different hops nor infer the route by analyzing the route discovery packets. An intermediate node can know that the packets it relays belong to one session for correct routing, but eavesdroppers are not able to correlate the packets to a session or identify the nodes participating in one route. For *Neighbor anonymity*, a long-time relation between a node and its neighbors enables the attackers to collect much information that will severely violate the neighbors' anonymity if the attackers could link a neighbor to a user. In our protocol, each node interacts with anonymous neighbors only during the session time. After the session ends, the nodes use new pseudonyms and the attacker cannot know if the nodes in neighborhood are the same or different.

In *packet-flow tracing* attack, the attackers try to infer a route by tracing packets backward/forward to the source/destination node. Unlike reference [10] where each node uses one pseudonym for all the packets of a session, our protocol can use one pseudonym per packet. Eavesdroppers cannot link a session's packets at one node or link a packet at different intermediate nodes of a route. In the protocols that do not use per-hop encryption/decryption operations, such as ANODR [9], if an eavesdropper captures a packet at different intermediate nodes of a route, he can correlate the packets.

Eq. 4 gives the probability (*Pr*) that an eavesdropper can trace a route in ANODR, where R_L is the number of nodes in the route including the source and destination nodes, n denotes the total number of nodes in the network, and n_m denotes the number of nodes that the attacker can overhear their transmissions. The probability of overhearing a node's transmission and the probability of participating in a session are uniformly distributed. Fig. 9 shows that the route tracing probability increases when the attacker can overhear the transmissions of more nodes, and it is more probable to trace the shorter routes than the longer ones.

$$P_r = \frac{n_m! \times (n-R_L)!}{(n_m-R_L)! \times n!} \qquad (4)$$

For *movement tracking*, attackers attempt to track a user's movement, e.g., by linking a device's transmissions to a user. The adversary can distribute a group of devices to monitor the transmissions in the places that the victim user may visit. Even if the users' real identities are hidden, the attackers should not be able to track the anonymous users to prevent them from gaining much information about the users' past visited locations if the attackers could identify the anonymous users. Moreover, these collected information can be used to identify the users, e.g., if an anonymous user is always located at one location at night, the attackers can infer that this location is the anonymous user's house. In our protocol, the real identity anonymity and the changeable and unlinkable pseudonyms can prevent tracking users. For *anonymous yet secure authentication*, in our protocol, the base stations mutually authenticate neighboring nodes in a route without disclosing their real identities.

In *fingerprint recording* attack, the attackers record a list of plaintexts and the corresponding ciphertexts computed by a node so that they are used as a fingerprint for the node. Each time the attackers observe a plaintext-ciphertext pair, they can identify the node. In our protocol, the session keys are used only for one session and all the packets have a variable part (time stamp or a fresh pseudonym) so that the same plaintext-ciphertext pair cannot be produced at different occasions. Thus, even if an attacker could link a plaintext-ciphertext pair to a node in one occasion, he can-

not benefit from this conclusion in the future. We call this property *forward privacy-preservation*, i.e., if an attacker could violate a user's privacy in one occasion, this privacy violation should not help violate the user's privacy in the future.

For *pseudonym unlinkability,* attackers should not be able to link the pseudonyms of one chain. This property is important because if an adversary could link a pseudonym to a node in one occasion, he will not benefit from this conclusion in the future. In our protocol, the unintended nodes cannot correlate the pseudonyms of a chain because they do not know the secret key used in generating them, i.e., given X and $H_K(X)$, it is computationally infeasible to know that $H_K(X)$ is resulted from hashing X if the secret key K is unknown.

Each node should not store the initial random value (R) that is used in computing pseudonyms, but store only the last pseudonym. This is because if the node is compromised and its key is revealed, the adversary cannot compute the past pseudonyms thanks to the unidirectionality property of the hash functions, i.e., it is infeasible to compute $H_{K_{SBs}}^{(i-1)}(R)$ from $H_{K_{SBs}}^{(i)}(R)$ even if R and K_{SBs} are known. If an attacker attaches a random value for a pseudonym, the probability to hit the correct value is extremely low, e.g., this probability is 6.84×10^{-49} by using SHA-1 with digest value of 20 bytes.

Pseudonym collision means that more than one node have identical pseudonyms because the hash function may generate the same hash value from hashing different inputs. Pseudonym collision may result in losing pseudonym synchronization, or forwarding packets to a wrong direction because pseudonyms are used as routes' identifiers. Using birthday paradox [20], the pseudonym collision probability is $2^{-K/2}$, where K is the number of bits of a pseudonym. For example, if $K =$ 64 bits, the pseudonym collision probability is 2.3×10^{-10}, which implies one pseudonym collision every $4.2\times10^{+9}$ pseudonyms. As studied in [22], the probability of pseudonym collision is given in Eq. 5 when m pseudonyms are selected and L is the pseudonym length in bits.

$$Pr_{Collision} = 1 - \frac{\prod_{i=0}^{m-1}(2^L - i)}{(2^L)^m} \quad (5)$$

Fig. 10 shows that the linear increase of L decreases the pseudonym collision probability exponentially and the increase of m can increase the collision probability. It can be seen that the collision probability can be negligible. In our pseudonym generation technique, a collision can be resolved automatically because the nodes use different keys in computing pseudonyms, i.e., if multiple nodes have the same pseudonym value at the same time, the next pseudonyms will be different because of using different keys in computing them. If a node loses pseudonym synchronization with the base station, the node can resynchronize by initiating a new authentication process. To reduce the packet overhead, pseudonyms can be truncated to shorter bit string without significantly increasing the probability of pseudonym collision as shown in Fig. 10. The truncation level depends on the size of the base station control area and the number of nodes.

In *fake route discovery attack,* the attackers initiate route discovery packets with the intention of collecting information about the nodes in the network. The attacker may attach a particular content, e.g., layered ciphertext, and then observes the processed packets (the encryption of the ciphertext) by neighbors to know if a victim node is in its neighborhood. In our protocol, the route discovery packets have a timestamp to make the packets sent by neighbors at different occasions look different.

A node may use a pseudonym shared with the base station for some time if it does not participate in a route which enables it to change pseudonym. This may be specifically applicable to the nodes at the network border because they are less frequently selected by the routing protocol. The attackers may make use of this fact by initiating fake *URREQ* packets and analyzing the packets sent by neighbors to learn whether a user is still in the neighborhood. As we discussed earlier, each node can change its pseudonym within a window of pseudonyms before receiving a fresh pseudonym from the base station without losing synchronization.

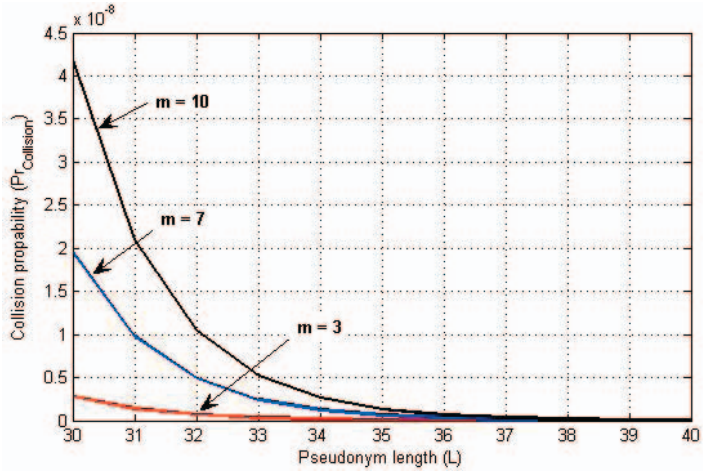


Fig. 10: The collision probability versus pseudonym length.

Table 1: The cryptographic operations required by our protocol.

	Route discovery	Data packet	*ACK*
$\mathcal{N}_S$	$2h, e, d$	$2h, e$	h, d
Uplink nodes	$2h, e, d$	h, e	h, d
Bs	$2\alpha h, \alpha e, \alpha d$	$2h, \alpha d$	$h, \alpha e$
Bd	$(2\beta+1)h, (\beta+1)e, \beta d$	$2h, \beta e$	$h, \beta d$
Downlink node	$3h, e, d$	h, d	h, e
$\mathcal{N}_D$	$3h, e, 2d$	$2h, d$	h, e

Table 2: Simulation results.

		RREQ	*DNOT*	*REST*	Data packet
Delay (ms)	**Min**	Route establishment		97.61	41.68
	Avg.			101.32	42.76
	Max			105.03	43.84
Avg. packet length (bytes)	**Min**	70.585	91.88	161.76	534.3
	Avg.	73.68	95.31	170.27	548
	Max	76.775	98.74	178.78	561.7

Moreover, the nodes can establish routes with the base stations to update their pseudonym window. To do this, the proposed protocol for establishing uplink routes explained in Fig. 5 can be used, and the padding can be a pre-defined value to inform the base station that the packet is for updating the pseudonym window and not for communication call. The base station replies with *UREST* packet with a fresh pseudonym.

In *pseudonym de-synchronization attack,* the attackers try to damage the pseudonym synchronization between a node and the base station. A source node broadcasts an *URREQ* packet containing a new pseudonym $H_{K_{SBs}}^{(i)}(R)$ and waits for $H_{K_{SBs}}^{(i+1)}(R)$ from the base station. If the packet does not reach the base station, the base station is still using $H_{K_{SBs}}^{(i-1)}(R)$ because it has not received $H_{K_{SBs}}^{(i)}(R)$. The node and the base station lose synchronization if the node matches the base station's pseudonym only to $H_{K_{SBs}}^{(i+1)}(R)$. In Subsection 4.1, we have discussed that the loss of pseudonym synchronization is unlikely because the node will not release $H_{K_{SBs}}^{(i+2)}(R)$ before receiving $H_{K_{SBs}}^{(i+1)}(R)$, and the base station will not release $H_{K_{SBs}}^{(i+3)}(R)$ before receiving $H_{K_{SBs}}^{(i+2)}(R)$.

6. PERFORMANCE EVALUATION

To measure the computational times of the cryptographic operations required for our protocol, we have implemented AES (128 bit key) symmetric key cryptosystem and SHA-1 (160 bit) hash function using the Crypto++5 [23] library and 1.6 GHZ processor. According to NIST [24], the secure key size should be at least 128 bits. The measurement results indicate that a hashing operation requires 16.79 Mbytes/s and encryption/decryption operations require 9.66 Mbytes/s. For the energy consumption, the measurements given in [25] indicate that a hashing operation and an encryption or decryption operation require 0.76 μJ/byte and 1.21 μJ/byte, respectively. These results confirm that hashing and symmetric-key operations require low overhead.

Table 1 gives the cryptographic operations required by our protocol. *h*, *e*, and *d* refer to a hashing, an encryption, and a decryption operation, respectively. α and β are the numbers of the uplink and downlink nodes including the source and the destination nodes, respectively. The results indicate that our protocol can assign more overhead to the base stations and it can balance the overhead on the mobile nodes. The base stations have more computational power and energy than the nodes do.

Using NS2, we simulate a hybrid ad hoc network by randomly deploying 45 mobile nodes in a square cell of 1200 m × 1200 m. A fixed base station is located at the center of the cell. The radio transmission range of the mobile nodes and the base station is 125 m. To emulate node mobility, we adopt the modified random waypoint model [26]. Specifically, a node travels towards a random destination uniformly selected within the network field; upon reaching the destination, it pauses for some time; and the process repeats itself afterwards. The node speed is uniformly distributed in the range [0, 2] m/s and the pause time is 3s. The constant bit rate traffic source is implemented in each node as an application layer. The source and destination pairs are randomly selected. Packets are sent at the rate of 2 packets/s. The number of concurrent connections is 7.

The cryptographic operations are simulated by adding their computational times to the packets processing time. Our simulation is executed for 15 minutes and the given results are averaged over 100 simulation runs and presented with 95% confidence interval. The length of truncated pseudonyms (δ), *Pad*, time stamp, real identity, and payload (M_C) are ten, 2δ, five, four, 512 bytes, respectively. With these parameters, the network connectivity is 0.96. The connectivity is measured by the number of established routes to the number of route requests sent by the source nodes.

The simulation results are given in Table 2. Route establishment delay is the average time interval between sending an *URREQ* packet by a source node and receiving the *UREST* packet. The data packet delay is the average time interval between sending a data packet by a source node and receiving it by the destination node. These delays include: processing delays at each node, queuing delay at the interface queue, retransmission delays, and propagation time. The simulation results indicate that the expected route establishment and data transmission delays are acceptable due to using lightweight cryptographic operations and precomputing the pseudonyms. For the *RREQ* and *REST* packets, the packet length varies at each node as the packet is relayed, so the average is computed by dividing the amount of data relayed at all hops by the number of hops. The results also indicate that the overhead of the data packets is 36 bytes that constitute 7% of the message size (512 byes). *REST* packet is large because it carries the nodes' session keys, but being unicated packet and reducing the packet size at each hop can alleviate this. The packet delivery ratio is the number of data packets received by the destination nodes to those sent by the source nodes. Our simulation results indicate that the average packet delivery ratio is 0.95.

7. CONCLUSION

We have proposed a lightweight secure and privacy-preserving protocol for hybrid ad hoc wireless network. Short-life pseudonyms, one-time session keys, and per-hop encryption/decryption operations are used to preserve users' privacy. Cryptographic operations and payment system are used to secure the communication. To reduce the overhead, lightweight cryptographic operations are used, efficient trapdoor technique is developed, and the payment can be secured without storing, submitting, or processing receipts. In addition, our pseudonym generation technique requires only lightweight hashing operations and does not require large storage area or frequently refilling pseudonyms from a trusted party. The pseudonyms are authenticated and can be precomputed which can reduce the packet delay. Our evaluations and simulation results demonstrate that the proposed protocol can preserve the nodes' privacy with low overhead and secure the payment, route establishment, and data transmission.

REFERENCES

[1] M. Mahmoud and X. Shen, "FESCIM: Fair, efficient, and secure cooperation incentive mechanism for hybrid ad hoc networks", IEEE Transactions on Mobile Computing, vol. 11, no. 5, pp. 753-766, 2012.

[2] M. Mahmoud and X. Shen, "Lightweight privacy-preserving routing and incentive protocol for hybrid ad hoc wireless networks", Proc. of IEEE INFOCOM'11- International Workshop on Security in Computers, Networking and Communications (SCNC), Shanghi, April 10-15, 2011.

[3] M. Mahmoud and X. Shen, "Anonymous and authenticated routing in multi-hop cellular networks", Proc. of IEEE International Conference on Communications (IEEE ICC'09), pp. 839-844, Dresden, Germany, June 14-18, 2009.

[4] N. Salem, L. Buttyan, J. Hubaux, and M. Jakobsson, "Node cooperation in hybrid ad hoc networks", IEEE Transactions on Mobile Computing, vol. 5, no. 4, pp. 365 – 376, April 2006.

[5] M. Mahmoud and X. Shen, "ESIP: Secure incentive protocol with limited use of public-key cryptography for multi-hop wireless networks", IEEE Transactions on Mobile Computing, vol. 10, no. 7, pp. 997-1010, July 2011.

[6] M. Mahmoud and X. Shen, "PIS: A practical incentive system for multi-hop wireless networks", IEEE Transaction on Vehicular Technology, vol. 59, no. 8, pp. 4012-4025, 2010.

[7] B. Lamparter, K. Paul, and D. Westhoff, "Charging support for ad hoc stub networks", Computer Communications, vol.26, no. 13, pp. 1504–1514, 2003.

[8] S. Capkun, J. P. Hubaux, and M. Jakobsson, "Secure and privacy-preserving communication in hybrid ad hoc networks", Technical Report IC/2004/10, EPFL-DI-ICA, 2004.

[9] J. Kong, X. Hong, and M. Gerla, "An identity-free and on-demand routing scheme against anonymity threats in mobile ad hoc networks", IEEE Transactions on Mobile Computing, vol. 6, no. 8, pp. 888–902, 2007.

[10] A. Boukerche, K. El-Khatib, L. Korba, and L. Xu, "A secure distributed anonymous routing protocol for ad hoc wireless networks", Journal of Computer Communications, NRC 47393, 2004.

[11] K. El-Defrawy and G. Tsudik, "Privacy-preserving location-based on-demand routing in MANETs", IEEE Journal on Selected Areas in Communications, vol. 29, no. 10, pp. 1-10, December 2011.

[12] X. Lin, X. Sun, P. Ho, and X. Shen, "GSIS: A secure and privacy-preserving protocol for vehicular communications", IEEE Transactions on Vehicular Technology, vol. 56, no. 6, pp. 3442-3456, 2007

[13] K. Ren, S. Yu, W. Lou, Y Zhang, "PEACE: a novel privacy-enhanced yet accountable security framework for metropolitan wireless mesh networks", IEEE Transactions on Parallel and Distributed Systems, vol. 21, issue 2, pp. 203-215, 2010.

[14] K. Ren and W. Lou, "A sophisticated privacy-enhanced yet accountable security framework for wireless mesh networks", Proc. of IEEE ICDCS, Beijing, China, June 17-20, 2008.

[15] M. Mahmoud and X. Shen, "Cloud-based scheme for protecting source location privacy against hotspot-locating attack in wireless sensor networks", IEEE Transactions on Parallel and Distributed Systems (IEEE TPDS), vol. 23, no. 10, pp. 1805-1818, 2012.

[16] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Securing mobile ad hoc networks with certificateless public keys", IEEE Transactions on Dependable and Secure Computing, vol. 3, no. 4, pp. 386 – 399, 2006.

[17] S. Yi, P. Naldurg, and R. Kravets, "Security-aware ad hoc routing for wireless networks", Proc. 2nd ACM Symp. Mobile Ad Hoc Net. and Comp. (Mobihoc'01), Long Beach, CA, pp. 299–302, October, 2001.

[18] Y. -C. Hu, A. Perrig, and D. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks", Proc. ACM Conference Mobile Computing and Networking (MobiCom), 2002.

[19] G. Acs, L. Buttyan, and I. Vajda, "Provably secure on-demand source routing in mobile ad hoc networks", IEEE Transactions on Mobile Computing, vol. 5, no. 11, pp. 1533 – 1546, November 2006.

[20] A. J. Menezes, P. van Oorschot, and S. Vanstone. Handbook of Applied Cryptography. CRC Press, 1996.

[21] C. Dıaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity", Proc. of Privacy Enhancing Technologies Workshop (PET'02), R. Dingledine and P. Syverson, Eds. Springer-Verlag, LNCS 2482, April. 2002.

[22] J. Kong and X. Hong, "Anodr: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks", Proc. of MobiHoc, 2003.

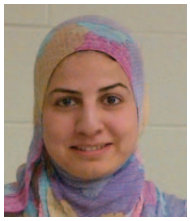
[23] W. Dai, "Crypto++ Library 5.6.0", http://www.cryptopp.com.

[24] National Institute of Standards and Technology (NIST), "Recommendation for key management - Part 1: General (Revised)", Special Publication 800-57 200, 2007.

[25] N. Potlapally, S. Ravi, A. Raghunathan, and N. Jha, "A study of the energy consumption characteristics of cryptographic algorithms and security protocols", IEEE Transactions on Mobile Computing, vol 5(2), pp. 128-143, 2006.

[26] J. Yoon, M. Liu, and B. Nobles, "Sound mobility models", Proc. of ACM MobiCom, San Diego, CA, USA, September 2003.

Mohamed M. E. A. Mahmoud received PhD degree from the University of Waterloo in April 2011. From May 2011 to May 2012, he worked as a postdoctoral fellow in the Broadband Communications Research group - University of Waterloo. From August 2012 to July 2013, he worked as a visiting scholar in University of Waterloo, and a postdoctoral fellow in Ryerson University. Currently, Dr Mahmoud is an assistant professor in Department Electrical and Computer Engineering, Tennessee Tech University, USA. The research interests of Dr. Mahmoud include security and privacy preserving schemes for smart grid communication network, mobile ad hoc network, sensor network, and delay-tolerant network. Dr. Mahmoud has received MITACS-PDF award, Canadian national award. He has also received the competitive NSERC-PDF award, Canadian national award. He won the prestigious Best Paper Award from IEEE International Conference on Communications (ICC'09), Dresden, Germany, 2009. Dr. Mahmoud is the first author for more than twenty three papers published in major IEEE conferences and journals, such as INFOCOM conference and IEEE Transactions on Vehicular Technology, Mobile Computing, and Parallel and Distributed Systems. He serves as an Associate Editor in Springer journal of peer-to-peer networking and applications. He served as a technical program committee member for several IEEE conferences and as a reviewer for several journals and conferences such as IEEE Transactions on Vehicular Technology, IEEE Transactions on Parallel and Distributed Systems, and the journal of Peer-to-Peer Networking.



Sanaa Taha received her B.Sc. (2001) and M.Sc. (2005) degrees from the Department of Information Technology, Faculty of Computers and Information, Cairo University, Egypt, and a Ph.D. degree (2013) in Electrical and Computer Engineering from the University of Waterloo, Canada. She is currently an assistant professor in the Department of Information Technology, Faculty of Computers and Information, Cairo University, Cairo, Egypt. Her research interests include wireless network security, mobile networks security, mobility management, and applied cryptography.

Jelena Misic (M91, SM08) Professor of Computer Science at Ryerson University in Toronto, Ontario, Canada. She has published over 90 papers in archival journals and more than 120 papers at international conferences in the areas of wireless networks, in particular wireless personal area network and wireless sensor network protocols, performance evaluation, and security. She serves on editorial boards of *IEEE Transactions on Vehicular Technology*, *IEEE Network*, *IEEE JSAC series on Smart Grid Communications*, *Computer Networks*, *Ad hoc Networks*, and *Security and Communication Networks*.

Xuemin (Sherman) Shen (IEEE M'97-SM'02-F09) received the B.Sc.(1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He was the Associate Chair for Graduate Studies from 2004 to 2008. Dr. Shen's research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He is a co-author/editor of six books, and has published more than 600 papers and book chapters in wireless communications and networks. Dr. Shen served as the Technical Program Committee Chair/Co-Chair for IEEE Infocom'14, IEEE VTC'10 Fall, and IEEE Globecom'07, the Symposia Chair for IEEE ICC'10, the Tutorial Chair for IEEE VTC'11 Spring and IEEE ICC'08, the General Co-Chair for Chinacom'07 and QShine'06, the Chair for IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He also serves/served as the Editor-in-Chief for IEEE Network, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor for IEEE Transactions on Wireless Communications; an Associate Editor for IEEE Transactions on Vehicular Technology, Computer Networks, and ACM/Wireless Networks, etc.; and the Guest Editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007 and 2010 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.