PIS: A Practical Incentive System for Multi-hop Wireless Networks

Mohamed Elsalih Mahmoud and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—In multi-hop wireless networks, the mobile nodes usually act as routers to relay other nodes' packets for enabling new applications and enhancing the network performance and deployment. However, selfish nodes may not cooperate and make use of the cooperative nodes to relay their packets, which has negative effect on the network fairness, security, and performance. Incentive systems implement micropayment in the network to stimulate the selfish nodes to cooperate. However, micropayment schemes have been originally proposed for web-based applications, so a practical incentive system should consider the differences between web-based applications and cooperation stimulation. In this paper, first, these differences are investigated and a payment model is developed for efficient implementation of micropayment in multi-hop wireless networks. Second, based on the developed payment model, an incentive system is proposed to stimulate the nodes' cooperation in multi-hop wireless networks. Third, reactive receipt submission mechanism is proposed to reduce the number of submitted receipts and protect against collusion attacks. Extensive analysis and simulations demonstrate that our incentive system can secure the payment, and reduce the overhead of storing, submitting, and processing the payment receipts significantly, which can improve the system's practicality due to the high frequency of low-value payment transactions.**

***Index Terms*—Cooperation Stimulation, Incentive Systems, Packet Drop Attack, Selfish Mobile Nodes.**

1. INTRODUCTION

THE interest in multi-hop wireless networks (MWN) such as mobile ad-hoc network (MANET), vehicular ad-hoc network (VANET), multi-hop cellular network (MCN), and wireless mesh network (WMN) has been increasing significantly [1]-[3]. In these networks, the traffic originated from a node is usually relayed through other nodes to the destination. Multi-hop relaying can extend the communication range using limited transmit power, improve area spectral efficiency, and enhance the network throughput and capacity [4], [5]. Moreover, these networks can be deployed more readily and at low deployment cost in developing or rural areas. However, due to involving autonomous nodes in packet relay, the routing process suffers from new security challenges that endanger the practical implementation of the MWNs.

Most existing routing protocols assume that the nodes of multi-hop wireless network are willing to relay other nodes' packets. This assumption is reasonable in disaster recovery and military applications because the nodes belong to a single authority and pursue a common goal, but it may not hold for civilian applications where the nodes are autonomous and aim to maximize their welfare. Although the proper network operation requires the nodes to collaborate, collaboration consumes their valuable resources such as energy and computing power, which stimulates the nodes to behave selfishly. Therefore, in civilian applications, selfish nodes are not voluntarily interested in cooperation without sufficient incentive and make use of the cooperative nodes to relay their packets, which has negative effect on the network fairness, performance, and security. Fairness issue arises when selfish nodes take advantage from the cooperative nodes without any contribution to the network, and the cooperative nodes are unfairly overloaded because the network traffic is concentrated through them. The selfish behavior also degrades the network performance significantly, which may result in failure of the multi-hop communication [6], [7].

Reputation-based and incentive systems [8], [9] have been proposed to enforce and stimulate node cooperation, respectively. In reputation-based systems, each network node usually monitors the transmissions of its neighbors to make sure that the neighbors forward others' traffic, and thus selfish nodes can be identified and punished. In incentive systems, forwarding other nodes' packets is a service not an obligation, so the communicating nodes pay credits (or virtual currency) to the intermediate nodes to relay their packets. However, reputation-based systems [10], [11] suffer from essential problems that may discourage implementing them practically. First, to monitor the transmissions of its neighbors, a network node usually works in the promiscuous mode that is not efficient because the node uses the full power transmission instead of adapting the transmission power according to the distance separating the transmitter and the receiver [12]. Furthermore, the directional antennas [13] that can improve the network capacity due to reducing the interference area make monitoring difficult. Second, reputation-based systems do not achieve fairness because the high-contribution nodes are not compensated, and the nodes are punished when they do not cooperate no matter how they have previously contributed to the network. For example, although the nodes situated at the network center relay more packets than those at the periphery, they are not compensated. Third, these systems suffer from unreliable detection of the selfish nodes and false accusation of the hon-

Manuscript received May 14, 2010; revised July 5, 2010; accepted July 19, 2010. First published ****; current version published **. The review of this paper was coordinated by Dr. Liqun Chen.

M. Mahmoud and X. Shen are with the Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: mmabdels@bbcr.uwaterloo.ca; xshen@bbcr.uwaterloo.ca).

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

Digital Object Identifier **.

est nodes because it is difficult to differentiate between a node's unwillingness and incapability to cooperate, e.g., due to low resources or network congestion. Moreover, the assumption that the transmitted packets by a node can be received by all the nodes in its neighborhood cannot be ensured, e.g., due to packet collision [14]. Finally, reputation-based systems have not considered the possibility that selfish nodes can collude with each other to boost their reputations to maximize their welfare.

Incentive systems are better for multi-hop wireless networks because in addition to cooperation stimulation, the systems can achieve fairness by charging or rewarding credits to balance between a node's contributions and benefits. A node's contribution can be relaying other nodes' packets or paying credits, whereas a node's benefit can be relaying its packets or earning credits. Moreover, since the network nodes pay for relaying their packets, incentive systems can discourage resource exhaustion attack where malicious nodes exchange bogus packets to exhaust the intermediate nodes' resources. Incentive systems can also be used for charging future services of mobile networks [15], [16] because communication sessions may occur without involving an infrastructure, and mobile nodes may roam among different foreign networks. In other words, by using incentive system, the network nodes can pay all parties involved in its communication without contacting distant home location registers.

However, the practicality of the existing incentive systems is questionable because they impose significant overhead cost. Micropayment schemes [17]-[19] are electronic payment schemes for frequent and low-value payments. The schemes were originally designed for the Internet electronic commerce applications to take advantage of the high volume of viewers by offering content for low price. Examples of the applications include buying data or news, listening to a song, playing an online game, and reading an article in a journal [20]. In order to implement such scheme in multi-hop wireless networks efficiently, the differences between web-based applications and cooperation stimulation should be taken into account. These differences are summarized in Table I.

TABLE I
PROPERTIES OF WEB-BASED APPLICATIONS AND COOPERATION STIMULATION

			Web-based	Cooperation Stimulation
P1	**Transaction parties**		One customer and one merchant	One or more merchants and two customers
P2	**Merchants'**	**number**	Low	Large
		identities	Known in advance	Unknown in advance
P3	**Customer-merchant relation**		Long	Very short
P4	**Transaction frequency**		High	Very high
P5	**Transaction value**		Low	Very low
P6	**Easiness of misbehavior**		Very easy	Less
P7	**Nodes' resources**		High	Low

For web-based applications, a transaction usually contains one customer and one merchant, and merchants' number is low and their identities are known before the transaction is held. For cooperation stimulation, each transaction usually contains two customers (the source and the destination nodes) and multiple merchants (the intermediate nodes), the merchants' number is large because any network node can work as a merchant (or packet relay), and the merchants' identities are known only at the transaction (session) time due to the nodes' mobility. Moreover, the relation between a customer and a merchant is usually short due to the network dynamic topology. The nodes are involved in low-value transactions very frequently because once a route is broken, which is frequently due to nodes' mobility and channel impairment, a new transaction should be held to re-establish the route. In wireless networks, the nodes have low resources such as energy and storage area, comparing with computers' resources in web-based applications. Although security is important in all payment applications, attacks can be launched easier in web-based applications because it is easier to launch attacks across the Internet than tampering devices.

In this paper, first, we develop a payment model that takes into account the features of cooperation stimulation, which can improve the practical implementation of micropayment in multi-hop wireless networks. Second, based on the developed payment model, we propose **PIS**, **P**ractical **I**ncentive **S**ystem, to stimulate the nodes' cooperation in multi-hop wireless network. Since the communication sessions may occur without involving an infrastructure, the communicating nodes issue digital receipts to the intermediate nodes that submit the receipts to the AC to claim their payment. The conclusive point for practical implementation of incentive systems is the receipts' submission and process overhead due to the high frequency of low-value transactions. In other words, submitting a large number of receipts implies significant communication and computation overhead, and implementation difficulty because the cost of receipt submission and process may exceed the transaction value. Therefore, in our incentive system, instead of generating a receipt per packet or group of packets, one fixed-size receipt is generated per session regardless of the number of transmitted packets. Moreover, a receipt size can be reduced significantly by attaching the hash of the nodes' signatures instead of the signatures, and different receipts can be aggregated together to a smaller-size receipt.

Our third contribution is proposing a reactive receipt submission mechanism to reduce the number of submitted and processed receipts, and protect against collusion attacks. In our incentive system, a receipt contains payment data for all the intermediate nodes, so it is sufficient to submit one copy to clear the receipt. However, it is not secure to trust one node to submit the receipt because it may collude with the communicating nodes (payers) so as not to submit the receipt to increase their welfare. The mechanism is a reactive one because unlike the *Preventive* mechanism [21] that requires submitting a large number of redundant receipts *to fail the collusion attacks*, our mechanism submits an incomparable number of redundant receipts *to identify the colluding nodes*, and the number of un-submitted receipts can be limited probabilistically. Extensive analysis and simulations demonstrate that our incentive system can secure the payment, and significantly reduce the number of submitted and processed receipts espe-

cially at high packet transmission rate, which can improve the system's practicality due to the high frequency of low-value transactions.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the system models. Section 4 proposes our incentive system. Security analysis and performance evaluation are provided in Sections 5 and 6, respectively, followed by conclusion in Section 7.

2. Related Work

The existing incentive systems can be classified into two categories: tamper-proof-device (TPD) and central-bank based systems. In TPD-based incentive systems [22]-[26], a tamper-proof device is installed in each device to manage its credit account and secure its operation. In central-bank based incentive systems [21], [27]-[34], a centralized unit called the accounting center (AC) stores and manages the nodes' accounts.

In Nuglets [22], the self-generated and forwarding packets are passed to the node's TPD to decrease and increase its credit account, respectively. Two payment models, called the packet purse model (PPM) and the packet trade model (PTM), have been proposed. In the PPM, the source node pays for relaying its packets by loading some credits in each packet before sending it, each forwarding node acquires the amount of credits that cover the packet's forwarding cost, and the packet is dropped if it runs out of credits. In the PTM, each intermediate node runs auctions to sell the session packets to the following node on the route, and the destination node pays the total cost. In SIP [23], after receiving a packet, the destination node sends a payment RECEIPT packet to the source node to issue a REWARD packet which increments the credit counters of the intermediate nodes. In CASHnet [24], [25], for each transmitted packet, the source node's traffic-credit account stored in the node is charged and signature is attached. Upon receiving the packet, the destination node's traffic-credit account is also charged and a digitally signed ACK packet is sent back to increase the helper-credit accounts of the intermediate nodes. Users regularly visit service points to buy traffic credits for real money and/or convert helper credits to traffic credits.

Centralized-bank based incentive systems can be classified into coin and receipt based systems. In coin-based incentive systems [27], a network node buys electronic coins from the AC before being involved in a session to pay for relaying its packets. In receipt-based incentive systems [21], [28]-[34], the communicating nodes issue receipts (or payment proofs) to the intermediate nodes that submit the receipts to the AC to update their accounts, i.e., a network node communicates first and pays later. In [27], each node on a communication session buys packets from the upstream node and sells them to the downstream node. Packets' buyer and seller contact the AC to get deposited coins and submit the coins to claim the payment, respectively, so the session's nodes contact the AC in each session.

In [28], the source node appends a token to each transmitted packet, and each intermediate node uses its secret key to check whether the token corresponds to a winning ticket. Winning tickets are submitted to the AC to reward the winning nodes. In a security flaw, colluders can collect and exchange tokens to be checked in each node to steal credits. Since the nodes are rewarded only for winning tickets, fairness issue arises when a node is not compensated adequately for its cooperation. In our earlier work [29], instead of submitting payment receipts to the AC, each node submits a smaller-size activity report containing its alleged charges and rewards in different sessions, and the AC uses a reputation system to identify the cheating nodes. However, due to the nature of the reputation systems, some honest nodes may be identified as cheaters falsely and colluding nodes may manage to steal credits. In [30], an incentive system has been proposed for hybrid ad hoc network, but the base station is involved in every communication session, which may lead to suboptimal routes when the communicating nodes reside in the same cell.

In Sprite [31], the source node signs the full path identities and appends its signature to each transmitted message. In Sprite and INPAC [32], the intermediate and destination nodes compose receipts and submit them to the AC to claim the payment. In Express [33], the source node generates a hash chain for every intermediate node ID_K by iteratively hashing random value (V_S) S times to obtain a final hash value V_0. The source node commits to the hash chain by digitally signing V_0 and sending the signature to ID_K. Each time ID_K relays a packet, the source node releases the pre-image of the last sent hash value, where V_1 is released first and V_2 second and so on. Payment non-repudiation is achievable because the hash function is one-way, i.e., only the source node can generate the hash chain. The source, intermediate, and destination nodes compose receipts and submit them to the AC. However, each node has to generate and store a large number of hash chains due to the large number of probable relays (P2 in Table I). In Sprite, INPAC, and Express, only the source node pays no matter how the destination node benefits from the communication. Moreover, since an intermediate node is rewarded for every successfully relayed packet even if it does not reach the destination, all the session nodes submit the receipts because a node's packet forwarding is considered successful if its next node on the path reports a valid receipt. We call this receipt submission mechanism *All-Submitters* because all the intermediate nodes submit all the receipts. In Sprite, INPAC and Express, significant communication and computation overhead is implied due to submitting and processing a large number of receipts because a receipt is generated per packet and submitted by all the nodes on the session.

In [34], an incentive system has been proposed for ad hoc network that is used to connect the nodes to the Internet. 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 to submit to the AC. Since a receipt contains payment data for all the intermediate nodes, one copy of the receipt is submitted to claim the payment for all the intermediate nodes, which we call *One-Submitter* receipt submission mechanism. However, the proposed system does not handle collusion attacks, e.g., the communicating nodes can communicate freely and the inter-

mediate nodes are not rewarded when the last intermediate node colludes with the payers so as not to submit the receipts.

In DSC [21], instead of generating a hash chain for each intermediate node like Express, one hash chain of size S is generated to pay for all the intermediate nodes on a session. A receipt contains payment data for all the intermediate nodes and at most S packets. A receipt for X packets contains the root hash value (V_0) and the last released hash value (V_{X-1}). The number of transmitted packets is computed from the number of hashing operations to map V_{X-1} to V_0. A new receipt is generated when S packets are sent or the route is broken. Furthermore, since a receipt contains payment data for all the intermediate nodes, *Preventive* receipt submission mechanism has been proposed to reduce the number of submitted receipts and prevent collusion attack. In this mechanism, each intermediate node submits a number of randomly chosen receipts that guarantees submitting a minimum number of un-repeated receipts probabilistically. The mechanism is preventive one because it aims to prevent the effectiveness of collusion attack by submitting redundant receipts, i.e., even if some colluding nodes do not submit the receipts, they may be submitted by other nodes, so a minimum number of un-repeated receipts can be submitted under collusion attack. Increasing the number of submitted receipts by each node increases the robustness against collusion attack but with additional redundant receipts.

3. System Models

3.1 Network and Communication Models

The considered multi-hop wireless network includes an AC, mobile nodes, and base stations in some types of the MWNs. The AC stores and manages the nodes' credit accounts, and generates private/public key pair and certificate with unique identity for each node to participate in the network. Once the AC receives a receipt (proof of payment), it updates the relevant nodes' accounts, and identifies and revokes the misbehaving nodes.

An on-demand routing protocol, such as DSR [35] and AODV [36], is implemented to establish an end-to-end communication session between the source and the destination nodes. The source node's packets may be relayed in several hops by the intermediate nodes to the destination. The network nodes can contact with the AC at least once during a time interval that can be in the range of few days. This connection can occur via base stations, Wi-Fi hotspots, or wired networks (e.g. Internet). During this connection, a network node renews or revokes its certificate, submits the payment receipts, and purchases credits by real money.

3.2 Threat and Trust Models

Since the mobile nodes are autonomous, we assume that an attacker has full control on his mobile node, and thus he can change its operation and infer the cryptographic data. Attackers can work individually or collude with each other under the control of one attacker to launch sophisticated attacks. Attackers are rational in the sense that they misbehave when they can achieve more benefits than behaving honestly. Specifically, attackers attempt to steal credits, pay less, and communicate freely. The base stations and the mobile nodes are probable attackers because they are motivated to misbehave to increase their welfare. However, the AC is fully trusted because it is impossible to realize secure payment between two entities without trusted third party [37]. For the trust models, the network nodes fully trust the AC to perform billing and auditing correctly, but the AC does not trust any node or base station in the network.

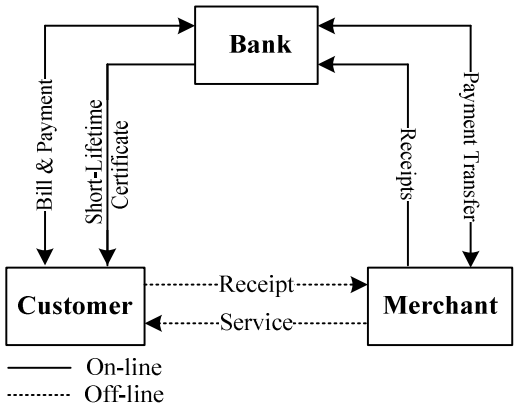


Fig. 1: The payment model's parties and relations.

3.3 Payment Model

3.3.1 Parties and Relations

The payment model contains three basic parties: the customer or the communicating nodes, the merchant or the intermediate nodes, and the bank or the accounting center. Fig. 1 portrays the relations among the different parties in our payment model. The operations among these parties can be divided into three phases: *Certificate Issuing*, *Payment*, and *Redemption*. In *Certificate Issuing Phase*, a customer has to register with the bank to create an account, and the bank issues a short-lifetime certificate, e.g., for seven to ten days. A customer contacts the bank periodically to renew his certificate and pay for the services (packet relay) he received from the merchants. In *Payment Phase*, a customer's certificate enables him to issue digital receipts to transact with merchants without involving the bank, i.e., customers mine their own electronic coins without the need of direct verification by the bank. In *Redemption Phase*, a merchant claims its payment by submitting its transactions' receipts. The AC verifies the receipts and clears them by rewarding the merchants and charging the customers. This payment architecture has two important properties that can improve the practical implementation of micropayment in multi-hop wireless networks: *no need for tamper proof device* and *flexible payment*.

a) No Need for Tamper-Proof-Device

The TPD-based incentive systems [22]-[25] may not find widespread acceptance for the following reasons. First, the assumption that the TPD cannot be tampered is neither secure nor practical for multi-hop wireless networks. Attackers can

communicate freely in undetectable way if they could compromise the TPDs [38] because the communications are decentralized and the network nodes are autonomous. Moreover, a small number of trusted manufactures can make the network devices, which is too restrictive for civilian networks. Second, a network node cannot communicate if it does not have sufficient credits. Unfortunately, the nodes at the network edge cannot gain as many credits as the nodes at other locations because they are less frequently selected by the routing protocol. Furthermore, the credit distribution has direct impact on the network performance, e.g., if a small number of nodes have large ratio of the network credits, the network performance degrades significantly because the rich nodes are not motivated to cooperate and the poor nodes cannot initiate sessions. Finally, since credits are cleared in real-time, the network performance degrades significantly if the network nodes do not have enough credits. Credits are lost normally because the total charges are not necessarily equal to the total rewards [26], e.g., the source node is charged full payment after sending a packet but some intermediate nodes might not be rewarded when the route is broken. In [23], a compensation mechanism is used to change the packet-relaying price proportionally to the nodes' speed to avoid the network credit decline. However, the compensation mechanism has to avoid credit inflation and depletion. For credit inflation, the nodes are rich and thus unmotivated to cooperation, whereas for credit depletion, the nodes are poor and incapable of initiating communication. The design of a decentralized compensation mechanism to stabilize the amount of credits in the network is difficult especially in large-scale networks.

In our payment model, the AC can convert credits to real money and sell credits for real money. This motivates the rich nodes to cooperate, enables the nodes that cannot gain credits because they are less frequently selected by the routing protocol to communicate, improves credit distribution, and protects the network from credit decline.

b) Flexible Payment

There are two ways for managing electronic payment: on-line and off-line payment. For on-line payment, a merchant verifies the payment sent by a customer with the bank before serving the customer; and for off-line payment, a merchant serves the customer without involving the bank at the transaction time, i.e., instead of interacting with the bank in each transaction, merchants accumulate the payments and redeem them in batch later. The payment management can also be classified into credit (or post-paid) and debit (or pre-paid) payment. For credit payment, customers are served first and charged later, e.g., customers issue receipts to the merchants that submit them to the bank to redeem the payment, so a customer's account will not be debited until receipts are processed. For debit payment, customers' accounts are charged before they are served, e.g., customers buy electronic coins in advance from the bank to pay to the merchants, or the bank has to be interactively involved in each session.

Off-line and credit payment is better for practical implementation of micropayment in MWNs for the following reasons. First, connection with the bank may not be available on regular basis, and even if it is available, involving a centralized unit in each transaction is very costly and creates bottleneck in the bank due to the high frequency of low-value transactions (P4 and P5 in Table I). Second, customers generate their own coins (or receipts), which provides many flexibilities. Coins are generated on-demand, and customers do not need to frequently contact with the bank to buy coins. In [26], it is shown that although some nodes have helper credits in CASHnet, they cannot communicate because they could not find a service point to convert the helper credits to traffic credits. Moreover, generating coins to pay for specific merchant [27]-[33] is not practical due to the large number of probable merchants in the network, and generating general coins to pay for any merchant are vulnerable to double spending attack or requires interactive and frequent contact to the bank.

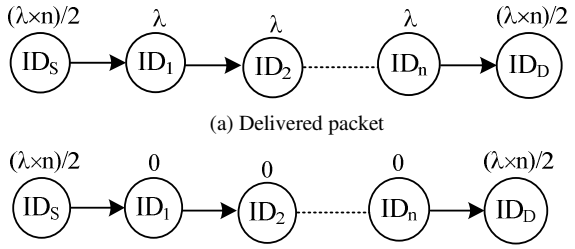
Although, the developed payment architecture has many positives, it is obvious that reducing the receipts' number is essential for practical implementation for the following reasons. First, since the transactions' number is large and multiple merchants may be involved in a transaction (P1 and P2 in Table I), generating a receipt per packet or customer increases the receipts' number significantly, and thus the transaction value may not cover its processing cost (P5 in Table I), and processing a large number of receipts may not be feasible. Second, the nodes have low resources (P7 in Table I) so the overhead of storing and submitting a large number of receipts may stimulate the nodes to behave selfishly. What makes it worse is that the nodes keep the receipts for some time because instantaneous contact with the bank may not be guaranteed.

3.3.2 Charging and Rewarding Policy

In most existing incentive systems [22], [23], [31]-[33], only the source node is charged. We argue that a more fair charging policy is to support cost sharing between the source and the destination nodes because both of them benefit from their communication. The payment ratio is adjustable and can be negotiated during the session establishment phase. To simplify our presentation, we suppose the source and the destination nodes agreed to halve the packet relaying expense, though any other payment-splitting ratio can be used. For rewarding policy, some incentive systems [39], [40] consider different packet relaying cost that corresponds to the incurred energy in packet relay. This rewarding policy is difficult to be implemented in practice without involving complicated route discovery process and calculation of en-route individual payments. Therefore, similar to [21], [23], [31], [33], we use fixed rewarding rate, e.g., λ credits per unit-sized packet.

In multi-hop wireless networks, packet loss may occur normally due to node mobility, packet collision, channel impairment, or other reasons. Ideally, any node that has ever tried to forward a packet should be rewarded no matter whether the packet eventually reaches its destination or not because forwarding a packet consumes the node's resources. However, it is difficult to corroborate an intermediate forwarding action in a trustable and distributed manner without involving too com-

plicated design. For example, rewarding the nodes for route establishment packets or packet retransmissions complicates the incentive system and increases the receipts' number significantly because large number of nodes may be involved in relaying route establishment packets and packet retransmissions happens frequently in wireless networks. Moreover, to reward the nodes for every relayed packet, all the intermediate nodes submit all the receipts to identify the last node that is relayed the packet before the route breakage [31]-[33].



(a) Delivered packet

(b) Undelivered packet

Fig. 2: Payment rewarding and charging policy.

TABLE II
USEFUL NOTATIONS

Symbol	Description
A, B	A is concatenated to B.
H(X)	The hash value resulted from hashing X.
ID_i	The identity of intermediate node i, or node with identity ID_i.
ID_S and ID_D	The identities of the source and the destination nodes, respectively.
M_i	The message sent in the ith data packet in a session.
n and n_C	The number of intermediate nodes and colluding *P-submitters*, respectively
P_S	The probability of submitting a receipt by a *P-submitter*.
R	The concatenation of identities of session nodes.
$Sig_i(X)$	The signature of intermediate node i on X.
$Sig_S(X)$ and $Sig_D(X)$	The signatures of the source and the destination nodes on X, respectively.
SR(C)	Session receipt for C packets.
TS	A session's establishment time stamp.

Therefore, in our incentive system, the first intermediate node after the source node submits the session receipt, and the other intermediate nodes submit the receipts probabilistically to protect against collusion attacks. In Fig. 2(a), the intermediate nodes are rewarded when the receipt proves that the message has been delivered, i.e., when the first intermediate node after the source node (ID_1) receives the message's ACK, or one of the other intermediate nodes receives the ACK and submits the receipt. However, in Fig. 2(a, b), the two communicating nodes are charged when the source node transmits a packet whether it reaches the destination or not. The value of λ is determined to compensate the nodes for their consumed resources in route establishment packets, packet retransmission, and undelivered packets. In Section 5, we will argue that our charging and rewarding policy can discourage rational attacks and encourage packet relay. Table II gives the used notations in this paper.

4. PIS: THE PROPOSED INCENTIVE SYSTEM

Fig. 3 shows that our incentive system consists of three phases: *Communication*, *Receipt Submission*, and *Payment Redemption and Colluder Identification*. In *Communication Phase*, the networks nodes are involved in sessions, and the communicating nodes issue payment receipts to the intermediate nodes. In *Receipt Submission Phase*, the nodes submit the receipts to the AC to claim their payments. In *Payment Redemption and Colluder Identification Phase*, the AC clears the receipts and identifies the colluding nodes that do not submit the receipts.

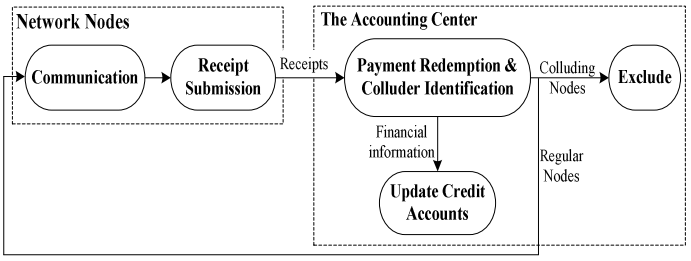


Fig. 3: The architecture of our incentive system.

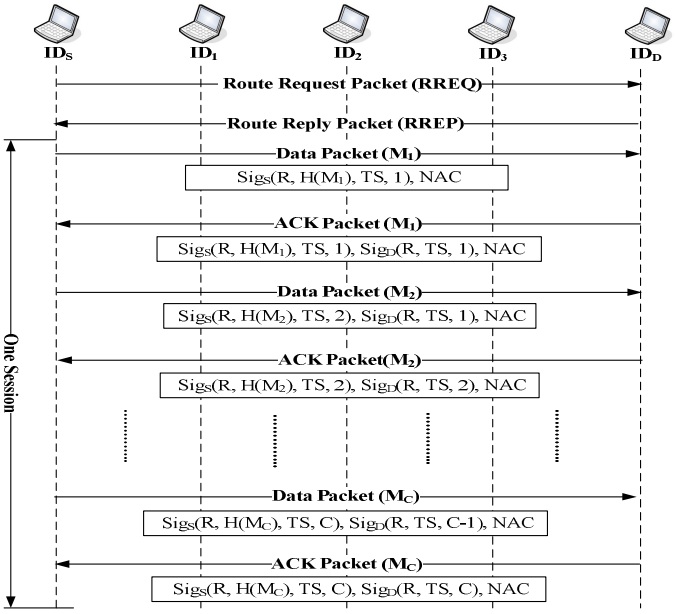


Fig. 4: The evolution of the session payment proof.

4.1 Communication Phase

4.1.1 Route Discovery

In order to establish an end-to-end session, the source node broadcasts the Route Request Packet (*RREQ*) that contains the identities of the source (ID_S) and the destination (ID_D) nodes, and the session establishment time stamp (TS). A network node appends its identity and broadcasts the packet if the time stamp is within a proper range. When the *RREQ* packet reaches the destination node, it unicasts the Route Reply Packet (*RREP*) that contains the session nodes' identities (e.g., R = ID_S, ID_1, ID_2, ID_3, ID_D in the session shown in Fig. 4), its certificate, and its signature $Sig_D(R, TS)$ for authentication and payment non-repudiation. After receiving the *RREP* packet, an intermediate node adds its certificate, signs the packet's signature to authenticate itself, and relays the packet. The source

node receives the *RREP* packet containing the session nodes' identities and authentication code (NAC = $Sig_1(Sig_2(Sig_3(Sig_D(R, TS))))$ in Fig. 4). In the first data packet, the source node appends its certificate and the NAC that is used in receipt composition.

4.1.2 Data Generation and Relay

The source node appends its signature ($Sig_S(R, H(M_C), TS, C)$) and the message (M_C) to the *C*th data packet in the session, and sends the packet to the first node on the route. The source node's signature contains the session nodes' identities (R), the message's hash value ($H(M_C)$), the number of transmitted messages (C), and the session establishment time stamp (TS). This signature is an approval from one payer to pay for C packets, and to ensure the message's authenticity and integrity. Signing $H(M_C)$ instead of M_C can reduce the receipt size because the less size $H(M_C)$ can be attached to the receipt. As illustrated in Fig. 4, upon receiving the packet, each intermediate node verifies the source node's signature and updates the session payment proof to contain the last source node's signature that is enough to prove transmitting *C* packets.

4.1.3 ACK Generation and Relay

After receiving the *C*th data packet, the destination node sends back signed ACK containing its signature ($Sig_D(R, TS, C)$) as approval to pay for *C* delivered messages. $H(M_C)$ is not included in the signature to avoid increasing the receipt size by attaching both $H(M_C)$ and $H(M_{C-1})$ when the session is broken before receiving the ACK packet. Fig. 4 shows that after receiving the ACK, intermediate nodes update the session payment proof to contain the latest destination node's signature. It can be seen that the payment proof always contains the latest received signatures from the source and the destination nodes.

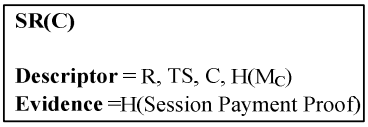


Fig. 5: The format of a session receipt for *C* packets.

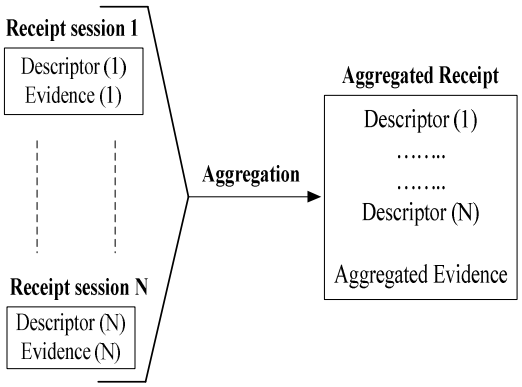


Fig. 6: Receipt aggregation technique.

4.2 Receipt Submission Phase

For each session, one receipt containing the payment data for all the intermediate nodes can be composed. It can be seen in Fig. 5 that a session receipt contains two main parts: *Descriptor* and *Evidence*. The *Descriptor* contains the payment data, i.e., the identities of the payers and the payees, the messages' number, and the session time stamp. The *Evidence* is a security token that prevents payment repudiation and manipulation and thus ensures that the receipt is non-deniable, non-modifiable, and non-forgeable. The *Evidence* is composed by hashing the session payment proof. Attaching the hash of the payment proof instead of the payment proof can reduce the receipt's size significantly. Fig. 6 shows that different receipts can be aggregated together to a smaller-size aggregated receipt. The aggregated receipt contains the descriptors of the individual receipts and *Aggregated Evidence*, where Descriptor(i) and Evidence(i) refer to the *Descriptor* and the *Evidence* of receipt number i, respectively. The *Aggregated Evidence* is computed by onion hashing the individual receipts' evidences, i.e., H(H(….H(H(Evidence(1), Evidence(2)), Evidence(3)), …), Evidence(N)). The onion-hashing technique enables the nodes to aggregate a newly issued receipt with old aggregated receipts, i.e., receipts are always stored in aggregated format, which can reduce the required storage area to store the receipts.

Since the communication sessions may occur without involving an infrastructure, the intermediate nodes have to submit the receipts to the AC for redemption. It is sufficient to submit one copy of the receipt because it contains payment data for all the intermediate nodes. However, it is not secure to trust one node to submit the receipt because it may collude with the communicating nodes so as not to submit the receipt to increase their welfare. Fig. 7 shows the nodes' charges and rewards for P_t delivered packets. If the communicating nodes collude with κ intermediate nodes and the receipt is not submitted, the colluding nodes can save $P_t \times \lambda \times (n-\kappa)$ credits. Obviously, colluders can achieve gains when $\kappa < n$, and thus payers can compensate the colluding intermediate nodes. In this section, we present a reactive receipt submission mechanism to protect against collusion attacks by submitting few redundant receipts.

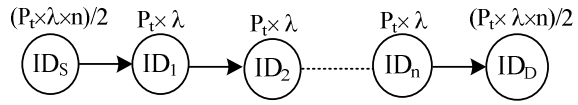


Fig. 7: The charges and rewards for P_t packets.

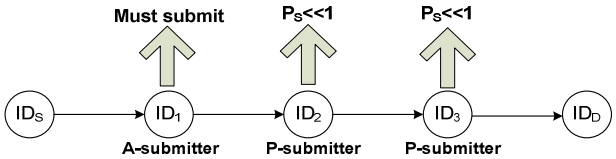


Fig. 8: Reactive receipt submission mechanism.

4.2.1 Basic Reactive Receipt Submission Mechanism

In Fig. 8, the first node after the source node (ID_1) is assigned to submit the session receipt and accused of collusion if

it does not submit it, so this node is called assigned submitter or *A-submitter*. To detect the colluding *A-submitter*, the other intermediate nodes submit the receipt probabilistically, so they are called probabilistic submitters or *P-submitters*. The *P-submitters*' receipt submission probability (P_S) is small so that *the colluding A-submitters can be identified by submitting small number of redundant receipts*. A colluding *A-submitter* can be identified once *P-submitter* submits the receipt. Equation (1) gives the probability that at least one *P-submitter* submits the receipt for a session of n intermediate nodes and n_C colluding *P-submitters*, and (2) gives the probability of identifying a colluding *A-submitter* after un-submitting γ receipts, or the probability that at least one *P-submitter* submits the receipt in γ sessions.

$$P = 1 - (1 - P_S)^{n - n_C - 1} \quad (1)$$

$$P_C(\gamma) = \sum_{k=1}^{\gamma} \binom{\gamma}{k} \cdot P^k \cdot (1 - P)^{\gamma - k} \quad (2)$$

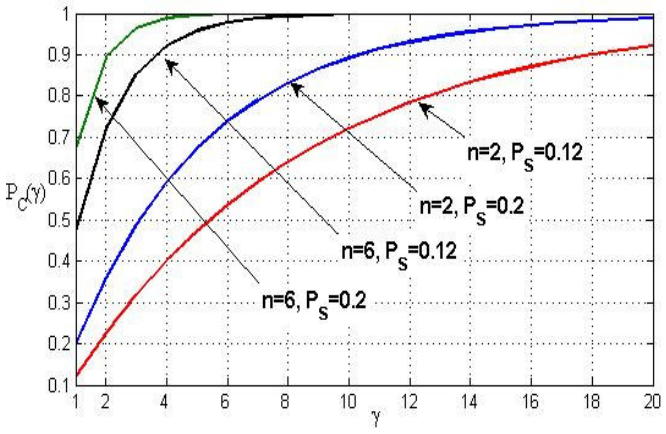


Fig. 9: $P_C(\gamma)$ Vs. γ with $n_C = 0$.

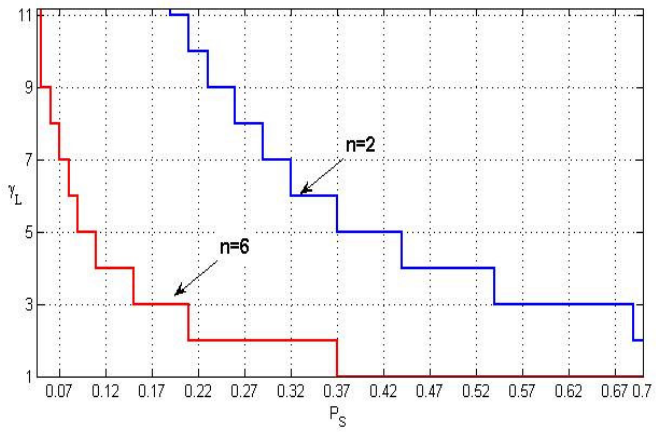


Fig. 10: γ_L Vs. P_S.

We define colluder's lifetime (γ_L) as the number of un-submitted receipts for the probability of identifying the colluding *A-submitter* to be 0.9. Collusion resistance or the immunity level to collusion attack can be measured by the colluder's lifetime, and the overhead can be measured by the number of redundant receipts which is proportional to P_S. Fig. 9 can clarify *an intuitive tradeoff between collusion resistance and overhead, and P_S can control this tradeoff*. For example, the colluding *A-submitter* can be identified in shorter time (or fewer γ) with the increase of P_S from 0.12 to 0.2, which implies the increase of the redundant receipts' number. *The receipt submission probability can be determined to achieve a specific colluder's lifetime, and thus P_S can limit the colluders' gains or the number of un-paid sessions*. For example, when n is two in Fig. 9, the colluder's lifetime can be 10 or 18 by P_S to be 0.12 or 0.2, respectively. Fig. 10 shows that choosing proper value for P_S can reduce the overhead and achieve the same γ_L. For example, when n is two, the increase of P_S from 0.37 to 0.44 increases the number of redundant receipts but does not improve γ_L.

In addition to collusion with an *A-submitter*, the communicating nodes may collude with some *P-submitters* to protect the *A-submitter*. The impact of this collusion is the extension of the *A-submitter*'s lifetime. The effect of collusion with *P-submitters* on the detection probability of the colluding *A-submitter* is shown in Fig. 11. It can be seen that when n_C is one, the colluding *A-submitter*'s lifetime is 7 un-submitted receipts, but when n_C is three or only one *P-submitter* is not colluder, the colluding *A-submitter*'s lifetime increases to 22 un-submitted receipts. Therefore, *the value of the P_S has to be determined to achieve reasonable worst-case colluder's lifetime when only one P-submitter is not colluder.*

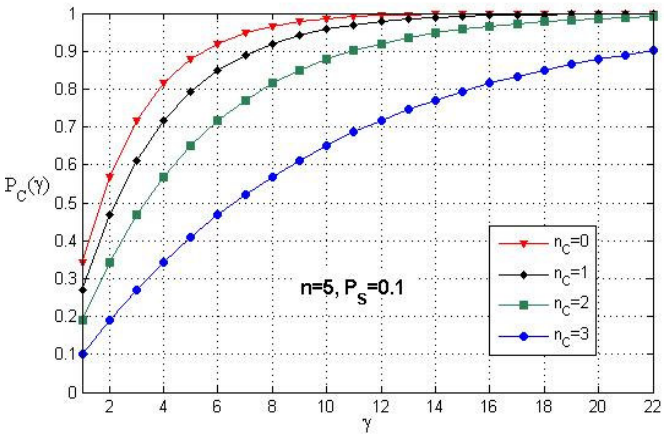


Fig. 11: The effect of colluding *P-submitters*.

4.2.2 Enhanced Reactive Receipt Submission Mechanism

In this section, we discuss two simple modifications to improve our reactive receipt submission mechanism. Fig. 9 shows that the colluder's lifetime depends on the number of intermediate nodes, e.g., at P_S to be 0.12, the colluder's lifetimes are 18 and 3.6 un-submitted receipts for n of two and six, respectively. The mechanism can have close $P_C(\gamma)$ regardless of n by making the receipt-submission probability function of n, which we call dynamic P_S, i.e., P_S should be larger for small n and smaller for large n. One way to implement dynamic P_S is to determine P_S to fix the colluding *A-*

submitter's lifetime, e.g., in Fig. 10, if γ_L of nine is desired, P_S should be 0.05 and 0.23 for n of six and two, respectively. Fig. 12 shows the relation between $P_C(\gamma)$ and γ at different n and with dynamic P_S. Comparing Fig. 9 with Fig. 12, dynamic P_S can make $P_C(\gamma_L)$ almost identical for different n.

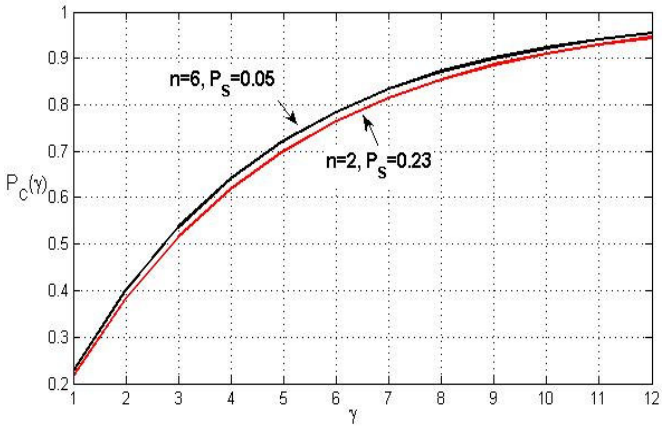


Fig. 12: $P_C(\gamma)$ Vs. γ with dynamic P_S.

$$P = 1 - \prod_{i=2}^{n}(1 - P_S(i)) \quad (3)$$

In order to identify the colluding *A-submitter*, it is sufficient that one *P-submitter* submits the receipt. However, in the basic reactive receipt submission mechanism, a receipt may be submitted by more than one *P-submitter*, so *if this receipt submission overlapping is reduced, the mechanism can be more robust against collusion attack for the same overhead (or the summation of the P-submitters' receipt submission probabilities)*. Nevertheless, it is not secure to trust one *P-submitter* because it may collude with the communicating nodes so as not to submit the receipt. One way to reduce the probability of submitting a receipt by more than one *P-submitter* is by using weighed P_S technique, or one *P-submitter* called main *P-submitter* has higher P_S than the other *P-submitters*. Equation (3) can prove that the weighed P_S technique can increase the probability of submitting a receipt without increasing the overhead or $\sum_{i=2}^{i=n} P_S(i)$, where $P_S(i)$ is the receipt submission probability for the *P-submitter* number i on the session. To assign the main *P-submitter*, a public function called selector function can be used. The input of the function is the session's unique identifier that contains the identities of the nodes on the route and the session establishment time stamp (R, TS). The selector function returns the position of the main *P-submitter* on the route (n_S), where $n_S \in \{2, 3, .., n\}$ for a session with n intermediate nodes. Obviously, changing the function input can change the position of the main *P-submitter*, which increases the difficulty of colluding with the main *P-submitter*. The selector function can be implemented by a hash function such as SHA-1 [41] with deriving n_S from its output.

To implement the weighted P_S technique, the P_S of the non-main *P-submitters* is P_{Smin} that can be determined to restrict the colluder's lifetime to γ_{Lmax} in the worst collusion attack, i.e., when only one *P-submitter* is not colluder. Moreover, the main *P-submitter* adjusts its P_S to achieve a colluder's lifetime of γ_{Lmin}, but P_S should not be less than P_{Smin}. In this way, if all the *P-submitters* are honest, the colluding *A-submitter*'s lifetime is γ_{Lmin}, but it may be extended up to γ_{Lmax} due to the collusion with the *P-submitters*. *The rationale here is that since collusion with one node is more likely or easier than collusion with two or more nodes, weighted P_S technique can improve the robustness against collusion attacks with high probability*. In Fig. 13, each *P-submitter*'s P_S is 0.12 in the equal P_S technique, and P_S is 0.45 and 0.01 for the main *P-submitter* and the other *P-submitters* in the weighted P_S technique, respectively. It can be seen that by using the weighted P_S technique, the colluding *A-submitter* can be identified after un-submitting smaller number of receipts for the same $\sum_{i=2}^{i=n} P_S(i)$ due to decreasing the probability of submitting a receipt by more than one *P-submitter*.

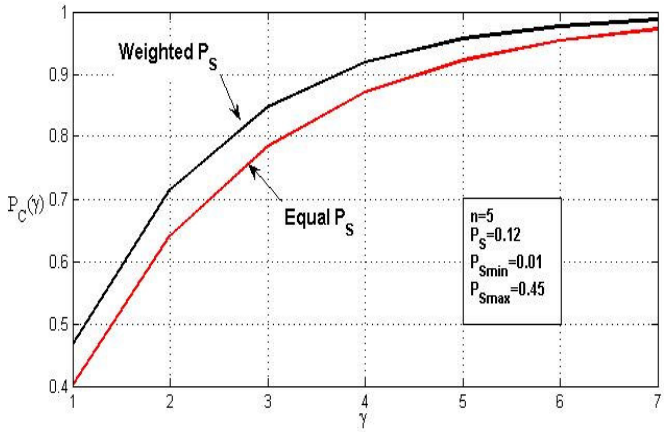


Fig. 13: $P_C(\gamma)$ Vs. γ in weighted and equal P_S techniques.

4.3 Payment Redemption and Colluder Identification Phase

The network nodes periodically submit the receipts to the AC to redeem them. Once the AC receives a receipt, it first checks if the receipt has been deposited before using its unique identifier (R, TS). Then, to verify the receipt's credibility, the AC generates the session payment proof and hashes it, and compares the resultant hash value with the receipt's *Evidence*. If the *A-submitter* does not submit the receipt but *P-submitter* does, the *A-submitter* is identified as colluder and excluded from the network by denying update its certificate. Finally, the AC clears the receipt according to the rewarding and charging policy discussed in Section 3.3.2.

5. Security Analysis

Our security objective is to prevent misbehaving nodes from achieving gains such as stealing credits or paying less. In our incentive system, the charges and rewards are based on receipts submitted by rational nodes, so a node or even a group of colluding nodes may attempt to cheat the system to increase their welfare. For *Double Clearance* attack, the attacker attempts to clear a receipt multiple times to increase its rewards. The AC can thwart the attack and identify the attacker because each receipt has unique identifier. For *Double Spending* attack, the attackers attempt to generate identical

receipts for different sessions to pay once. In our incentive system, even if attackers establish different sessions at the same time, the receipts' identifiers are different because at least one intermediate node is different. For *Receipt Forgery or Manipulation* attack, the attackers attempt to forge receipts or manipulate valid receipts to increase their rewards. This is almost impossible in our incentive system due to the difficulties of forging or modifying the payers' signatures, computing the private keys from the public ones, and computing the hash of the signatures without computing the signatures. Moreover, if an attacker attaches a random value for a receipt's *Evidence*, the probability to hit the correct value is extremely low, e.g., this probability is 6.84×10^{-49} by using SHA-1 [41] with digest value of 20 bytes, the AC can identify the attackers because their receipts' verifications fail.

For *Free Calling (or Riding)* attacks, the attackers attempt to communicate freely. Two colluding intermediate nodes on a legitimate session may manipulate the session packets to add their data. If the intermediate nodes cannot verify the payment data, the source node may transmit packets with invalid payment data. 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 names of others. For *Message and Payment Repudiation* attacks, the attacker attempts to deny initiating a session or the payment so as not to pay. To thwart these attacks, the communicating nodes' signatures can prevent the denial and the manipulation of the messages and the payment. The intermediate nodes verify the communicating nodes' signatures to verify the message integrity and authenticity, and the payment data. Moreover, a *RREQ* packet is dropped if the time stamp is not within a proper range to thwart *Packet-Replay* attack.

For *Fake A-submitter* attack, the communicating nodes attempt to insert non-existing *A-submitter* on the route to communicate freely because the fake *A-submitter* does not submit the receipt, also this *A-submitter* may be accused of collusion. To thwart this attack and other attacks [42] outside the scope of this work, the session nodes authenticate themselves in the route discovery phase and the nodes' authentication code (NAC) is included in the receipt's *Evidence*. For *Credit Collecting* attack, some nodes may insert non-existent neighbors to collect credits for them without participation in packet relay. This attack is a type of the known routing attack called *Route Lengthening*. First, the colluding nodes have to exchange their private keys because authentication is needed in our incentive system, which may discourage the attack because colluders can steal the credits of each other or commit malicious actions under their names. Second, the attack does not always work because it may lead to sub-optimal route due to the preference of shortest routes. Third, the AC can identify the attackers when it observes that some nodes appear in different locations at the same time. Finally, the proposed solutions for secure routing protocols such as ARAN [43] and Ariadne [44] can be implemented in our incentive system. For *Destination Node's Robbery* attack, the source node colludes with some intermediate nodes to steal credits from the destination node by sending bogus data to the destination. In our incentive system, the intermediate nodes are rewarded only when the destination node acknowledges receiving correct data, and a receipt cannot be composed if the destination node is not interested in the communication because its signature is required in the receipt composition.

For our reactive receipt submission mechanism, the colluding *A-submitter* can be identified once *P-submitter* submits the receipt, and colluders cannot know whether *P-submitter* is going to submit the receipt or not because *P-submitters* decide submitting the receipts independently. Moreover, the receipt submission probability can restrict the number of un-submitted receipts or colluders' gains. The AC can identify the colluding or un-cooperative *P-submitters* that do not submit the receipts by comparing their receipt submission ratios with P_S. For *Reduced Payment Receipt* attack, the colluding *A-submitter* submit the session receipt but with less payment. The AC can identify the colluding *A-submitter* by matching the payment in its receipt with that in the *P-submitter*'s receipt. In this case, the *P-submitter* cannot be attacker because it is difficult to manipulate the receipt's *Evidence* to increase the payment.

In our payment model, the communicating nodes can communicate even if they do not have sufficient credits, so to limit overspending, certificates' lifetime is short and the lifetime can depend on the node's available credits and its average credit consumption rate. As the network nodes are rational, without proper charging and rewarding policy, they may try to cheat to increase their welfare. Our charging and rewarding policy has been developed to counteract rational cheating actions and encourage the nodes' cooperation. Particularly, a rational node can exhibit one of the following actions:

1) To increase their rewards with consuming low resources, a node may compose the receipt but does not forward the message, or a group of colluding nodes may forward only the receipt instead of the message because submitting a receipt to the AC is sufficient for earning credits. In our payment model, the nodes are motivated to relay the messages because they are rewarded only when the destination node acknowledges receiving the messages.
2) The destination node receives a message but it does not send ACK so as not to pay. To prevent this, both the source and the destination nodes are charged for un-delivered messages.
3) The *A-submitter* colludes with the communicating nodes and claims that a message does not reach the destination node to increase their welfare. In our payment model, the communicating nodes are charged for un-delivered messages.

6. Performance Evaluation

Using public key cryptography for cooperation stimulation is necessary to prevent the communicating nodes from denying the payment, and to enable the intermediate nodes to verify the payment. Digital signature technology and hardware implementation have improved and fast signature schemes are currently available. For example, "online/offline" digital signature [45] is computed in two steps: an off-line step that is computationally more demanding and independent of the mes-

sage is performed before the message to be signed is available; and a lightweight on-line step is performed once the message to be signed becomes available. Moreover, FPGA implementation of the RSA signature scheme can perform the signing and verifying operations in several milliseconds [46]. In addition, instead of generating an ACK per message, ACK can be generated for a number of messages to reduce the number of digital signature operations, and to reduce the end-to-end delay, the destination node can generate its signature before receiving a message because the message is not included in its signature. Moreover, the end-to-end delay can be reduced by delayed verification where a node forwards the packet before verifying the signature.

Due to the high frequency of low-value transactions, reducing the receipts' number is essential for practical implementation of an incentive system to avoid making bottleneck in the AC, and to reduce storage, submission, and process overheads. In this section, simulations are performed to evaluate the expected reduction of the receipts' number in our incentive system comparing with the existing systems.

TABLE III
STATISTICS OF THE SIMULATED NETWORK.

Average Network Connectivity	$P(R_L \leq 4)$	$P(4 < R_L \leq 6)$	$P(6 < R_L \leq 8)$	$P(8 < R_L \leq 10)$	$P(R_L > 10)$
0.888	0.559	0.297	0.118	0.023	0.0041

6.1 Simulation Setup

MATLAB is used to simulate multi-hop wireless network by randomly deploying 35 mobile nodes with 125 m transmission range in a square cell of 1000 m × 1000 m. The constant bit rate (CBR) traffic source is implemented in each node as an application layer, and the source and destination pairs are chosen randomly. To emulate node mobility, we adopt the modified random waypoint model [47]. 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. A node's speed is uniformly distributed from the range [0, 3] m/s and the pause time is 20 s. We simulate the Ad hoc On-Demand Distance Vector (AODV) routing protocol [36] over an ideal and contention-free channel, i.e., all the nodes within transmission range receive packet transmission correctly. The time stamp (TS), node's identity (ID_i), and message number (C) are five, four, and two bytes, respectively. The simulation results are averaged over 400 runs. MATLAB is used instead of network simulator such as NS2 because the intention is to compare the receipts' overhead of our incentive system with the existing systems. The effect of the un-simulated models such as non-ideal channel, channel contention, node buffer, etc, should be the same on all the systems.

In Table III, statistics about route length and connectivity in our simulated network are given. $P(R_L \leq 4)$ is the probability that a route has four nodes or fewer including the source and the destination nodes. The network connectivity is the ratio of the connected routes to the total number of possible routes assuming any two nodes are source and destination pair. The statistics show that our simulated network is well connected and the route length is acceptable.

TABLE IV
AVERAGE RECEIPT SIZE (BYTES)

Sprite	Express	DSC	PIS
294	196	98	60

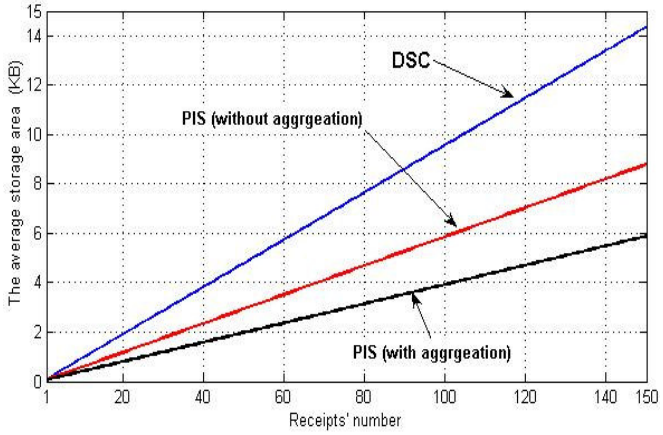


Fig. 14: The effectiveness of the receipt aggregation technique.

6.2 Simulation Results

6.2.1 Average Receipt Size

Using 1024-bit RSA signature scheme and SHA-1 hash function with digest width of 20 bytes [41], the average receipt size is given in Table IV. It can be seen that the receipt sizes in DSC and our incentive system are much smaller than that in Sprite and Express due to hashing the signatures. The receipt size in DSC is larger than that in PIS due to attaching the root and the last released hash value of the hash chain. For PIS and DSC, 1MB storage area can store up to 17,476.27 and 10,699.76 receipts, respectively.

6.2.2 Effectiveness of Receipt Aggregation Technique

Fig. 14 gives the relation between the receipts' number and their average storage area. Without receipt aggregation, PIS requires less storage area than DSC due to reducing the receipt size, and the receipt aggregation technique can reduce the storage area effectively, e.g., 150 receipts require average storage areas of 14.36, 8.79 and 5.8 Kbytes in DSC and PIS without and with receipt aggregation, respectively.

6.2.3 Number of Generated Receipts

Table V gives the number of generated receipts for ten-minute data transmission at different packet transmission rates and node speed. During the transmission, a new session is established each time the route is broken. It can be seen that Sprite and Express generate a large number of receipts due to generating a receipt per packet, and the increase of packet transmission rate increases the receipts' number significantly due to increasing the number of transmitted packets. Moreover, there is no effect to the nodes' speed on the receipts' number because receipts are generated for transmitted packets regardless whether they reach the destinations or not.

For DSC, a receipt is generated when a route is broken or S packets are transmitted, where S is the hash chain size, but a receipt is generated only when a route is broken in PIS. Table V indicates that more receipts are generated at high node mobility in DSC and PIS because the routes are more frequently broken, i.e., the data are transmitted over larger number of routes. *In PIS, the packet transmission rate has little effect on the receipts' number because one fixed-size receipt is generated per session (or route) regardless of the transmitted packets' number*.

For DSC, fewer receipts may be generated with the increase of S because a receipt can contain payment data for more packets. The increase of S from 11 to 22 can halve the receipts' number, but the increase from 22 to 33 reduces the receipts' number less because the routes are broken before releasing all the elements in the hash chain. When a route is broken, the unused hash values of the hash chain should not be used for other routes to secure the payment. Moreover, it is difficult to compute optimal value for S due to the difficulty of estimating the number of transmitted packets before the route is broken. *Consequently, to reduce the receipts' number in DSC, each node has to compute and store a large number of long hash chains*. In other words, DSC can reduce the ACK packets' processing overhead by replacing the destination's signatures with hashing operations but on the expense of increasing the receipts' size and number. Certainly, more receipts are generated with considering the effect of the non-ideal channel but our incentive system still generates smaller number of receipts comparing with the existing systems because the non-ideal channel has similar effect on all the systems.

TABLE V
THE AVERAGE RECEIPTS' NUMBER FOR DATA TRANSMISSION FOR TEN MINUTES.

Pkts/s / Speed		0.5	2	4
[0, 3] m/s	Sprite and Express	300	1200	2400
	DSC (S = 11)	28.8	111.5	220.4
	DSC (S = 22)	15.4	56.8	111.1
	DSC (S = 33)	11.2	39.3	74.7
	PIS	4.2	5.2	4.1
[0, 20] m/s	Sprite and Express	300	1200	2400
	DSC (S = 11)	29.9	117.5	233.4
	DSC (S = 22)	24	59.6	118.2
	DSC (S = 33)	22.7	51.5	84.6
	PIS	21.6	20.9	22.6

6.2.4 Number of Submitted Receipts

To compare our receipt submission mechanism with the existing ones, we assume that the receipts' number is 100 and the number of intermediate nodes is five. For the *Preventive* mechanism, the number of submitted receipts by each node is 59 to secure the mechanism up to two colluders, i.e., to guarantee that the probability of submitting at least 90% of the receipts is at least 0.9 under two colluders. For the *Reactive* mechanism, P_S is 0.038 and 0.075 to guarantee that $\gamma_L \leq 20$ and $\gamma_L \leq 10$ in case of two colluders, respectively. Moreover, in our evaluation, we consider the following two metrics. The security metric (Q) is the robustness against collusion attack measured by the number of submitted receipts under collusion attack, which is related to the colluder's lifetime in PIS. The efficiency metric (T) is the number of generated receipts to the submitted ones in normal (no collusion) case, where $T \in [0, 1]$. The optimal value for T is one when there is no submission to redundant receipts. From (4), the value of T in PIS depends on the receipt submission probability and the number of intermediate nodes. Fig. 15 shows that the increase of P_S degrades the efficiency in our receipt submission mechanism but decreases the colluding *A-submitter*'s lifetime as indicated in Fig. 10. Proper value for P_S can reduce the number of redundant receipts and restrict the colluders' gains. The value of P_S should depend on the likelihood or the easiness of attacking the incentive system, e.g., the easiness of obtaining multiple identities and compromising a device. In addition, the AC can change the value of P_S periodically according to the security situation in the network, e.g., P_S can be increased when discovering many collusion attacks in the network.

$$T = \frac{1}{1+P_S \cdot (n-1)} \tag{4}$$

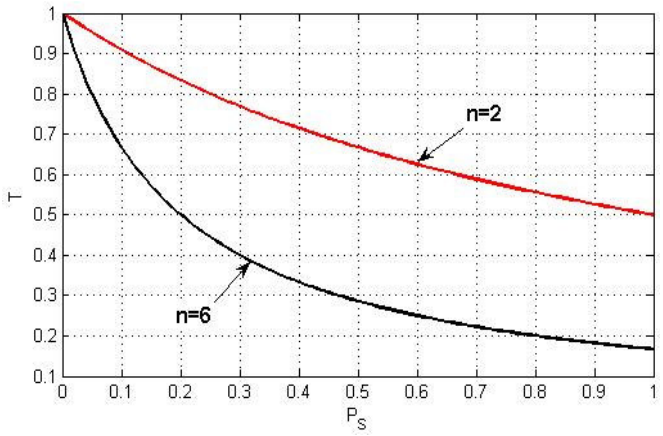


Fig. 15: T Vs. P_S.

From Table VI, *One-Submitter* mechanism can achieve the highest efficiency (T = 1) because each receipt is submitted once, i.e., there is no redundant receipts, but the mechanism is vulnerable to collusion attack because if one node colludes (Co = 1), all the receipts are not submitted (Q = 0) without identifying the colluding nodes. *All-Submitter* mechanism is not vulnerable to collusion attacks but it is not efficient because six and seven copies of each receipt are submitted in Sprite and Express, respectively.

For *Preventive* mechanism, if all the nodes are honest (Co = 0), the probability to submit 97 receipts is 0.97, so *in Reactive mechanism, receipts are not submitted in exceptional cases (collusion), but in Preventive mechanism, the un-chosen receipts are not submitted normally in the dominant non-collusion condition*. In *Preventive* mechanism, it may be difficult to identify the colluders that reduce the number of unrepeated receipts by submitting the same receipts. At three colluders, the probability to submit at least 78 receipts is 0.93 but the colluder's lifetime is 30 and 15 for P_S to be 0.038 and

0.075, respectively, so the *Preventive* mechanism can protect the network effectively when collusion attacks are very common and extensive, which contradicts property P6 in Table I. For efficiency, *Reactive* mechanism can reduce the redundant receipts' number significantly comparing with *Preventive* mechanism. Only 15.2 and 37.5 redundant receipts are submitted for P_S to be 0.038 and 0.075, respectively, but for *Preventive* mechanism, 195 redundant receipts are submitted because *each node has to submit a large number of receipts to guarantee submitting most of the receipts in case of no collusion*.

TABLE VI
EVALUATION OF THE RECEIPT SUBMISSION MECHANISMS.

Mechanism		Security (Q)				Redundant Receipts	T
		Co = 0	Co = 1	Co = 2	Co = 3		
***One-Submitter* [34]**		100	0	0	0	0	1
All-Submitters	**Sprite, INPAC**	100	100	100	100	500	0.17
	Express	100	100	100	100	600	0.143
***Preventive* [21]**		$P(Q\geq 97) = 0.97$	$P(Q\geq 95) = 0.94$	$P(Q\geq 90) = 0.92$	$P(Q\geq 78) = 0.93$	195	0.33
Reactive	**$P_S = 0.038$**	100	γ_L=15	γ_L=20	γ_L=30	15.2	0.87
	$P_S = 0.075$	100	γ_L=8	γ_L=10	γ_L=15	37.5	0.72

7. CONCLUSION

In this paper, we have proposed an incentive system to stimulate the nodes' cooperation in multi-hop wireless networks. The payment model has been developed to implement micro-payment for cooperation stimulation efficiently. Reducing the overhead of the payment receipts is necessary for practical implementation of an incentive system due to the high frequency of low-value transactions. Therefore, one fixed-size receipt is generated per session regardless of the packets' number. Attaching the hash of the signatures instead of the signatures can reduce the receipt size significantly, and receipt aggregation technique has been used to generate a smaller-size receipt for multiple sessions. In addition, reactive receipt submission mechanism has been proposed to reduce the number of submitted receipts and protect against collusion attacks by small number of redundant receipts with limiting the colluders' gains probabilistically. Our analysis and simulations demonstrate that the proposed incentive system can secure the payment, and significantly reduce the receipts' storage area and the number of generated and submitted receipts.

REFERENCES

[1] G. Shen, J. Liu, D. Wang, J. Wang, and S. Jin, "Multi-hop relay for next-generation wireless access networks", *Bell Labs Technical Journal*, vol. 13, no. 4, pp. 175-193, 2009.

[2] X. Li, B. Seet, and P. Chong, "Multihop cellular networks: technology and economics", *Computer Networks*, vol. 52, no. 9, pp. 1825–1837, June 2008.

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

[4] P. Gupta and P. Kumar, "The capacity of wireless networks", *IEEE Trans. on Information Theory*, Vol. 46, No. 2, pp. 388-404, March 2000.

[5] 3rd Generation Partnership Project, Technical Specification Group Radio Access Network, "Opportunity driven multiple access", *3G Technical Report* 25.924 version 1.0.0, December 1999.

[6] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks", *Proc. of ACM Mobile Computing and Networking (MobiCom'00)*, pp. 255–265, Boston, Massachusetts, USA, August 6-11, 2000.

[7] P. Michiardi and R. Molva, "Simulation-based analysis of security exposures in mobile ad hoc networks", *Proc. of European Wireless*, Florence, Italy, February 25–28, 2002.

[8] J. Hu, "Cooperation in mobile ad hoc networks", *Technical report* (TR-050111), Computer Science Department, Florida State University, Tallahassee, January 2005.

[9] G. Marias, P. Georgiadis, D. Flitzanis, and K. Mandalas, "Cooperation enforcement schemes for MANETs: A survey", *Wireless Communications and Mobile Computing*, vol. 6, issue 3, pp. 319–332, 2006.

[10] S. Bansal and M. Baker, "Observation-based cooperation enforcement in ad-hoc networks", *Techical Report*, Computer Science Department, Stanford University, CA, USA, July 2003.

[11] Q. He, D. Wu, and P. Khosla, "A secure incentive architecture for ad-hoc networks", *Wireless Communications and Mobile Computing*, vol. 6, no. 3, pp. 333-346, May 2006.

[12] L. Feeney, "An energy-consumption model for performance analysis of routing protocols for mobile ad hoc networks", *Mobile Networks and Applications*, vol. 3, no. 6, pp. 239–249, 2001.

[13] A. Spyropoulos and C. Raghavendra, "Energy efficient communications in ad hoc networks using directional antennas", *Proc. of IEEE INFOCOM'02*, New York, USA, June 2002.

[14] F. Milan, J. Jaramillo, and R. Srikant, "Achieving cooperation in multihop wireless networks of selfish nodes", *Proc. of workshop on Game Theory for Communications and Networks*, Pisa, Italy, October 14, 2006.

[15] K. Wang, M. Wu, W. Lu, P. Xia, and S. Shen, "An incentive mechanism for charging scheme in heterogeneous collaborative networks", *Proc. of IEEE CSCWD*, pp. 559-564, Xi'an, China, April 16-18, 2008.

[16] M. Peirce and D. O'Mahony, "Micropayments for mobile networks", *Technical Report of the Department of Computer Science*, Trinity College, Dublin, Ireland, 1999.

[17] S. Micali and R. Rivest, "Micropayments revisited", Topics in Cryptology — CT-RSA 2002, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, vol. 2271, pp. 171-203, 2002.

[18] C. Gentry and Z. Ramzan, "Microcredits for verifiable foreign service provider metering", Financial Cryptography, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, vol. 3110, pp. 9-23, 2004.

[19] I. Papaefstathiou and C. Manifavas, "Evaluation of micropayment transaction costs", *Electronic Commerce Research*, vol. 5, no. 2, pp. 99-113, 2004.

[20] J. Palmer and L. Eriksen, "Digital newspapers explore marketing on the Internet", *ACM Communications*, vol. 42, no. 9, pp. 33-40, 1999.

[21] M. Mahmoud and X. Shen, "DSC: Cooperation incentive mechanism for multi-hop cellular networks", *Proc. of IEEE ICC'09*, Germany, June 14-18, 2009.

[22] L. Buttyan and J. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks", *Mobile Networks and Applications*, vol. 8, no. 5, pp. 579-592, October 2004.

[23] Y. Zhang, W. Lou, and Y. Fang, "A secure incentive protocol for mobile ad hoc networks", *ACM Wireless Networks*, vol. 13, no. 5, pp. 569-582, October 2007.

[24] A. Weyland and T. Braun, "Cooperation and accounting strategy for multi-hop cellular networks", *Proc. of IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*, pp. 193-198, Mill Valley, CA, USA, April 25-28, 2004.

[25] A. Weyland, "Cooperation and accounting in multi-hop cellular networks", Ph.D. thesis, University of Bern, November 2005.

[26] A. Weyland, T. Staub, and T. Braun, "Comparison of motivation-based cooperation mechanisms for hybrid wireless networks", *Computer Communications*, vol. 29, pp. 2661–2670, 2006.

[27] J. Pan, L. Cai, X. Shen, and J. Mark, "Identity-based secure collaboration in wireless ad hoc networks", *Computer Networks* (Elsevier), vol. 51, no. 3, pp. 853-865, 2007.

[28] M. Jakobsson, J. Hubaux, and L. Buttyan, "A micro-payment scheme encouraging collaboration in multi-hop cellular networks", *Proc. of the 7th Financial Cryptography (FC'03)*, vol. 2742, pp. 15–33, La Guadeloupe, January 2003.

[29] M. Mahmoud and X. Shen, "Stimulating cooperation in multi-hop wireless networks using cheating detection system", *Proc. of IEEE INFOCOM*, San Diego, California, USA, March 14-19, 2010.
[30] N. Salem, L. Buttyan, J. Hubaux, and M. Jakobsson, "Node cooperation in hybrid ad hoc networks", *IEEE Trans. on Mobile Computing*, vol. 5, no. 4, pp. 365-376, April 2006.
[31] 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, USA, March 30-April 3, 2003.
[32] T. Chen and S. Zhong, "INPAC: An enforceable incentive scheme for wireless networks using network coding", Proc. of IEEE INFOCOM, San Diego, CA, USA, 14-19 March 2010.
[33] H. Janzadeh, K. Fayazbakhsh, M. Dehghan, and M. Fallah, "A secure credit-based cooperation stimulating mechanism for MANETs using hash chains", *Future Generation Computer Systems*, vol. 25, issue 8, pp. 926-934, September 2009.
[34] B. Lamparter, K. Paul, and D. Westhoff, "Charging support for ad hoc stub networks", *Computer Communications*, vol. 26, no. 13, pp. 1504–1514, 2003.
[35] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks", *Mobile Computing*, Chapter 5, Kluwer Academic Publishers, pp. 153-181, 1996.
[36] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing", *Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, USA, pp. 90–100, February 1999.
[37] H. Pagnia and F. Gartner, "On the impossibility of fair exchange without a trusted third party", *Technical Report TUD-BS-1999-02*, Darmstadt University of Technology, March 1999.
[38] J. Hubaux, L. Buttyán, and S. Capkun, "The quest for security in mobile ad hoc networks", *ACM Symposium on Mobile Ad Hoc Networking and Computing*, October 2001.
[39] S. Zhong, L. Li, Y. G. Liu, and Y. R. Yang, "On designing incentive compatible routing and forwarding protocols in wireless ad-hoc networks", *Proc. of ACM MobiCom*, pp. 117–131, New York, NY, USA, August 2005.
[40] L. Anderegg and S. Eidenbenz, "Ad Hoc-VCG: A trustful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents", *Proc. of ACM MobiCom*, San Diego, CA, USA, September 2003.
[41] A. Menzies, P. Oorschot, and S. Vanstone, "Handbook of applied cryptography", CRC Press, http://www.cacr.math.uwaterloo.ca/hac, Boca Raton, Fla., 1996.
[42] B. Wu, J. Chen, J. Wu, and M. Cardei, "A survey of attacks and countermeasures in mobile ad hoc networks", *Springer Wireless Network Security*, Network Theory and Applications, vol. 17, pp. 103-135, 2007.
[43] K. Sanzgiri, D. LaFlamme, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer, "Authenticated routing for ad hoc networks", *IEEE Selected Areas in Communications*, vol. 23, no. 3, pp. 598–610, March 2005.
[44] Y. Hu, A. Perrig, and D. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks", *Proc. of ACM MobiCom*, Atlanta, GA, USA, September 2002.
[45] S. Even, O. Goldreich, and S. Micali, "On-line/off-line digital signatures", *Advances in Cryptology--Crypto '89*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, vol. 435, pp. 263-277, 1990.
[46] O. Nibouche, M. Nibouche, A. Bouridane, and A. Belatreche, "Fast architectures for FPGA-based implementation of RSA encryption algorithm", *Proc. of IEEE Field-Programmable Technology conference*, Brisbane, Australia, December 2004.
[47] J. Yoon, M. Liu, and B. Nobles, "Sound mobility models", *Proc. of ACM MobiCom*, San Diego, CA, USA, September 2003.

Mohamed Elsalih Mahmoud received the B.Sc. and M.Sc. degrees with honors in electrical communications engineering from Banha University, Egypt, in 1998 and 2003, respectively. He received the best paper award for the Communication and Information Systems Security Symposium in the International Conference on Communications (ICC'09), Dresden, Germany, 14-18 June, 2009. He is currently working toward the Ph.D. degree with the Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is also currently a member of the Broadband Communications Research Group, University of Waterloo. His research interest includes wireless network security, privacy in hybrid ad hoc networks, and cooperation incentive mechanisms in multi-hop wireless networks.

Xuemin Shen (M'97–SM'02–F'09) received the B.Sc. degree in electrical engineering from Dalian Maritime University, Dalian, China, in 1982 and the M.Sc. and Ph.D. degrees in electrical engineering from Rutgers University, Camden, NJ, in 1987 and 1990, respectively.

He is currently a Professor and the University Research Chair with the Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is the author or a coauthor of three books and more than 400 papers and book chapters on wireless communications and networks, control, and filtering. He serves as the Editor-in Chief for Peer-to-Peer Networking and Application and an Associate Editor for Computer Networks, ACM/Wireless Networks, and Wireless Communications and Mobile Computing. He has also served as a Guest Editor for ACM Mobile Networks and Applications. His research focuses on mobility and resource management in interconnected wireless/wired networks, ultra wideband wireless communications networks, wireless network security, wireless body area networks, and vehicular ad hoc and sensor networks.

Dr. Shen is a Registered Professional Engineer in the Province of Ontario, and a Distinguished Lecturer of IEEE Communications Society. He received the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premier's Research Excellence Award in 2003 from the Province of Ontario, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. He served as the Tutorial Chair for the 2008 IEEE International Conference on Communications, the Technical Program Committee Chair for the 2007 IEEE Global Telecommunications Conference, the General Cochair for the 2007 International Conference in Communications and Networking in China and the 2006 International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, and the Founding Chair
for IEEE Communications Society Technical Committee on P2P Communications and Networking. He also serves as a Founding Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and the KICS/IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS. He has also served as a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, and the IEEE Communications Magazine.