Lightweight Privacy-Preserving Routing and Incentive Protocol for Hybrid Ad Hoc Wireless Network

Mohamed Elsalih Mahmoud and Xuemin (Sherman) Shen
Department of Electrical and Computer Engineering, University of Waterloo, Canada
Emails: {mmabdels, xshen}@bbcr.uwaterloo.ca

Abstract—**We propose a privacy-preserving routing and incentive protocol, called PRIPO, for hybrid ad hoc wireless network. PRIPO uses micropayment to stimulate node cooperation without submitting payment receipts. The lightweight hashing and symmetric-key-cryptography operations are implemented to preserve the users' privacy. The nodes' pseudonyms are efficiently computed using hashing operations. Only a trusted party can link these pseudonyms to the real identities for charging and rewarding operations. Moreover, PRIPO protects the location privacy of the anonymous source and destination nodes. Extensive analysis and simulations demonstrate that PRIPO can secure the payment and preserve the users' privacy with acceptable overhead.**

Index Terms—Security, privacy-preserving routing, cooperation stimulation.

I. INTRODUCTION

In hybrid ad hoc wireless network, the mobile nodes usually act as routers to relay others' traffics for enhancing the network performance and deployment [1]. Multi-hop packet relay extends the base stations' coverage area without additional cost. It also enhances the network throughput and capacity due to reducing the transmission interference area by transmitting the packets over shorter hops. However, the nature of the wireless transmission and multi-hop packet relay makes the network highly vulnerable to serious security challenges.

Although the proper network operation requires the nodes' cooperation in relaying others' packets, the selfish nodes will not cooperate without sufficient incentive to save their resources. This behavior significantly degrades the network connectivity and packet delivery ratio and may result in failure of the multi-hop communication [2]. Moreover, the attackers can analyze the network traffic to learn the users' locations in number of hops and their communication activities causing a severe threat for the users' privacy. These attacks can be launched in undetectable way because the attackers just overhear the transmissions without disturbing the communication.

Incentive schemes [3-6] use credits (or micropayment) to stimulate node cooperation. The source and the destination nodes pay credits to the intermediate nodes for relaying their packets. These schemes can also enforce fairness because credits compensate the nodes for consuming their resources in packet relay. Since the users pay for transmitting their messages, incentive schemes discourage launching *Resource-Exhaustion* attack by sending bogus messages to exhaust the intermediate nodes' resources. However, each node has to use a unique identity in the existing schemes for charging and rewarding operations, which jeopardizes the users' privacy. They also incur much overhead due to using public key cryptography and submitting receipts (proofs of packet relay) to a trusted party (Tp). Moreover, the existing privacy-preserving routing protocols [7-9] heavily depend on packet broadcasting and public key cryptography, which makes these protocols infeasible for hybrid ad hoc networks due to the constraints on the nodes' resources.

In this paper, we propose **PRIPO**, a **P**rivacy-Preserving **R**outing and **I**ncentive **P**r**O**tocol for hybrid ad hoc wireless network. PRIPO can foster node cooperation and preserve the privacy of the users' locations and communication activities using lightweight hashing and symmetric-key-cryptography operations and without submitting receipts. The nodes' pseudonyms are efficiently computed using hashing operations. Only trusted parties can link these pseudonyms to the real identities for charging and rewarding operations. Extensive evaluations and simulations demonstrate that PRIPO can secure the payment and preserve the users' privacy with acceptable overhead. To the best of our knowledge, PRIPO is the first protocol that addresses both cooperation stimulation and user privacy for hybrid ad hoc networks.

The remainder of this paper is organized as follows. We review the related work in Section II. In Section III, we present the network and threat models. We propose PRIPO in Section IV. Security analysis and performance evaluation are given in Sections V and VI, respectively, followed by conclusion in Section VII.

II. RELATED WORK

A. *Incentive Schemes*

In Sprite [3], the source node signs each message and the identities of the nodes in the route and appends the signature to the data packet. The intermediate and the destination nodes compose a receipt per message and submit the receipts to Tp for clearance. An intermediate node is rewarded for relaying a message if a next node in the route submits the receipt. However, Sprite charges only the source node no matter how the destination node benefits from the communication. The receipts overwhelm the network because of generating and submitting a large number of receipts, which consumes the nodes' storage and energy and the network bandwidth, and requires a massive processing overhead to clear the receipts.

In PIS [4], the source node attaches a signature to each message and the destination node replies with a signed ACK. PIS can reduce the receipts' overhead and charge the source and the destination nodes when both of them benefit from the communication. A fixed-size receipt is generated per session regardless of the messages' number, and only one node has to submit the session receipt instead of submitting it by all the intermediate nodes. However, the extensive use of the public key cryptography operations is too costly for mobile nodes.

In CDS [5], instead of submitting payment receipts, each node submits a smaller-size activity report containing its alleged charges and rewards for different sessions. Tp uses statistical tools to identify the cheating nodes by measuring how frequently the nodes' reports are inconsistent with others.

However, due to the nature of the statistical tools, some honest nodes may be falsely identified as cheaters and colluding nodes may manage to steal credits. In [6], Salem et al. have proposed an incentive scheme for hybrid ad hoc network, but the nodes have to submit receipts to the base station when a route is broken to secure the payment. Unlike this work, PRIPO can preserve the users' privacy and eliminate the need for submitting receipts.

B. Privacy-preserving Routing Protocols

In [7], Capkun et al. have proposed a privacy-preserving communication protocol for hybrid ad hoc network. Each node stores a set of public/private key pairs and certificates. The nodes have different pseudonyms that are certified by Tp. The node uses its public/private key pairs to establish symmetric keys shared with its neighbors. It frequently changes its pseudonym by changing the public/private key pair and establishing new symmetric keys with its neighbors. The authors demonstrate that the sufficient frequency of pseudonym change is in the order of 1/min. The data is encrypted using the base station's public key so that the intermediate nodes relaying the data to the base station cannot interpret the content. However, the nodes periodically contact Tp to refill their public/private key pairs. Generating and distributing a large number of public/private keys with certificates is very costly. Since the network nodes have a large number of certificates, certificate revocation is a real challenge.

In ANODR [8], the source node attaches a trapdoor to the *Route Request Packet* (*RREQ*) to anonymously inform the destination node about the session. The trapdoor contains the destination node's real identity and a random value encrypted by a shared key with the destination. Each node X tries to open the trapdoor, and if it is not the destination, X adds a nonce N_X and encrypts the packet with onetime key K_X creating onion message encrypted by all the intermediate nodes along the route. The destination node adds the onion message to the *Route Reply Packet* (*RREP*) and broadcasts the packet. The nodes discard the packet if they cannot open the onion message using K_X and N_X, otherwise, they are intermediate nodes in the route. However, the trapdoor used in the *RREQ* packet is not practical or scalable because each node has to decrypt the trapdoor with every key it shares with other nodes. This is because the identities of the source and the destination nodes are hidden for anonymity. The source and the destination nodes cannot establish session keys shared with the intermediate nodes to make cryptographic onion for the communication data, and thus, packet un-likability is unachievable. The processing overhead of the *RREQ* and *RREP* packets is not negligible because they are broadcasted.

In SDAR [9], the source node attaches a onetime public key and a trapdoor to the *RREQ* packet. The trapdoor contains the destination node's identity and a onetime session key encrypted by the public key of the destination node. Each node tries to open the trapdoor with its private key and if it is not the destination, the node adds a nonce as a pseudonym, a session key, and onetime public key. The destination node broadcasts the *RREP* packet which contains the pseudonym of the next node in the route and an onion message. Each intermediate node decrypts one layer of the message using the session key and broadcasts the packet that contains the pseudonym of the next node in the route. The source and destination nodes create a cryptographic onion for their communication data using the session keys they share with the intermediate nodes. However, the protocol is not efficient as it require every node to perform a private key decryption, a public key encryption and a signature generation for every RREQ message it forwards. The sizes of the *RREQ* and the *RREP* packets are large consuming much energy and bandwidth. Moreover, the destination node learns the identities of all the nodes in route and the location of the destination node is disclosed to the source node. In the *RREQ* packet, an intermediate node can delete the last part of the routing message that was attached by its previous nodes and make a new routing message.

III. SYSTEM MODEL

A. Network Model

The considered hybrid ad hoc wireless network consists of mobile nodes, a set of base stations, and Tp. The base stations are connected with each other and with Tp by a backbone network. A mobile node X should register with Tp to get a permanent shared symmetric key K_X and a unique identity ID_X. Tp manages the nodes' credit accounts and maintains their keys. The source node (S) sends its packets to the source base station (Bs), if necessary in multiple hops. Bs forwards the packets to the destination base station (Bd) if the destination node (D) resides in a different cell, and finally, the packets are sent to D, possibly in multiple hops again. The part of the route between S and Bs is called uplink, and the part of the route between Bd and D is called downlink.

Our payment model supports a cost sharing between the source and the destination nodes when both of them benefit from 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 the destination nodes are charged and the uplink intermediate nodes are rewarded only for the messages received by Bs even if they do not reach to D. The downlink intermediate nodes are rewarded only when Bd receives *Acknowledgement* packet (*ACK*) from D. In Section V-A, we will argue that this rewarding and charging policies can discourage cheating actions and encourage cooperation without submitting payment receipts. PRIPO uses fixed rewarding rate, e.g., λ credits per unit-sized message. The nodes at the network border cannot earn as many credits as the nodes at other locations because they are less frequently selected by the routing protocol. To enable these nodes to communicate, the nodes can purchase credits for real money.

B. Threat and Trust Models

The attackers have full control on their nodes and thus they can change the nodes' operations. The attackers work individually or collude with each other to launch sophisticated attacks. Specifically, the attackers attempt to steal credits, pay less, and communicate freely. Legitimate nodes or eavesdroppers may attempt to learn the nodes' real identities and locate individual nodes in number of hops and track their movements. The attackers also aim to launch traffic analysis attacks to monitor the communication activities of the nodes. The mobile nodes are probable attackers because they are motivated to misbehave to increase their welfare. However, Tp and the base stations are secure because they are operated by a single operator that is motivated to ensure the network security. The nodes' real identities and locations are known to the base stations and Tp in order to route the messages accordingly and for charging and rewarding operations. Nevertheless, the nodes' long-term keys are known only to Tp.

We do not consider the global eavesdropper that can monitor every radio transmission on every communication link in the network at all time. This is because these attacks are too complicated to occur in civilian applications and scalable networks, and the countermeasures usually require much overhead. In PRIPO, the global eavesdroppers may locate the source and destination nodes and identify the route if there is only one active session in the network, but they cannot link the nodes' pseudonyms to the real identities. For the trust models, the nodes trust Tp and the base stations with performing billing and auditing correctly and with preserving their location and identity privacy, but they do not trust the mobile nodes.

IV. The Proposed PRIPO

A. Pseudonyms and Shared Keys

To protect a node's identity privacy, the node uses pseudonyms such that only an intended node can link the pseudonyms to each other and to the real identity. In this way, even if an attacker could link a pseudonym to a node, he cannot violate the node's privacy for a long time. As shown in Fig. 1, if the nodes W and X share a secret key K and a public seed R, they can generate shared pseudonyms by iteratively keyed hashing R, where $H_K^{(n)}(R)$ refers to the message authentication code resulted from iteratively hashing R n times using the key K. The hash values generated from hashing R with odd numbers ($H_K^{(1)}(R)$, $H_K^{(3)}(R)$, etc) are used by node W and the those generated from hashing R with even numbers ($H_K^{(2)}(R)$, $H_K^{(4)}(R)$, etc) are used by node X. The frequency of pseudonym change (i) is the number of packets that use one pseudonym, e.g., each pseudonym is used only for one packet if i is one.

In order to keep pseudonym synchronization between W and X, each node compares a packet's pseudonym with the current and next pseudonyms. For example, in the packets (1 to i), W compares X's pseudonym to $H_K^{(2)}(R)$ and $H_K^{(4)}(R)$. Moreover, a node does not change its pseudonym more than once before the other node changes its pseudonym. In this way, if packet (i+1) is lost, the nodes do not lose synchronization because W does not use $H_K^{(5)}(R)$ before receiving $H_K^{(4)}(R)$ from X. After X receives $H_K^{(2)}(R)$, it knows that W wants to change pseudonym and thus it also changes its pseudonym by sending $H_K^{(4)}(R)$. The main advantage of this pseudonym generation technique is that the nodes do not have to change their pseudonyms at a fixed frequency. Pseudonym change can be arbitrarily triggered by X or W without losing synchronization. A pseudonym generation requires only one lightweight hashing operation and does not require large storage area or frequently contacting Tp to re-fill pseudonyms. This enables the nodes to reduce the lifetime of each pseudonym to improve the users' privacy. Pseudonyms can also be computed before receiving a packet to avoid delaying the packet relay.

PRIPO requires three types of symmetric keys and pseudonyms:-

1) Node-to-Tp: Node X and Tp share a long-term key K_X. Using this key, they can generate a long term pseudonyms ID_{XTp} and ID_{TpX}.

2) Node-to-Base Station: Each node shares a symmetric key and pseudonyms with its cell's base station. Once the node leaves the cell, the key and the pseudonyms become invalid. When node X first joins a new cell, Tp mutually authenticates the node and the cell's base station. As shown in Fig. 2, node X sends an *Authentication Request* (*AREQ*) packet containing a pseudonym shared with Tp (ID_{XTp}) and the encryption of its real identity and ID_{XTp}, where (M)K refers to the ciphertext resulted from encrypting M with K. *AREQ* authenticates X to Tp because the secret key K_X is required to compose the packet. Tp replies with the node's real identity, the shared key between X and Bs ($K_{XBs} = K_{BsX}$), and the seed of the pseudonyms (R). R and K_{XBs} are used to generate pseudonyms shared between X and Bs. In this way, Tp mutually authenticates X and Bs without revealing the node's long-term secret key.

3) Node-to-Node: In route establishment phase, the base station authenticates each two neighboring nodes W and X to each other, and distributes a one-session shared key ($K_{WX} = K_{XW}$) to generate one-session pseudonyms ID_{WX} and ID_{XW}.

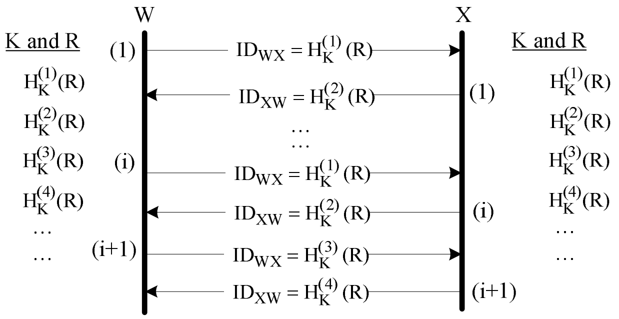


Fig. 1: Pseudonyms generation technique.

X → Bs→ Tp: <*AREQ*, ID_{XTp}, $(ID_{XTp}, ID_X)K_X$>
Tp → Bs: <$(ID_X, K_{BsX}, R, ID_{TpX}, (R, K_{XBs})K_X)K_{TBs}$>
Bs → X: <*AREP*, ID_{TpX}, $(R, K_{XBs})K_X$>

Fig. 2: Authentication phase.

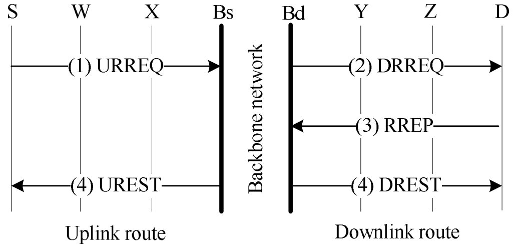


Fig. 3: Route Establishment phase.

B. Route Establishment Phase

As shown in Fig. 3, S broadcasts an *Uplink Route Request* (*URREQ*) packet that is forwarded by Bs to Bd if D resides in a different cell. Bd broadcasts the *Downlink Route Request* (*DRREQ*) packet and D sends back the *Route Reply Packet* (*RREP*) packet. Finally, Bs and Bd send the *Uplink* and the *Downlink Route Establishment packets* (*UREST* and *DREST*) to establish the uplink and the downlink routes.

URREQ: As shown in Fig. 4, the *URREQ* packet contains dummy bits called padding (Pad) and the encryption of the source and the destination nodes' real identities, the padding length (P_L), and a unique request identifier (Un_i). Un_i contains the pseudonym shared with Bs and time stamp. The encryption part authenticates S to Bs. The random-length padding prevents the attackers from learning the anonymous source node's location from the packet size and confuses the neighbors of S whether the packet is sent or relayed by S. Each intermediate node adds its pseudonym shared with Bs and broadcasts the packet. It also stores Un_i in the routing table and drops any further requests with the same identifier to broadcast the request once and avoid routing loops. For the first *URREQ* packet, Bs decrypts the encryption part to know the real identity of the destination node and the padding length, and forwards the request to D.

DRREQ: As shown in Fig. 5, the *DRREQ* packet contains Time-To-Live (TTL), a unique request identifier (Dn_i) that contains the pseudonym shared with D and time stamp, and the real identity of the source node encrypted by the shared key with D. Bd does not add padding because we do not aim to preserve the base station's location privacy. Each intermediate node adds its pseudonym shared with Bd and broadcasts the packet if it is not the destination and TTL is greater than zero. Each node stores Un_i in the routing table and drops any further requests with the same identifier to broadcast the request once and avoid routing loops. D broadcasts the packet as well after adding its pseudonym to deprive the attackers from inferring the destination of the packet. PRIPO uses very efficient trapdoor to inform D about the session. D only compares the packet pseudonym with its one. This is important because the *DRREQ* packets are received by a large number of nodes.

RREP: Fig. 5 shows that the *RREP* packet contains the identities of the nodes in the route and padding to protect the location privacy of the destination node. Each intermediate node relays the packet after replacing its pseudonym with the pseudonym of the next hop node.

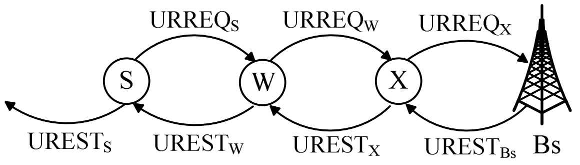
UREST: The objective of the *UREST* packet is to inform the intermediate nodes to act as packet forwarders and distribute the session keys shared between each two neighboring nodes. From Fig. 4, the *UREST* packet contains a fresh pseudonym shared with each node and session key. Each intermediate node removes one encryption layer using the shared key with Bs, removes its pseudonym and saves the session key shared with its previous neighbor in the route. The node hashes this key to get the shared key with the other neighbor, e.g., node W uses K_{SW} to communicate with S and $H_{KWBs}(K_{SW})$ to communicate with X. Obviously, $H_{KWBs}(K_{SW})$ is similar to K_{XW}. In this way, the number of distributed keys is nearly halved in order to reduce the packet overhead. Only the intended nodes can decrypt the packet, which is important for authorizing the network access and securing the payment. Padding is added to preserve the source node's location privacy, i.e., it is difficult to infer the source node's location from the *UREST* packet size. The source node relays the *UREST* to prevent its neighbors from knowing that it is the source.

DREST: From Fig. 5, the format of the *DREST* packet is the same as the *UREST* packet.

C. Data Transfer Phase

As shown in Fig. 6, the data packet at S contains the shared pseudonym with W (ID_{SW}), the message's number (C), and the message (M_C) and its message authentication code ($H_{KSBs}(M_C)$), all encrypted with the shared key with W. Each intermediate node replaces the packet's pseudonym with the one shared with the next node, and encrypts the iteratively-encrypted part with the shared key with the next node. The source base station removes the encryption layers, checks the message integrity, and forwards it to the destination base station.

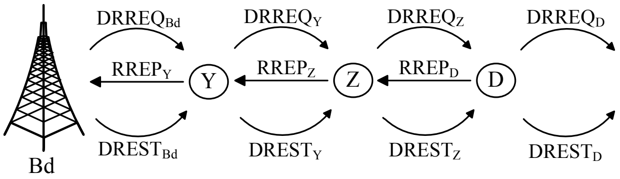
The destination base station iteratively encrypts the message with the keys shared between each two nodes. Each intermediate node checks whether the packet's pseudonym belongs to it and decrypts one layer of the data onion and changes the pseudonym with the one shared with the next node and relays the packet. The destination node acknowledges the messages it correctly receives. In this way, each intermediate node performs only one encryption or decryption operation but the base stations perform more operations. PRIPO can be used for bidirectional communication without any modification. The packet overhead is only one pseudonym instead of attaching the whole route identities similar to DSR routing protocol.



URREQ$_S$: $<Un_i, (Un_i, ID_S, ID_D, P_L)K_{SBs}, Pad>$
URREQ$_W$: $<Un_i, (Un_i, ID_S, ID_D, P_L)K_{SBs}, Pad, ID_{WBs}>$
URREQ$_X$: $<Un_i, (Un_i, ID_S, ID_D, P_L)K_{SBs}, Pad, ID_{WBs}, ID_{XBs}>$

UREST$_{Bs}$: $<ID_{BsX}, (Un_i, K_{XBs}, ID_{BsW}, (Un_i, K_{WX}, ID_{BsS}, (Un_i, K_{SW}, Pad)K_{BsS})K_{BsW})K_{BsX}>$
UREST$_X$: $<ID_{BsW}, (Un_i, K_{WX}, ID_{BsS}, (Un_i, K_{SW}, Pad)K_{BsS})K_{BsW}>$
UREST$_W$: $<ID_{BsS}, (Un_i, K_{SW}, Pad)K_{BsS}>$
UREST$_S$: <Pad>

Fig. 4: Anonymous uplink route establishment.

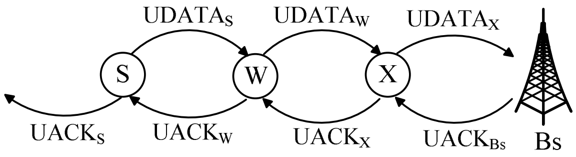


DRREQ$_{Bd}$: $<Dn_i, TTL, (Dn_i, ID_D, ID_S)K_{BdD}>$
DRREQ$_Y$: $<Dn_i, TTL\text{-}1, (Dn_i, ID_D, ID_S)K_{BdD}, ID_{YBd}>$
DRREQ$_Z$: $<Dn_i, TTL\text{-}2, (Dn_i, ID_D, ID_S)K_{BdD}, ID_{YBd}, ID_{ZBd}>$
DRREQ$_D$: $<Dn_i, TTL\text{-}3, (Dn_i, ID_D, ID_S)K_{BdD}, ID_{YBd}, ID_{ZBd}, ID_{DBd}>$

RREP$_D$: $< ID_{ZBd}, (ID_{DBd,} ID_S, ID_{D,} ID_{ZBd}, ID_{YBd,} Pad)K_{DBd}>$
RREP$_Z$: $<ID_{YBd}, (RREP_D)K_{ZBd}>$
RREP$_Y$: $<ID_{YBd}, (RREP_Z)K_{YBd}>$

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

Fig. 5: Anonymous downlink route establishment.



UDATA$_S$: $<ID_{SW}, (C, M_C, H_{KSBs}(M_C))K_{SBs}>$
UDATA$_W$: $<ID_{WX}, ((C, M_C, H_{KSBs}(M_C))K_{SBs})K_{WX}>$
UDATA$_X$: $<ID_{XBs}, (((C, M_C, H_{KSBs}(M_C))K_{SBs})K_{WX})K_{XBs}>$

DDATA$_{Bd}$: $<ID_{BdY}, (((C, M_C, H_{KBdD}(M_C))K_{BdD})K_{YZ})K_{BdY}>$

UACK$_{Bs}$: $<ID_{BsX}, (((H_{KBsS}(M_C))K_{BsS})K_{XW})K_{BsX}>$
UACK$_X$: $<ID_{XW}, ((H_{KBsS}(M_C))K_{BsS})K_{XW}>$
UACK$_W$: $<ID_{WS}, (H_{KBsS}(M_C))K_{BsS}>$
UACK$_S$: <Random>

Fig. 6: Anonymous uplink data transmission.

D. Accounting and Auditing Phase

To avoid instantaneously contacting Tp in each session, the base stations manage payment reports for the nodes in their cells and submit the reports to Tp. The payment reports contain the number of messages sent, received, and relayed by the nodes. Once Tp receives the payment reports from the base stations, it updates the nodes' credit accounts accordingly.

V. SECURITY ANALYSIS

A. Defense against Payment Manipulation

The iterative encryption/decryption operations can protect against several attacks. First, removing the encryptions and verifying the correctness of the resulting packet implicitly authenticates the intermediate nodes and ensures that the packet is relayed through the route it was supposed to take. Second, in *Free-Riding* attacks, two colluding nodes C1 and C2 in a legitimate session manipulate the session packets to add their data to communicate freely. The iterative encryption/decryption operations can thwart the attack because the data sent by C1 cannot be interpreted by C2 because it is encrypted (or decrypted) by at least one intermediate node. Third, the iterative encryption/decryption operations make the packets look different as they are relayed, which makes packet likability and tracing not possible, as we will discuss in Section V-B.

For *Packet-Replay* attack, the internal and external attackers may record valid packets and replay them in different place and/or time claiming that they are fresh to establish sessions under the name of others to communicate freely. In PRIPO, if the attacker replays the *URREQ* packet, he cannot establish the session because he cannot generate a fresh pseudonym or decrypt the *UREST* packet to get the key shared with his neighbor. In addition, since the source node encrypts a time stamp in the *URREQ* packet, the attacker cannot send valid packets without knowing a secret key because the packets are eventually dropped at the base station.

For *Impersonation* attack, the attackers attempt to impersonate other nodes to communicate freely. This attack is not possible in PRIPO because the nodes have to authenticate themselves using the long-term keys shared with Tp in order to share a key with the base station. For *Man-in-the-Middle* attack, the attacker residing between a node and Tp may attempt to get the key shared between the node and the base station to communicate freely under the name of the node. PRIPO is not vulnerable to this attack because the shared key with the base station is encrypted with the node's long-term key.

For *Destination-Node-Robbery* attack, the source node colludes with some intermediate nodes to steal credits from the destination node by sending bogus data. In PRIPO, the intermediate nodes are rewarded only when the destination node acknowledges receiving correct data, and the session cannot be established if the destination node is not interested in the communication because it has to send the *RREP* packet. For *Credit-Overspending* attack, the nodes may spend more than the amount of credits they have at the time of the communication. Most of the existing incentive schemes [3-6] are vulnerable to this attack because they use post-paid payment policy, i.e., the nodes communicate first and pay later. PRIPO is not vulnerable to this attack because the base stations know the nodes' total credits from Tp during authentication phase and do not allow the nodes to overspend their credits.

Although, the charges are always more than or equal to the rewards, the payment model does not make credits disappear because purchasing credits for real money can compensate the credit loss. The payment model can encourage node cooperation and counteract cheating actions without submitting payment receipts as follows:

1) The nodes are motivated to relay the data packets because the nodes are rewarded only when the packets are delivered;

2) Relaying the route establishment packets is beneficial for the nodes to participate in a session and thus earn credits. Relaying the *ACK* packets can trigger the source node to generate more packets and thus earn more credits. It is also beneficial for the downlink nodes because they are rewarded only when the *ACK* packets reach to Bd; and

3) If the nodes are charged only when the destination node receives a message, the node may claim that it does not receive the message in order not to pay. To prevent this, both S and D are charged for un-delivered messages.

B. Defense against Privacy Violation

Identity Privacy: The real identity is always kept confidential and never disclosed in clear. The nodes use pseudonyms in their communications to preserve identity privacy. A node's pseudonyms cannot be linked to the real identity or to each other without knowing a secret key. Since pseudonym generation requires a lightweight hashing operation, a pseudonym can be used for a very short time to significantly improve the identity privacy. In *AREQ*, *URREQ*, and *DRREQ* packets, the real identities are concatenated with a varying part before encryption, e.g., in *AREQ*, ID_X and K_X are fixed but ID_{XTp} is dynamic. This makes the packets look different at each time the node sends them, and thus even if an attacker could link a packet to a node, he cannot benefit from this conclusion in future. In data transfer phase, if an attacker could link an onion data to a node, this will not help in future because the onion data will look different even if the same message is sent because the nodes use one-session keys.

The base stations know the real identities of the nodes in its cells but they do not know their long-term secret keys. PRIPO can easily be modified to hide the nodes' real identities from base stations, but more overhead is encountered for contacting Tp to route messages from Bs to Bd. PRIPO offers both sender and receiver anonymity as well as sender-receiver relationship anonymity. In PRIPO, S and D know the real identities of each other but they do not know the locations of each other. The intermediate and eavesdropping nodes cannot learn the real identities of S and D and their locations in number of hops.

Pseudonym De-synchronization: In Section IV-A, we have shown that the loss of pseudonym synchronization is difficult. However, if a node loses pseudonym synchronization with the base station for any reason, the node can re-synchronize by initiating a new authentication process. Since a node cannot change its pseudonym more than once before the base station changes its pseudonym to avoid synchronization loss, the nodes may use one pseudonym in the *RREQ* packets for a long time if they do not participate in a route. This may be specifically applied to the nodes at the network border because they are less frequently selected by the routing protocol. The attackers may initiate *RREQ* packets to learn whether a node is still in the neighborhood. The proposed protocol for establishing the uplink route shown in Fig. 4 can be used but for identity change request. The Pad can be a pre-defined value to inform the base station that the packet is for identity change. Bs replies with *URREP* packet containing a new pseudonym.

Location Privacy: Padding is used to prevent the external and internal attackers from locating the source and the destination nodes from packet size. Moreover, the destination node relays the data and the route establishment packets to confuse its neighbors whether the node is intermediate or destination. Since *UREST* and *DREST* packets are relayed fixed TTL hops regardless of the location of the intended node, an attacker cannot know the locations of S and D. The attacker can know that he has a neighbor with a certain pseudonym but once the neighbor changes its pseudonym, it is difficult to know whether the new pseudonym is for the old neighbor or for a new one.

Route Privacy: It is the capability of preventing the attackers from tracing a packet flow backward to its original source or forward to its final destination. The iterative encryption/decryption operations make the same packet appear quite different across links. Thus, the attackers overhearing the transmissions of two nodes in a route cannot recognize that the two nodes relay the same communication flow. Moreover, the base station can shuffle the received packets and relay them in a random order to prevent the attackers from using temporal dependency to correlate the ingoing and outgoing packets.

VI. Performance Evaluation

A. Cryptographic Overhead

To evaluate the computational time of the cryptographic operations used in PRIPO, we have implemented AES symmetric key cryptosystem and SHA-1 (160 bit) hash function using Crypto++ library [10]. The secure key size is at least 128 bits according to NIST [11]. The mobile node is a laptop with an Intel processor at 1.6 GHZ (CPU) and 1.00 GB Ram, and Windows XP operating system. The results demonstrate that a hashing operation requires 16.79 Megabytes/s and encryption and decryption operations require 9.66 Megabytes/s. these results are scaled by the factor of ten to emulate a limited resource node. For the energy consumption, it is shown in [12] that a hashing operation requires 0.76 μJ/byte and encryption and decryption operations require 1.21 μJ/byte.

B. Communication Overhead

PRIPO was simulated using a network simulator written in MATLAB. 35 mobile nodes are randomly deployed in a square cell of 1000 m × 1000 m, and a base station is located at the center. The radio transmission range of the mobile nodes and the base station is 125 m. The random waypoint model is used to emulate the node mobility. The node speed is uniformly distributed in the range [0, 3] m/s and the pause time is 20 s. The constant bit rate traffic source is implemented in each node as an application layer. The source and destination pairs are randomly selected. The packets are sent at the rate of 2 packets/s. Distributed Coordination Function (DCF) of IEEE 802.11 is simulated as a medium access control (MAC) layer protocol. Our simulation is executed for 15 minutes and the results represent the average of 50 runs. The pseudonyms can be truncated into shorter length without significantly increasing the probability of pseudonym collision. The length of the truncated pseudonym (δ) depends on the cell size and the number of nodes in the cell. δ can be frequently computed by the base station and broadcasted. The length of δ, Pad, time stamp, real identity, and M_C are 10, $2 \cdot \delta$, 5, 4, and 512 bytes, respectively.

The simulation results given in Table 1 indicate that the expected delay is acceptable due to lightweight cryptographic operations. The average length of the *URREQ* packet is computed by dividing the amount of relayed data in all links by the number of links. The simulation results show that only 24-byte packet overhead are added to each message.

Table 1: Simulation results.

	URREQ	**RREP**	**DREST**	**Data Packet**
Avg. packet length (bytes)	73.68	95.31	170.27	534
Avg. delay (ms)	19.3647	21.351	21.242	32.7612

VII. Conclusion and Future Work

We have proposed a privacy-preserving routing and incentive protocol for hybrid ad hoc wireless network. Micropayment is used to stimulate node cooperation without submitting receipts. Our protocol can achieve a high protection level for user privacy using lightweight cryptographic tools. For efficient generation of pseudonyms, only the lightweight hashing operations are required. Extensive evaluations and simulations demonstrate that node cooperation and privacy preservation can be securely and efficiently integrated in one protocol.

Similar to the existing incentive schemes, PRIPO thwarts selfishness attacks but cannot identify the malicious nodes that drop packets to launch *Denial-of-Service* attacks. The base stations can inform Tp how frequency the nodes drop the packets. However, packets can be dropped normally, e.g., due to mobility, or maliciously, but the high frequency of packet drop is an obvious malicious behavior. In our future work, we will study how Tp can precisely differentiate between honest and malicious nodes in order to reduce the false positive ratio.

References

[1] A. Abdrabou and W. Zhuang, "Statistical QoS routing for IEEE 802.11 multihop ad hoc networks", IEEE Transactions on Wireless Communications, vol. 8, no. 3, pp. 1542 - 1552, March 2009.

[2] K. Liu, J. Deng, and K. Balakrishnan, "An acknowledgement-based approach for the detection of routing misbehavior in MANETs", IEEE Transaction on Mobile Computing, vol. 6, no. 5, May 2007.

[3] S. Zhong, J. Chen, and R. Yang, "Sprite: A simple, cheat-proof, credit based system for mobile ad-hoc networks", Proc. of IEEE INFOCOM, vol. 3, pp. 1987-1997, San Francisco, CA, March 30- April 3, 2003.

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

[5] M. Mahmoud and X. Shen, "Stimulating cooperation in multi-hop wireless networks using cheating detection system", Proc. of IEEE INFOCOM, pp. 776–784, San Diego, CA, March 14–19, 2010.

[6] 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.

[7] 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.

[8] J. Kong and X. Hong, "ANODR: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks", Proc. of ACM MobiHoc, pp. 291-302, Annapolis, Maryland, USA, June 1-3, 2003.

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

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

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

[12] 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, no. 2, pp. 128-143, March-April 2006.