Efficient Packet-Drop Thwarting and User-Privacy Preserving Protocols for Multi-hop Wireless Networks

by

Mohamed Mohamed Elsalih Abdelsalam Mahmoud

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2011

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In multi-hop wireless network (MWN), the mobile nodes relay others' packets for enabling new applications and enhancing the network deployment and performance. However, the selfish nodes drop the packets because packet relay consumes their resources without benefits, and the malicious nodes drop the packets to launch *Denial-of-Service* attacks. Packet drop attacks adversely degrade the network fairness and performance in terms of throughput, delay, and packet delivery ratio. Moreover, due to the nature of wireless transmission and multi-hop packet relay, the attackers can analyze the network traffic in undetectable way to learn the users' locations in number of hops and their communication activities causing a serious threat to the users' privacy. In this thesis, we propose efficient security protocols for thwarting packet drop attacks and preserving users' privacy in multi-hop wireless networks.

First, we design a fair and efficient cooperation incentive protocol to stimulate the selfish nodes to relay others' packets. The source and the destination nodes pay credits (or micropayment) to the intermediate nodes for relaying their packets. In addition to cooperation stimulation, the incentive protocol enforces fairness by rewarding credits to compensate the nodes for the consumed resources in relaying others' packets. The protocol also discourages launching *Resource-Exhaustion* attacks by sending bogus packets to exhaust the intermediate nodes' resources because the nodes pay for relaying their packets.

For fair charging policy, both the source and the destination nodes are charged when the two nodes benefit from the communication. Since micropayment protocols have been originally proposed for web-based applications, we propose a practical payment model specifically designed for MWNs to consider the significant differences between web-based applications and cooperation stimulation. Although the non-repudiation property of the public-key cryptography is essential for securing the incentive protocol, the public-key cryptography requires too complicated computations and has a long signature tag. For

efficient implementation, we use the public-key cryptography only for the first packet in a series and use the efficient hashing operations for the next packets, so that the overhead of the packet series converges to that of the hashing operations. Since a trusted party is not involved in the communication sessions, the nodes usually submit undeniable digital receipts (proofs of packet relay) to a centralized trusted party for updating their credit accounts. Instead of submitting large-size payment receipts, the nodes submit brief reports containing the alleged charges and rewards and store undeniable security evidences. The payment of the fair reports can be cleared with almost no processing overhead. For the cheating reports, the evidences are requested to identify and evict the cheating nodes. Since the cheating actions are exceptional, the proposed protocol can significantly reduce the required bandwidth and energy for submitting the payment data and clear the payment with almost no processing overhead while achieving the same security strength as the receipt-based protocols.

Second, the payment reports are processed to extract financial information to reward the cooperative nodes, and contextual information such as the broken links to build up a trust system to measure the nodes' packet-relay success ratios in terms of trust values. A node's trust value is degraded whenever it does not relay a packet and improved whenever it does. A node is identified as malicious and excluded from the network once its trust value reaches to a threshold. Using trust system is necessary to keep track of the nodes' long-term behaviors because the network packets may be dropped normally, e.g., due to mobility, or temporarily, e.g., due to network congestion, but the high frequency of packet drop is an obvious misbehavior. Then, we propose a trust-based and energy-aware routing protocol to route traffics through the highly trusted nodes having sufficient residual energy in order to establish stable routes and thus minimize the probability of route breakage. A node's trust value is a real and live measurement to the node's failure probability and mobility level, i.e., the low-mobility nodes having large hardware resources can perform packet relay more efficiently. In this way, the proposed protocol stimulates the nodes not only to cooperate but

also to improve their packet-relay success ratio and tell the truth about their residual energy to improve their trust values and thus raise their chances to participate in future routes.

Finally, we propose a privacy-preserving routing and incentive protocol for hybrid ad hoc wireless network. Micropayment is used to stimulate the nodes' cooperation without submitting payment receipts. We only use the lightweight hashing and symmetric-key-cryptography operations to preserve the users' privacy. 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. Moreover, our protocol protects the location privacy of the anonymous source and destination nodes.

Extensive analysis and simulations demonstrate that our protocols can secure the payment and trust calculation, preserve the users' privacy with acceptable overhead, and precisely identify the malicious and the cheating nodes. Moreover, the simulation and measurement results demonstrate that our routing protocols can significantly improve route stability and thus the packet delivery ratio due to stimulating the selfish nodes' cooperation, evicting the malicious nodes, and making informed decisions regarding route selection. In addition, the processing and submitting overheads of the payment-reports are incomparable with those of the receipts in the receipt-based incentive protocols. Our protocol also requires incomparable overhead to the signature-based protocols because the lightweight hashing operations dominate the nodes' operations.

Acknowledgements

I would like to express my deepest gratitude to Professor Xuemin (Sherman) Shen, my advisor. I thank you for your continuing guidance and support during my four years of research. Your sharp sense of research direction, great enthusiasm, and strong belief in the potential of this research has been a tremendous force for the completion of this work. I have learned so many things from you, including doing research, writing papers, giving seminars, and many more. Most importantly, I thank you for encouraging me in each step of my growing path. Your strong belief in me and continuous encouragement have made this research such an exciting experience that our collaboration finally produces something that we are both proud of.

This thesis would not have been possible without the assistance of many people. I would also like to express my extreme appreciation to my thesis committee members: Professor Ajit Singh, Professor Zhou Wang, Professor Xinzhi Liu, and Professor Yanchao Zhang. They have contributed their precious time to read my thesis, and provided valuable suggestions and comments that helped to improve the quality of this thesis.

I would also like to thank my colleagues and friends at Security Discussion Group of BBCR Lab. My discussions with Xiaodong Lin, Rongxing Lu, Haojin Zhu, Chenxi Zhang, Xiaohui Liang, Sanaa Taha, and Albert Wasef have given me many inspirations. I thank them all. There are many other people whose names are not mentioned here. It does not mean that I have forgotten you or your help. It is a privilege for me to work and share life with so many bright and energetic people. Your talent and friendship have made Waterloo such a great place to live.

I would never get this far without the support of my parents, wife, and kids. Thank you for always believing in me and supporting me. Your love and encouragement have been and will always be a great source of inspiration in my life.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AREQ	Authentication Request
AREP	Authentication Reply
BAR	Best Available Route
BS	Base Station
DoS	Denial of Service
DREST	Downlink Route Establishment packet
DRREQ	Downlink Route Request packet
MWN	Multi-hop Wireless Network
NDT	Neighbor Density Table
PDR	Packet Delivery Ratio
PPM	Packet Purse Model
PTM	Packet Trade Model
RREQ	Route Request Packet
SRR	Shortest Reliable Route
SRT	Simple Rating Technique
TTL	Time to Live
Tp	Trusted Party
TPD	Tamper proof device
UREST	Uplink Route Establishment packet
URREQ	Uplink Route Request packet
WRT	Weighted Rating Technique

Chapter 1

Introduction

Multi-hop wireless networks (MWNs) such as mobile ad-hoc, vehicular ad-hoc, multi-hop cellular, and wireless mesh networks [1-7] have been emerging for enabling new applications and enhancing the network performance and deployment [8-13]. In MWNs, the mobile nodes usually act as routers to relay others' traffic to the destination. The network nodes commit bandwidth, data storage, CPU cycles, battery power, etc, forming a pool of resources that can be shared by all of them. The utility that the nodes can obtain from the pooled resources is much higher than that they can obtain on their own. Multi-hop packet relay can extend the communication range using limited transmit power because packets are transmitted over shorter distances. It also can improve the area spectral efficiency and the network throughput and capacity [6], [13]. Moreover, MWNs can be deployed more readily and at low cost in developing and rural areas. However, due to the nature of wireless transmission and multi-hop packet relay, MWNs are vulnerable to serious security challenges that endanger their practical implementation.

1.1 Security Challenges

The assumption that the network nodes are willing to relay others' packets is reasonable for disaster recovery or military applications because the nodes belong to a single authority and pursue a common goal. However, it may not hold for civilian applications where the nodes are autonomous and self-interested in the sense that they aim to maximize their welfare and minimize their contributions. Although the proper network operation requires the nodes to collaborate, collaboration consumes their valuable resources such as energy and computing power and does not provide direct benefits, which stimulates the nodes to behave selfishly. The selfish nodes (or the rational packet droppers) will not be 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 and performance. The fairness issue arises when the selfish nodes take advantage of the cooperative nodes without any contribution to them, 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 [14, 15].

Moreover, the malicious nodes (or the irrational packet droppers) launch DoS (Denial-of-Service) attacks by actively involving themselves in communication sessions and dropping the packets to break the sessions intentionally to adversely degrade the network fairness and performance in terms of throughput, delay, and packet delivery ratio. The malicious nodes may be compromised nodes or malfunctioned nodes having faulty hardware or software. In addition, due to the nature of wireless transmission and multi-hop packet relay, the attackers can analyze the network traffic to learn the users' communication activities and locations causing a serious threat for the users' privacy. The attackers may also trace the network packets backward or forward to identify the source and the destination nodes. The attackers may locate the users in number of hops and track their movements. In a destructive attack, the attackers eavesdrop the traffic to identify the network topology to

detect critical network nodes and then launch directed attacks on them to degrade the network connectivity. The privacy violation attacks can be launched in undetectable way when the attackers just overhear the transmission around them without disturbing the communication or altering the packets.

Reputation-based mechanisms and incentive protocols [16-20] have been proposed to protect against packet drop by enforcing and stimulating the nodes' cooperation, respectively. For reputation-based mechanisms, each node usually monitors the transmissions of its neighbors to make sure that the neighbors relay others' traffics and thus the uncooperative nodes (malicious or selfish) can be identified and punished. Each node maintains a reputation value for each neighbor. A neighbor's reputation value is improved when the neighbor relays a packet and degraded when the neighbor drops a packet. Once a neighbor's reputation reaches a threshold, the neighbor is identifies as uncooperative. For incentive protocols, relaying other nodes' packets is a service not an obligation, and thus the source and the destination nodes pay credits (or micropayment) to the intermediate nodes for relaying their packets. In other words, credits are used to motivate the nodes to collaborate by making cooperation more beneficial than behaving selfishly.

However, the reputation-based mechanisms suffer from essential problems that discourage implementing them practically. First, monitoring the nodes' transmissions by overhearing the channel is not energy-efficient for transmitters. The full power transmission is used instead of adapting the transmission power according to the distance separating the transmitter and the receiver to enable more neighboring nodes to overhear the packet transmission [21]. Furthermore, the directional antennas [22] that can improve the network capacity due to reducing the interference area make monitoring difficult. Second, the reputation-based mechanisms cannot achieve fairness because they do not compensate the nodes for the consumed resources in relaying others' packets, and force the nodes to serve the network without any benefits and punish them 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 much more packets than those at the periphery, they are not compensated. Third, these mechanisms suffer from unreliable detection of the selfish nodes and false accusation of the honest nodes. The reputation-based mechanisms should observe a node's behavior over long time and different sessions to differentiate between the node's unwillingness and incapability to cooperate because packet drop may just happen accidently, e.g., due to low resources, bad channel, and network congestion, but the mechanisms may not have sufficient time to judge a node's behavior precisely due to the node mobility. 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 [23]. Finally, reputation-based mechanisms have not considered the possibility that selfish nodes can collude with each other to boost their reputations to maximize their welfare.

In addition to cooperation stimulation, incentive protocols can achieve fairness by charging and rewarding credits to balance 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 nodes pay for relaying their packets, incentive protocols can discourage *Resource-Exhaustion* attacks where the attackers exchange spurious messages to exhaust the intermediate nodes' resources. Incentive protocols can also be used for charging the future services of the mobile networks because the communication sessions may occur without involving an infrastructure and the mobile nodes may roam among different foreign networks [24], [25], i.e., a mobile node can pay different network operators without contacting a distant home location register.

1.2 Motivations

The efficient implementation of the existing incentive protocols is highly questionable because they impose significant overhead. First, a fair charging policy is to charge both the source and the destination nodes when both of them benefit from the communication. To

securely implement this charging policy, two signatures are usually required per message (one from the source node and the other from the destination node) to prevent payment repudiation and manipulation. However, the extensive use of the public key cryptography requires too complicated computations to be used efficiently in limited-resource nodes, and the secure public-key cryptosystems usually have long signature tags, which degrades the network performance and stimulates the nodes to behave selfishly.

Second, since the communication sessions may occur without involving a trusted party, the nodes usually compose undeniable proof of packet relay, called receipts, and submit the receipts to a trusted party (Tp) to claim the payment. A conclusive point for the practical implementation of the incentive protocols is the receipts' submission and process overhead due to the high frequency of low-value payment transactions. In other words, submitting a large number of receipts implies significant communication and processing overhead, and implementation difficulty because the cost of the receipts' submission and process may exceed the transaction value. Reducing the receipts' number is essential for the practical implementation of an incentive protocol to avoid making bottleneck at the Tp, and to reduce the storage, submission, and processing overheads. The large-size receipts consume the nodes' resources in submitting them and large processing overhead is required to clear the payment.

Third, most of the existing incentive protocols use payment models that have been designed for the Web-based micropayment applications. In order to efficiently implement micropayment in multi-hop wireless networks, the payment model should consider the differences between the Web-based applications and cooperation stimulation.

Fourth, the incentive protocols assume that the nodes are rational packet droppers in the sense that they relay the network packets when they can achieve more benefits than dropping the packets. This assumption cannot be guaranteed in MWNs because unlike the single-hop networks that are run by the operators' equipments, packet routing is performed by the user provided equipment. Some attackers may launch *DoS* attacks by dropping the packets to

disrupt the network proper operation without caring about their interests. A broken node may have a software or hardware fault that prevents it from relaying the packets. In IP networks, the malfunction of the network equipment is an important source for the network unavailability [26]. Moreover, the nodes loaded with low hardware resources may act as irrational packet droppers because they lack the CPU cycles or the buffer space to relay the packets. In Chapter 7, we will show that a low ratio of the irrational packet droppers can significantly degrade the packet delivery ratio.

Fifth, route stability is essential for high-performance MWNs and reliable data transmission. Frequent route failures adversely affect the network performance in terms of throughput and delay. The presence of even a small number of misbehaving nodes could result in repeatedly broken routes and thus the network nodes have to rely on cycles of time-out and route discoveries to communicate. These new route discoveries incur network-wide flooding of routing requests which consumes the network resources such as batteries' energy and bandwidth. However, thwarting selfish and malicious nodes is not sufficient for route stability that requires establishing the routes through the highly trusted nodes having sufficient energy for packet relay. This is because the network nodes may possess different hardware capabilities in terms of CPU cycles and buffer size and thus the high-hardware-resource nodes having sufficient energy are more capable of performing packet relay.

Finally, in the existing incentive protocols, each node has to use a unique identity in its lifetime for charging and rewarding operations, which jeopardizes the users' privacy. Moreover, the existing privacy-preserving routing protocols heavily depend on packet broadcasting and public key cryptography, which makes these protocols infeasible for MWNs due to the constraints on the nodes' resources.

1.3 Contributions

In this thesis, we propose efficient security protocols for thwarting packet drop attacks and preserving users' privacy in MWNs. First, we develop a payment model that takes into

account the features of cooperation stimulation in order to improve the practical implementation of the micropayment in MWNs. Second, based on this payment model, we propose a fair, efficient and secure cooperation incentive protocol to stimulate the selfish nodes' cooperation in MWNs. For fair charging policy, both the source and the destination nodes are charged when the two nodes benefit from the communication. For efficient implementation, our protocol uses the public-key cryptography only for the first packet in a series and uses the efficient hashing operations for the next packets, so that the overhead of the packet series converges to the lightweight overhead of the hashing operations. As we will discuss in details in Chapter 7, the overhead cost of the hashing operations in terms of computational time and energy is incomparable to those of the signing and verifying operations.

Furthermore, instead of generating and submitting a large-size payment receipt per message, each node submits a brief payment report containing its alleged rewards and charges for different sessions and stores undeniable security evidences. We propose a mechanism for investigating the reports' credibility. The payment of the fair reports can be cleared with almost no processing overhead. For the cheating reports, the evidences are requested to identify and evict the cheating nodes that report incorrect payment data. Since the cheating actions are exceptional, the proposed incentive protocol can significantly reduce the required bandwidth and energy for submitting the payment data and clear the payment with almost no processing overhead while achieving the same security strength as the receipt-based protocols. To the best of our knowledge, the proposed incentive protocol is the first proposal that mainly relies on the efficient hashing operations and the first payment-report based (instead of receipt-based) incentive protocol for MWNs.

Thirdly, we propose a trust/reputation system to evaluate the nodes' packet-relay success ratios in terms of trust/reputation values by processing the payment reports. The reports can be processed to extract financial information to reward the intermediate nodes and charge the source and destination nodes, and contextual information such as the broken

links to build up a trust/reputation system to measure the nodes' packet-relay success ratios in terms of trust/reputation values. The system depends on the following fact: if a message is received by node ID_i, this means that all the nodes from the source to ID_{i-1} have successfully relayed the message. A node's trust/reputation value is degraded whenever the node drops a message and improved whenever the node relays a message. Once a node's reputation value reaches to a threshold, the node is identified as malicious and excluded from the network. Using trust/reputation system is necessary to keep track of the nodes' long-term behaviors because the network packets may be dropped normally, e.g., due to mobility, or temporarily, e.g., due to the network congestion, but the high frequency of packet drop is an obvious misbehavior. In our protocol, un-cooperation will not be abused because the nodes are stimulated not forced to cooperate, but frequently breaking the communication sessions is an obvious abuse due to disrupting the routing process. In this way, our protocol uses credits to stimulate the selfish nodes (or rational packet droppers) to cooperate, and reputation values to force the malicious nodes to behave rationally to avoid eviction. To the best of our knowledge, our proposal is the first protocol that can both stimulate the selfish nodes' cooperation and identify the malicious nodes.

Fourthly, the used threshold to identify the malicious nodes should be tolerant to reduce the negative positive ratio and thus the nodes having medium or relatively low trust values may be considered as honest nodes. The honest nodes may have different packet-relay success ratios because they have different hardware capabilities and mobility levels. A node's trust value is a real and live measurement to the node's failure probability and mobility level, i.e., the low mobility nodes having large hardware resources can perform packet relay more efficiently. Therefore, we propose a trust-based and energy-aware routing protocol to route the traffic through the highly trusted nodes having sufficient residual energy in order to minimize the probability of route breakage and maximize route stability, which can significantly improve the network performance in terms of packet delay, throughput, and packet delivery ratio. In other words, the routing protocol selects the nodes that performed

packet relay more successfully in the past and have sufficient energy to relay the session messages. In this way, the protocol stimulates the nodes not only to cooperate but also to improve their packet-relay success ratios and tell the truth about their residual energy to raise their chances to participate in future routes. To the best of our knowledge, this protocol is the first protocol that can both stimulate the nodes' cooperation and make informed routing decisions based on the nodes' past behavior and residual energy to maximize the packet delivery probability.

Fifthly, we propose an efficient privacy-preserving routing and incentive protocol for hybrid ad hoc wireless network. Micropayment is used to foster the nodes' cooperation without submitting payment receipts. The protocol uses only the lightweight hashing and symmetric-key-cryptography operations to preserve the users' privacy. Users' anonymity can be achieved by using pseudonyms instead of the real identities. 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. Moreover, the proposed protocol protects the location privacy of the anonymous source and destination nodes. To the best of our knowledge, this proposal is the first protocol that addresses both cooperation stimulation and user privacy for hybrid ad hoc networks.

Extensive security analysis and simulation results demonstrate that our protocols can secure the payment and trust calculation, and precisely identify and evict the malicious and the cheating nodes. The proposed protocol can provide high protection level for the users' privacy, i.e., any transmission in the network cannot be traced to an individual's real identity and cannot be used to identify the users' locations.

Moreover, our performance evaluations and simulation results demonstrate that our protocols can clear the payment with almost no processing overhead and the submission overhead of the payment-reports are incomparable with that of the receipts in the receipt-based incentive protocols. Our incentive protocol also requires incomparable overhead to the signature-based protocols because the lightweight hashing operations dominate the nodes'

operations. The simulation results demonstrate that the proposed incentive protocol requires less than 10% of the computation time and energy for a series of 13 packets, and the packet overhead is 66% of that of the DSA-based incentive protocol with a probability of 0.95. In addition, the simulation results demonstrate that the trust-based and energy-aware routing protocol can significantly improve the routes' stability and thus the packet delivery ratio compared to the shortest-path routing protocols due to stimulating the selfish nodes' cooperation, evicting the malicious nodes, and making informed decisions regarding the route selection.

1.4 Outline of This Thesis

The remainder of this thesis is organized as follows.

Chapter 2 reviews the related work including the reputation-based mechanisms, incentive protocols, and privacy-preserving protocols. We summarize the mechanisms and the protocols and discuss their characteristics, weaknesses, and strengths.

In Chapter 3, we present the system models including the network and communication model, the threat and trust model, and the payment model.

The proposed incentive protocol and the proposed trust-based and energy-aware routing protocol are presented in Chapter 4. In this chapter, we will discuss how the payment reports are processed to calculate the nodes' trust values. We also will discuss how the malicious nodes can be identified. Finally, we will discuss how the traffic is directed to the highly trusted nodes having sufficient energy.

In Chapter 5, we propose a privacy-preserving routing and incentive protocol for hybrid ad hoc network. Security analysis and performance evaluations are provided in Chapter 6 and 7, respectively, followed by conclusion and future work in Chapter 8.

Chapter 2

Related Work

2.1 Incentive Protocols

The existing incentive protocols can be classified into two categories: tamper-proof-device (TPD) and central-bank based protocols. For TPD-based incentive protocols [27-32], a tamper-proof device is installed in each node to store and manage its credit account and secure its operation. In central-bank based incentive protocols [33-42], a centralized unit, such as Tp, stores and manages the nodes' accounts.

In Nuglets [27, 28], 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 intermediate 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 an auction to sell the packets to the following node in the route. In this way, each intermediate node earns some credits and the destination node pays the total

packet relay cost. In SIP [29], after receiving a packet, the destination node sends a payment RECEIPT packet to the source node to issue a REWARD packet to increment the intermediate nodes' credit accounts. Each message requires three packets between the source and the destination nodes. In CASHnet [30], [31], 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 with real money and/or convert helper credits to traffic credits. The extensive use of the digital-signature operations in both the data and ACK packets is not efficient for the limited-resource nodes. It is shown in [32] that in spite of having helper credits, some nodes starve because they cannot find a service point to convert them to traffic credits.

Centralized-bank based incentive protocols can be classified into electronic-coin and receipt based protocols. For electronic-coin-based incentive protocols [33], each node buys electronic coins from Tp before being involved in a session to pay for relaying its messages. In receipt-based incentive protocols [34-42], the nodes compose undeniable payment receipts (proofs of packet relay) that contain the identities of the payers and the payees, and the payment amount. Since the connection to Tp may not be available on regular basis, the nodes accumulate the receipts and submit them in batch to Tp to update their accounts.

In [33], each node in a session buys packets from the previous node and sells them to the next node. The packets' buyer and seller contact Tp to get deposited coins and submit the coins to claim the payment, respectively. However, the interactive involvement of Tp in each communication session creates a bottleneck at Tp and causes high latency. In Sprite [34], the source node signs a message and the identities of the nodes in the route and appends its signature to the data packet. The intermediate and destination nodes verify the signature and compose a receipt per message and submit the receipts to Tp to claim the payment. In Express [35], the source node generates a hash chain for each intermediate node ID_K and

commits to the hash chain by digitally signing the root of the hash chain 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. The source, intermediate, and destination nodes compose receipts and submit them to Tp. However, the nodes have to generate and store a large number of hash chains because any node in the network may act as an intermediate node due to the node mobility. The packet overhead is large especially if the number of intermediate nodes is large because the source node attaches one hash value for each intermediate node. In Sprite, Express, and INPAC [36], only the source node pays no matter how the destination node benefits from the communication. Moreover, since each intermediate node is rewarded for every relayed message even if it does not reach the destination, all the nodes in a session have to submit the receipts because packet relay is considered successful by a node if its next node in the session reports a valid receipt. The receipts overwhelm the network because of generating and submitting a large number of receipts, which consumes the nodes' storage and energy resources and the network bandwidth, and requires a massive processing overhead to clear the receipts.

The proposed protocol in [37] reduces the receipts' number by rewarding the nodes probabilistically. The source node appends a payment token to each packet, and the intermediate nodes check whether the tokens correspond to winning tickets that are submitted to Tp. The source and the destination nodes are charged per packet but the intermediate nodes are rewarded per winning ticket. However, the fairness issue arises when the nodes' rewards are not proportional to the number of packets they relay, and the colluding nodes can intercept and exchange tokens to be checked locally in each node to steal. In CDS [38], instead of submitting payment receipts to Tp, each node submits a smaller-size activity report containing its alleged charges and rewards for different sessions. Tp uses statistical methods to identify the cheating nodes that submit incorrect reports by measuring how frequency the nodes' reports are inconsistent with others. However, due to the nature of the statistical methods, some honest nodes may be falsely identified as cheaters and colluding nodes may

manage to steal credits. Moreover, credit clearance may be delayed for a long time until identifying the cheating nodes. In [39], Salem et al. have proposed an incentive protocol for hybrid ad hoc network. When a route is broken, the nodes that received the message submit receipts to the base station to secure the payment. Unlike this work, our protocol presented in Chapter 5 can preserve the users' privacy and eliminate the need for submitting receipts.

In [40], an incentive protocol has been proposed for ad hoc network that is used as an access network to connect the nodes to the Internet. For each message, the source node appends a signature to the identities of the nodes in the route, and the destination node signs a receipt and sends it to the last intermediate node to submit to Tp. 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. However, generating two signatures per message is resource consuming for the mobile nodes. Moreover, the source and the destination nodes can communicate freely and the intermediate nodes are not rewarded if the last intermediate node colludes with the source and the destination nodes so as not to submit the receipts. Figure 2.1 shows the charges and rewards for relaying X messages in a session with n intermediate nodes assuming λ credits are paid per message and the source and destination nodes are charged the ratios P_r and $(1-P_r)$ of the total payment, respectively. If the source and destination nodes collude with the last intermediate node and the receipt is not submitted, the colluders can save $X \cdot \lambda \cdot (n-1)$ credits, and thus the source and destination nodes can compensate the colluding intermediate node. On the other hand, it is not efficient to submit a receipt by each intermediate node [34], [35], [36], due to significantly increasing the redundant receipts' number.

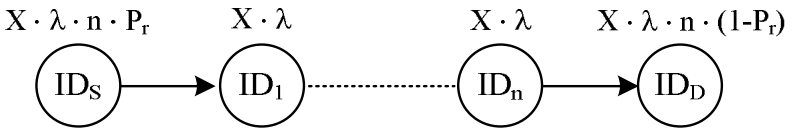


Figure 2.1: The charges and rewards for relaying X messages.

In PIS [41], 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 route regardless of the messages' number. Instead of submitting the receipts by all the intermediate nodes, a reactive and preventive receipt submission schemes have been proposed in PIS [41] and DSC [42], respectively to reduce the number of submitted receipts and protect against collusion attacks. In the reactive scheme [41], the first intermediate node after the source has to submit the receipt and the other intermediate nodes submit the receipt with very low probability to thwart the receipt-submission collusion attacks with submitting few additional receipts and limiting the number of un-submitted receipts probabilistically. Unlike the reactive receipt submission scheme that aims to identify and evict the colluding nodes, the preventive receipt submission scheme [42] aims to fail the collusion attack. In this scheme, each intermediate node submits a randomly chosen ratio of the receipts to guarantee submitting the receipts probabilistically even if some intermediate nodes collude with the source and destination nodes. Unlike PIS that requires two signatures for each message (one from the source node and the other from destination node), DSC can reduce the public-key-cryptography operations by replacing the destination node's signature with the hashing operations. The destination node generates a hash chain by iteratively hashing a random value to obtain the hash chain root. The destination node signs the hash chain root, and releases one hash value from the chain in each ACK packet.

2.2 Reputation-based Mechanisms

The reputation-based mechanisms use reputation systems to differentiate between the cooperative nodes that drop the packets normally, e.g., due to mobility and bad channel, and the uncooperative nodes that drop the packets intentionally. The reputation system sets and updates a reputation value for each node. A node's reputation is improved when it relays packets successfully, but the reputation is degraded when the node drops packets. Once a

node's reputation degrades to a threshold, the node is identified as uncooperative and punished.

In [14], two modules called watchdog and path-rater are implemented in each node. When node ID_1 transmits a packet to ID_2 to relay to ID_3, the watchdog module of ID_1 overhears the medium to make sure that ID_2 relays the packet. ID_1 increases the reputation of ID_2 when it overhears the packet transmission; otherwise, ID_1 decreases the reputation of ID_2. ID_1 accuses ID_2 of un-cooperation as soon as its reputation degrades beyond a threshold. Based on the watchdog's accusations, the path-rater module chooses the path that avoids the uncooperative nodes without punishing them, which imposes extra load on the cooperative nodes without any benefits. In OCEAN [43], a node's reputation is initialized to neutral (0), every positive behavior (relaying a packet) results in an increment (+1), and every negative behavior (dropping a packet) results in a decrement (−2). Once a node's reputation falls below a threshold (−40), the node is identified as uncooperative. However, in [14] and [43], the nodes depend only on their observations to evaluate a node's behavior and they do not share their evaluations, which may degrade the mechanism's effectiveness because the cooperative nodes that drop the packets temporarily due to the network congestion or other reasons may be falsely identified as uncooperative.

CONFIDANT [44] prefers the good-reputation nodes in route selection, which imposes more burdens on these nodes without immediate benefits. For CONFIDANT and CORE [45], each node combines its evaluation with other nodes' evaluations to calculate a node's reputation value. Only the positive evaluations are propagated in CORE to prevent defaming the nodes' reputations by propagating false negative evaluations, but only the negative evaluations are propagated in CONFIDANT to prevent the colluders from falsely boosting their reputations by propagating false positive evaluations. However, in order to precisely judge a node's real behavior, both the negative and the positive behaviors should be considered. A node's reputation is decremented along the time in CORE when the node does not relay packets, which is not fair for the nodes that reside at the network edge because they

are less frequently selected by the routing protocol. In CORE and CONFIDANT, the uncooperative nodes are excluded from the network by avoiding them in routing and denying them cooperation. Nevertheless, the isolation of the uncooperative nodes is performed unilaterally, which may result in false accusations because when a node ID_i denies relaying the uncooperative nodes' packets, the neighbors of ID_i may consider its behavior illegal.

In SORI [46], each node counts the packets relayed both by and for neighboring node, and the ratio of these counts is combined with reports from other nodes to calculate the node's reputation. However, the less frequently selected nodes by the routing protocol such as those at the network perimeter have falsely bad reputation. In [47], 2-hop *ACK* technique is used to monitor a node's behavior instead of using the medium overhearing technique. The node ID_1 accuses the next node ID_2 of dropping the packet, if ID_1 does not receive *ACK* from the 2-hop away node ID_3, but the mechanism completely fails when two neighboring nodes collude to issue fake *ACK*s.

In the reputation-based mechanisms, the nodes may relay the packets to avoid punishment, but do not monitor their neighbors to save their resources and make use of the other nodes' evaluations to avoid routing their packets through the uncooperative nodes, which degrades the mechanism effectiveness. Moreover, the medium overhearing technique suffers from inaccuracy problems because the assumption that the transmitted packets by a node can be overheard by all the nodes in its neighborhood cannot be ensured for the following reasons [14]: (1) When a node ID_2 relays a packet to ID_3, it is possible that ID_1 cannot overhear the transmission due to another concurrent transmission in ID_1's neighborhood [23]; and (2) Since ID_1 can know whether ID_2 has relayed a packet but not if ID_3 received it, a misbehaving node ID_2 can save its energy and circumvent the monitoring technique if ID_1 is closer than ID_3 by adjusting its transmission power such that the signal is strong enough to be overheard by the monitoring node ID_1 but too weak to be received by the true recipient ID_3 [21].

2.3 Privacy-Preserving Routing Protocols

In [48], 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 certificates have different pseudonyms and signed by Tp to certify the keys. 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 with 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 resource consuming. Since the network nodes have a large number of certificates, certificate revocation is a real challenge, and these keys can be used to launch *Sybil* attacks [49].

In ANODR [50], 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 with a shared key with the destination. Each node X tries to open the trapdoor. If it is not the destination, X adds a nonce N_X and encrypts the packet with a 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 to know if it is the destination. This is because the identities of the source and the destination nodes are hidden for anonymity. Moreover, the source and the destination nodes cannot establish session keys shared with the intermediate nodes to make a

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 [51], 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 with the public key of the destination node. Each node tries to open the trapdoor with its private key. 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 the 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 requires every node to perform a decryption operation using a private key, an encryption operation using a public key, and a signature operation for every *RREQ* packet it forwards. The sizes of the *RREQ* and the *RREP* packets are large which consumes much energy and bandwidth. The destination node learns the identities of all the nodes in the route. The location of the destination node is also disclosed to the source node. In the *RREQ* packet, a malicious intermediate node can delete the last part of the routing information that is attached by its previous nodes and makes new routing information.

Chapter 3

System Models

3.1 Network and Communication Models

As illustrated in Figure 3.1, the considered multi-hop wireless network includes a trusted party (Tp), mobile nodes, and base stations in some types of the MWNs. The base stations are connected with each other and with Tp by a backbone network that may be wired or wireless. Tp stores and manages the nodes' credit accounts and trust values, and generates private/public key pair and certificate with a unique identity for each node to participate in the network. Each node stores a unique identity and public/private key pair with a certificate, the public key of Tp, and the required cryptographic data to enable any two nodes to share a symmetric key. For efficient implementation, an identity-based key exchange protocol based on bilinear pairing can be used because the nodes do not need to exchange messages to compute the shared keys. Tp generates a prime p, a cyclic additive group (G), and a cyclic multiplicative group (G_T) of the same order p such that an efficiently computable bilinear pairing $\hat{e}$: $G \times G \rightarrow G_T$ is known. The bilinear mapping has the following properties:

- **Bilinear**: $\hat{e}(a \cdot P, b \cdot Q) = \hat{e}(b \cdot P, a \cdot Q) = \hat{e}(P, Q)^{a \cdot b}$, for all $P, Q \in G$ and $a, b \in Z_P^*$.

- **Non-degeneracy**: $\hat{e}(P, Q) \neq 1_{GT}$.

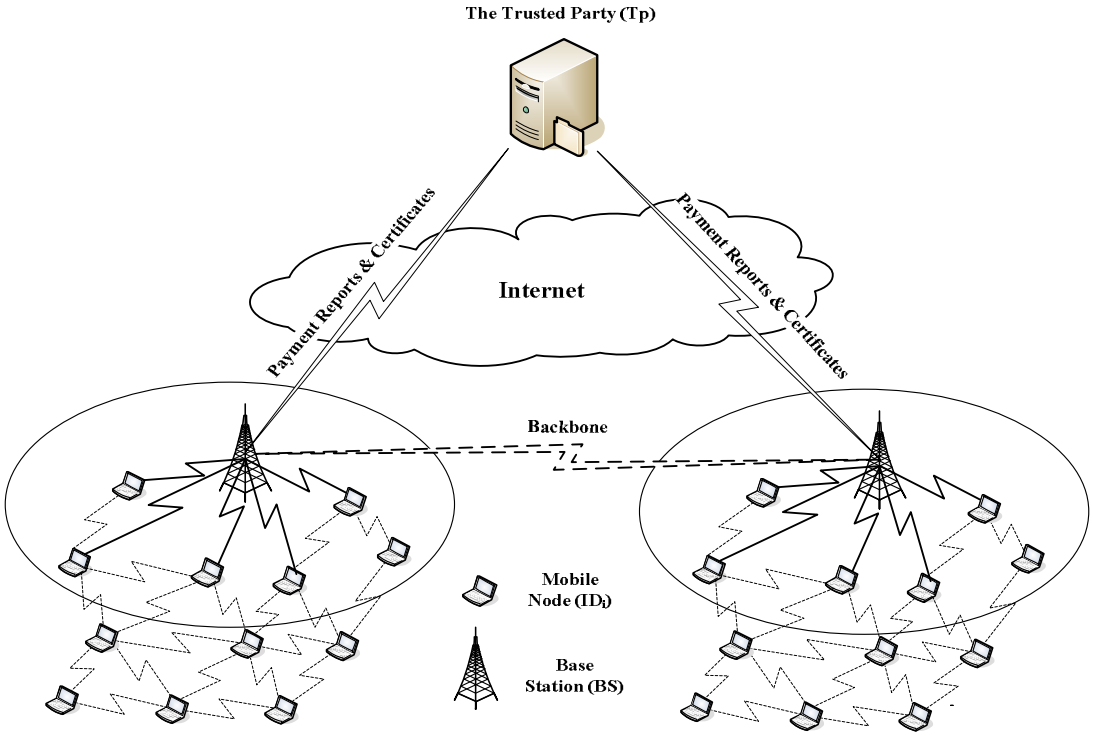


Figure 3.1: The architecture of the multi-hop wireless network.

- Symmetric: $\hat{e}(P, Q) = \hat{e}(Q, P)$, for all $P, Q \in G$.

- Admissible: there is an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in G$.

The bilinear map $\hat{e}$ can be implemented efficiently using the Weil and Tate pairings on elliptic curves [52]. Tp selects a random element $\mu \in Z_P^*$ known as the master key, and computes the secret keys for the nodes based on their identities. The secret key for node ID_i is $Sk_i = \mu \cdot H(ID_i) \in G$, where $H:\{0,1\}^* \rightarrow G$.

Once Tp receives the payment reports from the nodes, it processes them to update the nodes' credit accounts and trust values. Once Tp detects malicious or cheating nodes, Tp revokes the nodes by not renewing their certificates. The source node's packets may be relayed in several hops by the mobile nodes, and the base station(s) whenever it is necessary, to the destination node. The network nodes can contact Tp at least once during a time interval

called updating time that can be in the range of few days. During this connection, the nodes submit the payment reports, renew their certificates, and convert credits to real money and/or purchase credits with real money. This connection can occur via the base stations, Wi-Fi hotspots, or wired networks such as the Internet. All the communications are unicast and the nodes can communicate in one of two modes: pure ad hoc or hybrid. For pure ad hoc mode, the source and the destination nodes communicate without involving base stations. The source node's messages may be relayed in several hops by the intermediate nodes to the destination node. For hybrid mode, at least one base station is involved in the communication. The source node transmits its messages to the source base station (Bs), if necessary in multiple hops. If the destination node resides in a different cell, the messages are forwarded to the destination base station (Bd) that transmits the messages to the destination node possibly in multiple hops.

3.2 Threat and Trust Models

The mobile nodes and the base stations are probable attackers but Tp is fully secure. The mobile nodes and the base stations are motivated to misbehave to increase their welfare because the mobile nodes are autonomous and self-interested and the base stations may be owned by different providers. It is impossible to realize secure payment between two entities without trusted third party [53]. The attackers have full control on their devices and thus they can change their operation and infer the cryptographic data. The attackers can be classified into two classes: rational attackers and irrational packet droppers. The rational attackers misbehave when they can achieve more benefits than behaving honestly. Specifically, they attempt to steal credits, pay less, and communicate freely. On the contrary, the irrational packet droppers aim to disrupt the packet transmission by involving themselves in sessions and dropping the data packets to break the sessions without considering their interests and the attack cost such as energy and credits. The irrational packet droppers may launch *Black-Hole* attack by continuously breaking all the sessions they participate in, or launch *Gray-Hole*

attack by intentionally breaking some sessions and behaving regularly in other sessions to circumvent Tp but the ratio of the broken sessions should be high to launch effective attacks. The irrational packet droppers may be compromised, malfunctioned, or low-hardware-resource nodes.

Non-participation in packet relay is not abuse because the nodes are stimulated not forced to cooperate, but the high frequency of packet drop is an abuse due to disrupting the data transmission process. The attackers may work individually or collude with each other to launch sophisticated attacks. The colluding irrational packet droppers may launch two attacks: *Trust-Boost* and *False-Accusation* attacks. For *Trust-Boost* attack, the attackers attempt to augment their trust and reputation values falsely to escape the consequence of dropping the packets; and for *False-Accusation* attacks, the attackers try to defame the trust values of honest nodes to evict them from the network. The gained experience from the currently used protocols in civilian applications emphasizes that large-scale irrational collusion attacks are highly unlikely [54], [55]. The trust/reputation systems are susceptible to the large-scale collusion attacks due to the nature of these systems. Our security objective is to protect the payment against large-scale collusion attacks, and to protect the trust/reputation system against small-scale irrational collusion attacks launched by low number of colluders, e.g., in the range of ten, and improve the system's robustness against large-scale attacks. For the trust models, the nodes fully trust Tp to perform billing and auditing and trust calculations, but Tp does not trust any node or base station in the network.

3.3 Payment Model

Micropayment schemes [56-58] 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 these applications include buying data or news, listening to a song,

playing an online game, and reading an article in a journal [59]. In order to efficiently implement such scheme in MWNs, the payment model should consider the differences between web-based applications and cooperation stimulation. These differences are summarized in Table 3.1. 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 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 the nodes' mobility and the 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 to the computers' large resources in Web-based applications. Although security is important in all payment applications, the attacks can be launched easier in Web-based applications because it is easier to launch attacks across the Internet than tampering devices.

Table 3.1: Properties of Web-based applications and cooperation stimulation.

			Web-based applications	**Cooperation stimulation**
P1	**Transactions' 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

3.3.1 Parties and Relations

The payment model has three basic parties: the customer or the source and the destination nodes, the merchant or the intermediate nodes, and the bank or Tp. Figure 3.2 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. The customer contacts the bank periodically to renew his certificate and pay for the services (packet relay) he received from the merchants. In Payment Phase, the customer's certificate enables him to issue payment evidences to transact with merchants without involving the bank, i.e., customers mine their own electronic coins without the need for direct verification by the bank. In Redemption Phase, each merchant claims its payment by submitting payment reports. Tp clears the fair reports by rewarding the merchants and charging the customers and requests the evidences to identify the cheating nodes. 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.

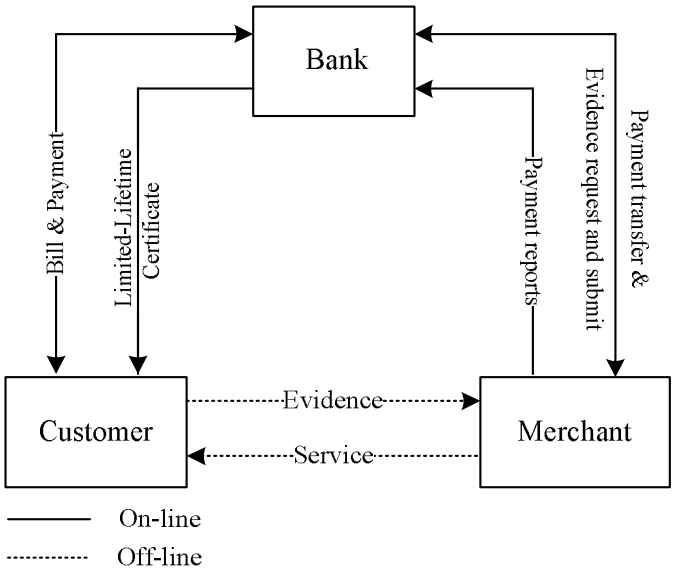


Figure 3.2: The payment model's parties and relations.

A. No Need for Tamper-Proof-Device

The TPD-based incentive protocols [27-32] may not find widespread acceptance for the following reasons. First, the assumption that the TPD cannot be tampered is neither secure nor practical for MWNs. That is because the nodes are autonomous and self-interested and the attackers can communicate freely in undetectable way if they could compromise the TPDs [60]. Moreover, since the security protection of these protocols completely fails if the TPDs are tampered, only a small number of manufactures can be trusted to make the network nodes, which is too restrictive for civilian networks. Second, a node cannot communicate if it does not have sufficient credits at the communication time. Unfortunately, the nodes at the network edge cannot earn 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 significantly degrades because the rich nodes are not motivated to cooperate and the poor nodes cannot initiate communications. Finally, since credits are cleared in real-time, the network performance degrades if the network does not have enough credits circulating around. In [32], it is shown that the overall credits in the network gradually decline because the total charges are not necessarily equal to the total rewards. That is because the source node is fully charged after sending a packet but some intermediate nodes may not be rewarded when the route is broken. In [29], a compensation mechanism is used to change the packet-relaying price proportionally to the nodes' speed to avoid the 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.

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. In our payment

model, Tp sells credits with real money to enable the border nodes to communicate, improve credit distribution, and protect the network from credit decline. However, we do not consider this as a fairness problem because the philosophy behind the incentive protocols is that packet relay is a service not an obligation. This service may not be requested from some nodes, i.e., the customers request the packet-relay service from the best service providers. If the traffic is directed through the border nodes, obviously, we sacrifice the network performance because the routes may be long. Due to the nodes' mobility, the border nodes can change their location and earn more credits as shown in [29]. Moreover, the border nodes do not relay as many packets as others, and thus it is fair to charge the border nodes real money to compensate the other nodes that relayed more packets.

B. Flexible Payment

There are two ways for managing the 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, the 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, the customers are served first and charged later, e.g., the 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 the payment reports are processed. For debit payment, the customers' accounts are charged before they are served, e.g., customers buy electronic coins in advance from the bank to pay 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, the 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 at the bank due to the high frequency of low-value

transactions (P4 and P5 in Table 3.1). Second, customers generate their own coins (or evidences), which provides many flexibilities. Coins are generated on-demand, and customers do not need to frequently contact the bank to buy coins. In [32], 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 a specific merchant [33, 35] is not practical due to the large number of probable merchants in the network, and generating general coins to pay for any merchant is vulnerable to *Double-Spending* attack or requires interactive and frequent contact with the bank. Although, the developed payment architecture has many positives, it is obvious that reducing the overhead of submitting and clearing the payment reports is essential for the 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 3.1), generating a receipt (or payment report) per message [34] or customer [35] significantly increases the receipts' number, and thus the transaction value may not cover its processing cost (P5 in Table 3.1). Second, the nodes have low resources (P7 in Table 3.1) so the overhead of storing and submitting a large number of receipts may stimulate the nodes to behave selfishly.

3.3.2 Charging and Rewarding Policy

In most existing incentive protocols, only the source node is charged no matter how the destination node benefits from the communication. We argue that a fair charging policy is to support 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 destination nodes agree on the payment-splitting ratio during the session establishment phase. For rewarding policy, some incentive protocols such as [61, 62] consider different packet relaying rewards that correspond to the incurred energy in relaying the packets. 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 [33-42], we use fixed rewarding rate, e.g., λ credits per unit-sized message.

In MWNs, packet loss may occur normally due to the node mobility, packet collision, channel impairment, etc. 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 complicated design. For example, rewarding the nodes for route establishment packets or packet retransmissions complicates the incentive protocol and significantly increases the payment reports' number because a large number of nodes may be involved in relaying route establishment packets and packet retransmissions happens frequently in wireless networks. Our charging and rewarding policy rewards the intermediate nodes only for the delivered messages as indicated in Figure 3.3(a), but the source and the destination nodes are charged for every transmitted message whether it reaches the destination or not, as illustrated in Figure 3.3(a, b). For fair rewarding policy, the value of λ is determined to compensate the nodes for the consumed resources in relaying route establishment packets, packet retransmission, and undelivered packets. In Chapter 6, we will argue that this charging and rewarding policy can discourage the rational attacks and encourage the nodes' cooperation. Table 3.2 gives the used notations in this thesis.

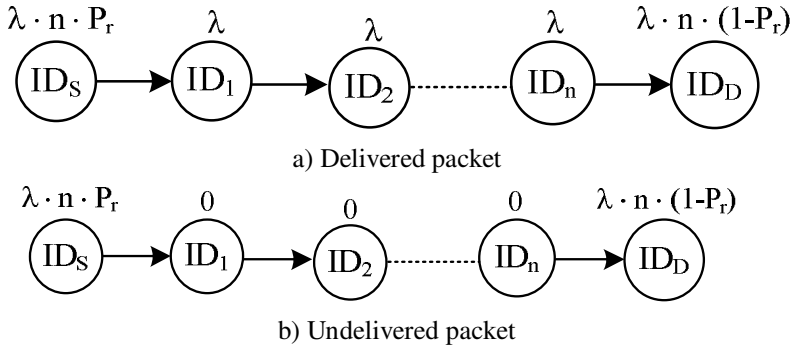


Figure 3.3: The payment rewarding and charging policy.

Table 3.2: Useful Notations.

Symbol	Description
(M)K	The ciphertext resulted from encrypting M with key K.
A, B	A is concatenated to B.
Bs	The source node's base station.
Bd	The destination node's base station.
$Cert_S$ and $Cert_D$	The certificates of the source and the destination nodes, respectively.
CSR(C)	A complete session report for C delivered messages.
H(X)	The hash value resulted from hashing X.
$H_{KSi}(X)$	The hash value resulted from keyed hashing X using the key K_{Si}.
$HS(M_X)$	Hash series of message M_X. A concatenation of the keyed hash values generated by the shared keys between the source (S) and the session nodes (A, B,…, D), e.g. $HS(M_X)=H_{KSA}(M_X)$, $H_{KSB}(M_X)$,…., $H_{KSD}(M_X)$.
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.
ID_{XY}	The used pseudonym when node X communicates to node Y.
ISR(C)	An incomplete session report for C-1 delivered messages and one sent message.
K_{Si}	The symmetric key shared between the source node and the intermediate node *i*.
M_i	The message sent in the *i*th data packet in a session.
n	The number of intermediate nodes.
N	The hash chain size.
$P_C(R_L)$	The payment clearance delay for a session with R_L nodes.
Pr	The payment-splitting ratio that is paid by the source node. The ratio of (1-Pr) is paid by the destination node
P_L	Padding length.
r	A random number.
R	The concatenation of the identities of the nodes in a route, e.g. ID_S, ID_1,…., ID_D
$R_{A,J}$	The rating of node A in session j
R_m and R_h	The honest and malicious nodes' reputation thresholds, respectively.
R_L	The route length, i.e., the number of nodes in a route including the source and the destination nodes, where $R_L = n + 1$.
$R_{St,i}(t)$ and $R_{Lt,i}(t)$	The short-term and long-term reputation values of node i at time t, where $R_{St,i}(t)$ and $R_{Lt,i}(t) \in [0, 1]$.
$S_i(t)$	The state of node i at time t. $S_i(t) \in \{+1, 0, -1\}$ which corresponds to {Honest, Suspicious, Malicious}.
SI	The session identifier that includes the path identities and establishment time.
$Sig_i(X)$	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.
T_i	A random variable denoting the time between each two report submissions of node i
T_{Cert}	The certificate lifetime.
$T_{C,A}(t)$	The trust value number C of node A at time t, where $C \in \{1, 2, 3\}$
Ts	A session's establishment time stamp.
V_S^X, V_D^X	The hash value number X in the hash chains generated by the source and the destination nodes, respectively.

Chapter 4

ESIP: Efficient and Secure Protocol for Thwarting Packet Drop

In this chapter, we propose an **E**fficient and **S**ecure protocol for thwart**I**ng **P**acket drop in MWNs, called **ESIP**. ESIP can stimulate the selfish nodes' cooperation and identify and evict the irrational packet droppers. We also propose two trust-based and energy-aware routing protocols, called **BAR** (**B**est **A**vailable **R**oute) and **SRR** (**S**hortest **R**eliable **R**oute), to establish the routes through the highly trusted nodes having enough energy to relay the session messages. ESIP has four phases shown in Figure 4.1. In *Data Transmission* phase, the source node transmits messages to the destination, and the source, intermediate and destination nodes save security tokens called *Evidence*s and compose payment reports. The nodes submit the payment reports containing the payment data of different sessions to Tp to redeem the payment. In *Route Establishment* phase, BAR and SRR routing protocols can be implemented to establish the routes through the highly trusted nodes having sufficient energy.

In *Classifier* phase, the Tp classifies the reports into fair and cheating. In cheating reports, some nodes in the session report incorrect payment, e.g., to steal credits. These reports are passed to *Cheaters Identification* phase that requests the *Evidence*s to identify the cheating nodes. In *Processing* phase, the payment reports are processed to extract contextual and financial information. The financial information is passed to the *Credit-Account Update* phase to update the nodes' credit accounts. The contextual information is passed to the *Trust Update* phase to update the nodes' trust and reputation values. The malicious and cheating are evicted by denying renewing their certificates.

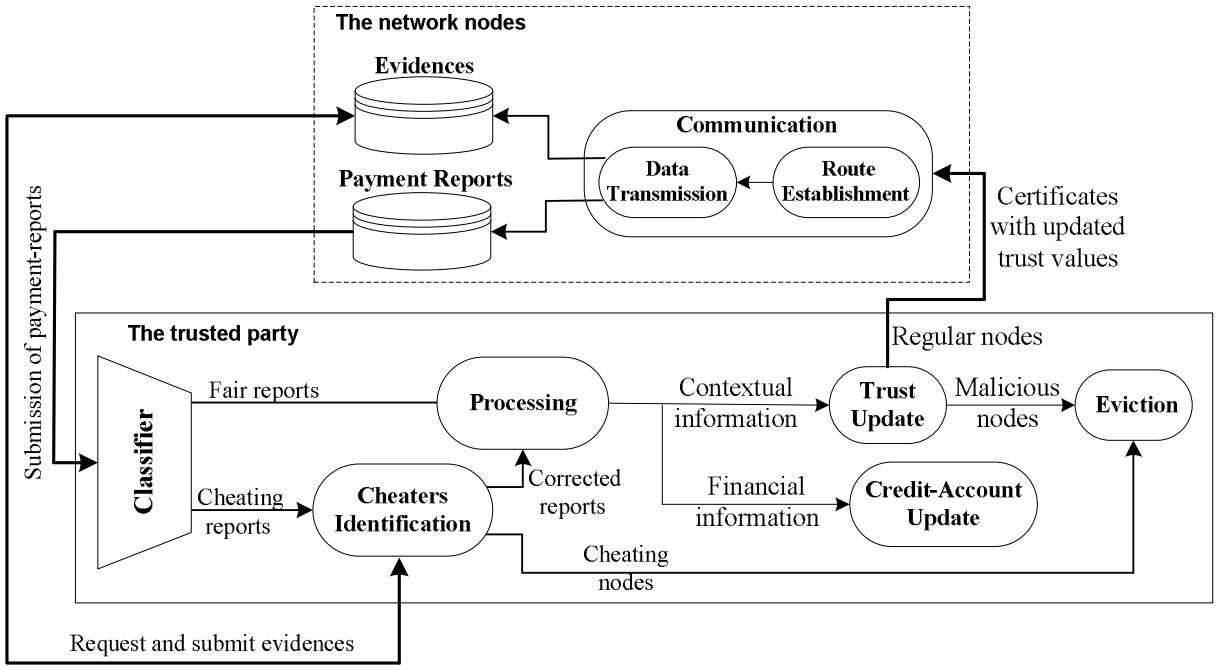


Figure 4.1: The architecture of ESIP.

4.1 Data Transmission Phase

In this phase, the source node sends data packets to the destination along the established route, and the destination node replies with *ACK* packets to acknowledge receiving the packets.

As illustrated in Figure 4.2, each node in the route shares a symmetric key with the source node to compute the messages' keyed hash values. Two nodes with identity/secret key pairs (ID_S, SK_S) and (ID_A, SK_A) can independently compute the shared key as follows:

$$\mathbf{K_{SA}} = \hat{e}(H(ID_A), Sk_S)$$

$$= \hat{e}(H(ID_A), \mu \cdot H(ID_S))$$

$$= \hat{e}(\mu \cdot H(ID_A), H(ID_S)) \qquad (\textit{Bilinear property})$$

$$= \hat{e}(Sk_A, H(ID_S))$$

$$= \hat{e}(H(ID_S), Sk_A) \qquad (\textit{Symmetric property})$$

$$= \mathbf{K_{AS}}$$

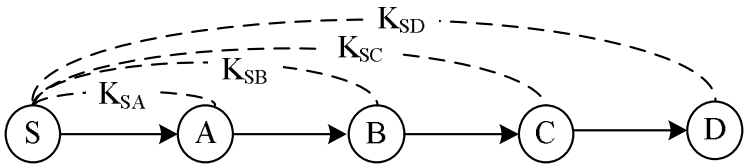


Figure 4.2: The source node shares a key with each node in the route.

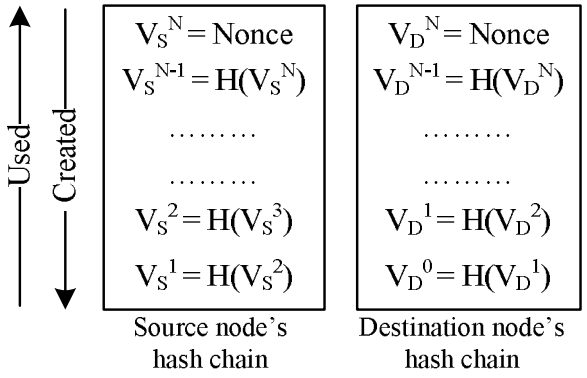


Figure 4.3: The source and destination nodes' hash chains.

As shown in Figure 4.3, the source and the destination nodes generate hash chains by

iteratively hashing random values V_S^N and V_D^N N and N+1 times to obtain final hash values V_S^1 and V_D^0, respectively where $V_S^{i-1} = H(V_S^i)$ and $V_D^{i-1} = H(V_D^i)$. The hash values are released in the direction from V_S^1 to V_S^N and V_D^0 to V_D^N. Payment non-repudiation is achievable because it is difficult to compute V_S^i from V_S^{i-1} or V_D^i from V_D^{i-1} for $2 \leq i \leq N$. In order to authenticate the hash chains and link them to the session, the source and the destination nodes sign the roots of the hash chains and the session unique identifier (SI), where SI contains the identities of the nodes (the payers and the payees) in the route (R), the session establishment time stamp (Ts), and Pr, e.g., R = ID_S, ID_A, ID_B, ID_C, ID_D and SI = R, Ts, Pr for the session shown in Figure 4.2. The source node's signature is sent in the first data packet while the destination node's signature is sent in the *RREP* packet.

4.1.1 Data Generation and Relay

For the first data packet, Figure 4.4 shows that the source node appends the message M_X, V_S^1 and its signature ($Sig_S(SI, V_S^1)$). This signature proves the source node's approval to pay for the session and authenticates its hash chain and links it to the session, i.e., the sender cannot deny generating the hash chain or initiating the session. In order to ensure the hop-by-hop message's authenticity and integrity, the message's hash value ($H(M_1)$) can be included in the signature, but that increases the *Evidence* size because $H(M_1)$ has to be attached to the evidence. Therefore, the source node attaches the hash series $HS(M_1)$ which contains a truncated keyed hash value for each node, e.g., $HS(M_1) = H_{KSA}(M_1)$, $H_{KSB}(M_1)$, $H_{KSC}(M_1)$, $H_{KSD}(M_1)$ in Figure 4.4. Each intermediate node verifies the source node's signature to compose valid *Evidence*. Then it verifies its message's truncated hash value to ensure the message's authenticity and integrity, and relays the packet after dropping its hash value as shown in Figure 4.5. Each intermediate node saves the source node's signature and V_S^1 to be used in the *Evidence* composition.

For the successive packets (X > 1), Figure 4.4 illustrates that the source node appends the pre-image of the last sent hash value (V_S^X) as an approval to pay for one more packet,

and the truncated hash series ($HS(M_X)$). Before relaying a data packet, each intermediate node verifies its message's truncated keyed hash value, verifies that V_S^{X-1} is generated from hashing V_S^X, and relays the packet after dropping its hash value. The intermediate nodes store only the last received hash value to be used in the *Evidence* composition, i.e., after receiving the *X*th data packet, the intermediate node deletes V_S^{X-1} and store V_S^X. V_S^X alongside with the source node's signature are enough to prove that *X* messages have been transmitted and *X-1* messages have been delivered. Each node in the session restarts a timer each time the node transmits or relays a packet. The session is considered broken when the timer expires.

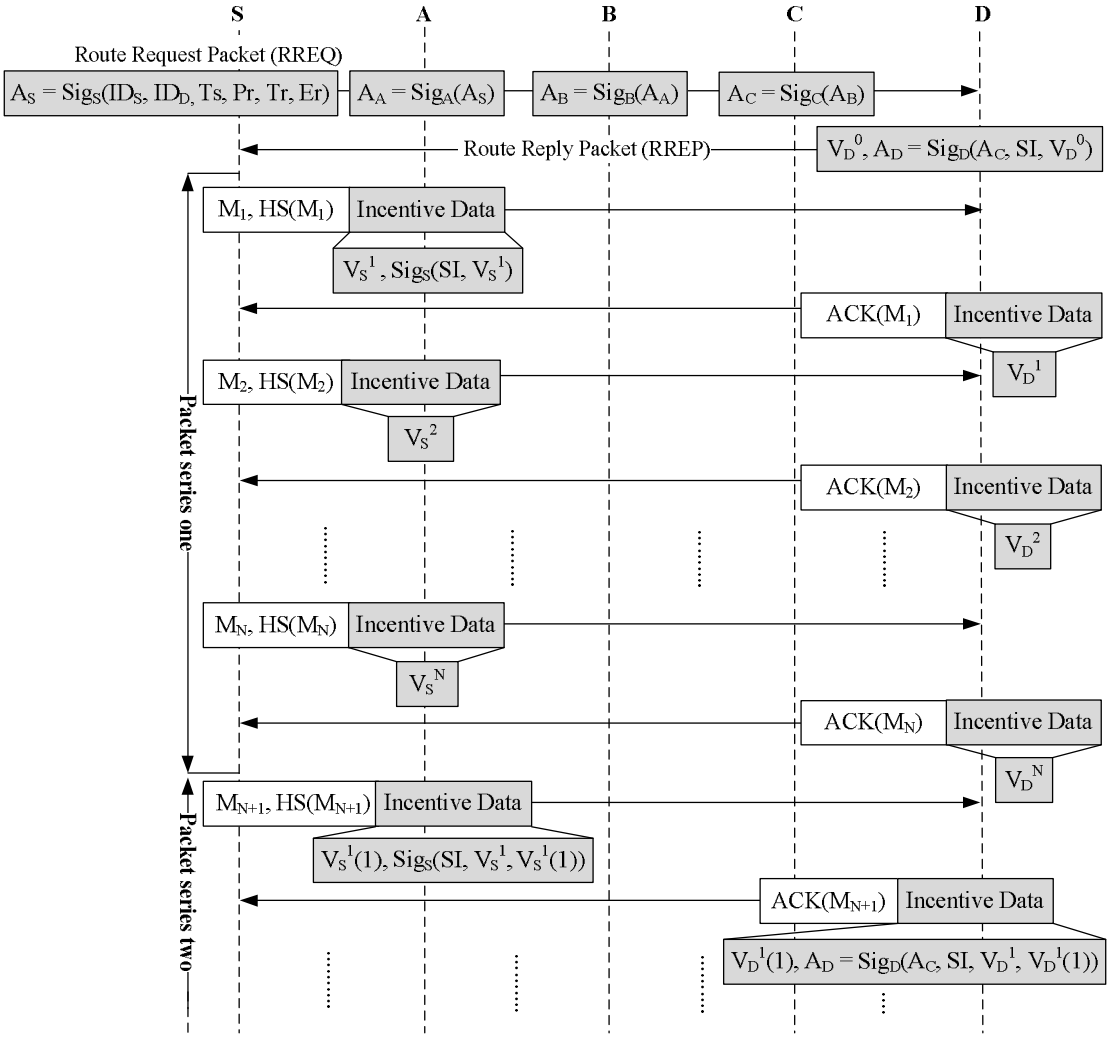


Figure 4.4: The exchanged security tags in a session.

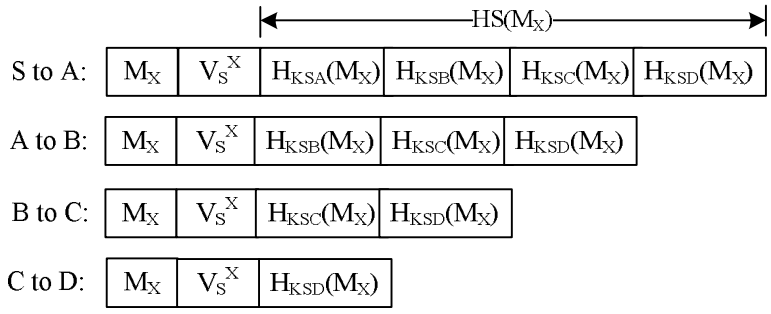


Figure 4.5: The hop-by-hop security packet-overhead in the *X*th data packet, (X > 1).

4.1.2 ACK Generation and Relay

Upon receiving the *X*th data packet and X ≤ N, Figure 4.4 shows that the destination node sends back *ACK* packet containing the pre-image of the last released hash value, or V_D^X, to acknowledge receiving the message in undeniable way. Therefore, instead of generating a signature per *ACK* packet, one signature is generated per N *ACK*s. Payment non-repudiation and non-manipulation are achievable because the hash function is one-way, i.e., only the destination node could have generated the hash chain because it is not possible to compute V_D^{X+1} from V_D^X. Each intermediate node verifies that V_D^X is generated from hashing V_D^{X-1}, and stores the last hash value (V_D^{X-1}) to be used in the *Evidence* composition.

As illustrated in Figure 4.4, after releasing all the hash values of the first hash chain, the source and the destination nodes create new hash chains by iteratively hashing random values $V_S^N(1)$ and $V_D^N(1)$ *N* times. In the data packet number *N+1*, the source node authenticates its new hash chain and links it to the session by signing SI and the roots of all used hash chains (V_S^1 and $V_S^1(1)$), instead of signing only the last hash chain's root. In this way, the intermediate nodes store only the last signature for the *Evidence* composition because it can authenticate all the used hash chains in the session. In the *ACK* of the message number *N+1*, the destination node sends its signature for the authentication code (A_C), SI, the roots of all used hash chains (V_D^0 and $V_D^1(1)$). More details about A_C will be given in Section 4.8. The nodes store the hash chains' roots and the last signature for the *Evidence* composition.

Table 4.1: Numerical example for payment reports from node ID_A.

Session Identifier	Messages' number (X)
$\underline{\mathbf{ID_A}}$, ID_B, ID_N, ID_C, Pr_1, Ts_1, 0	12
ID_F, ID_C, ID_B, ID_H, $\underline{\mathbf{ID_A}}$, Pr_2, Ts_2, 1	17
ID_C, ID_F, $\underline{\mathbf{ID_A}}$, ID_H, Pr_3, Ts_3, 0	15
…….	…….
……	

4.2 Evidence Composition and Payment-Report Submission

Whether the route is completed or broken, each node in the route composes a payment report and security *Evidence*. The payment report contains the session identifier (SI), a flag bit (F) indicating whether the last received packet is data or *ACK*, and the number of sent, relayed, or received messages by the source, intermediate, and destination nodes, respectively. Since the connection to Tp may not be available on regular basis, the nodes accumulates the payment reports and submit them in batch to Tp for redemption. Table 4.1 gives numerical example for payment reports from node A. For the first report, A is the source node, C is the destination, and B and N are the intermediate nodes. In its report, node A claims that it sent 12 messages but did not receive the *ACK* of M_{12} because F is zero. For the second report, A is the destination node and claims receiving 17 messages and sending the *ACK* of M_{17}. For the third report, A is an intermediate node and claims relaying 15 messages, but it did not receive the *ACK* of M_{15}.

In this thesis, *Evidence* is defined as information that is used to establish proof about the occurrence of an event and the amount of payment to resolve a dispute about the amount of payment resulted from a session. The general format of the session evidence is shown in Figure 4.6(a), where the brackets "[Y]" mean that Y may not exist in some cases. An *Evidence* contains two main parts called descriptor (D) and security token (S_T). The descriptor contains SI, the roots and seeds of the source and the destination nodes' hash chains, and the last releases hash values ($V_D^L(v)$, $V_S^L(v)$). S_T is an undeniable proof that prevents payment

repudiation and manipulation and thus ensures that the payment is undeniable, unmodifiable, and unforgeable. In order to significantly reduce the *Evidence*'s size, S_T can be composed by hashing the source and the destination nodes' last signatures instead of attaching the large-size signatures, e.g., $S_T = H(\underline{Sig_S(SI, V_S^1, [V_S^1(1),\ldots]), Sig_D(A_D, SI, V_D^0, [V_D^1(1),\ldots])})$. In Section 4.8, more details will be given about the energy and trust requirements (Er and Tr). The *Evidence* size depends on the number of used hash chains because two hash values should be attached for each hash chain, and thus properly choosing the hash chain size can minimize the *Evidence* size. An *Evidence* contains V_S^X when the node receives the *X*th data packet and V_D^X when it receives the *X*th *ACK* packet.

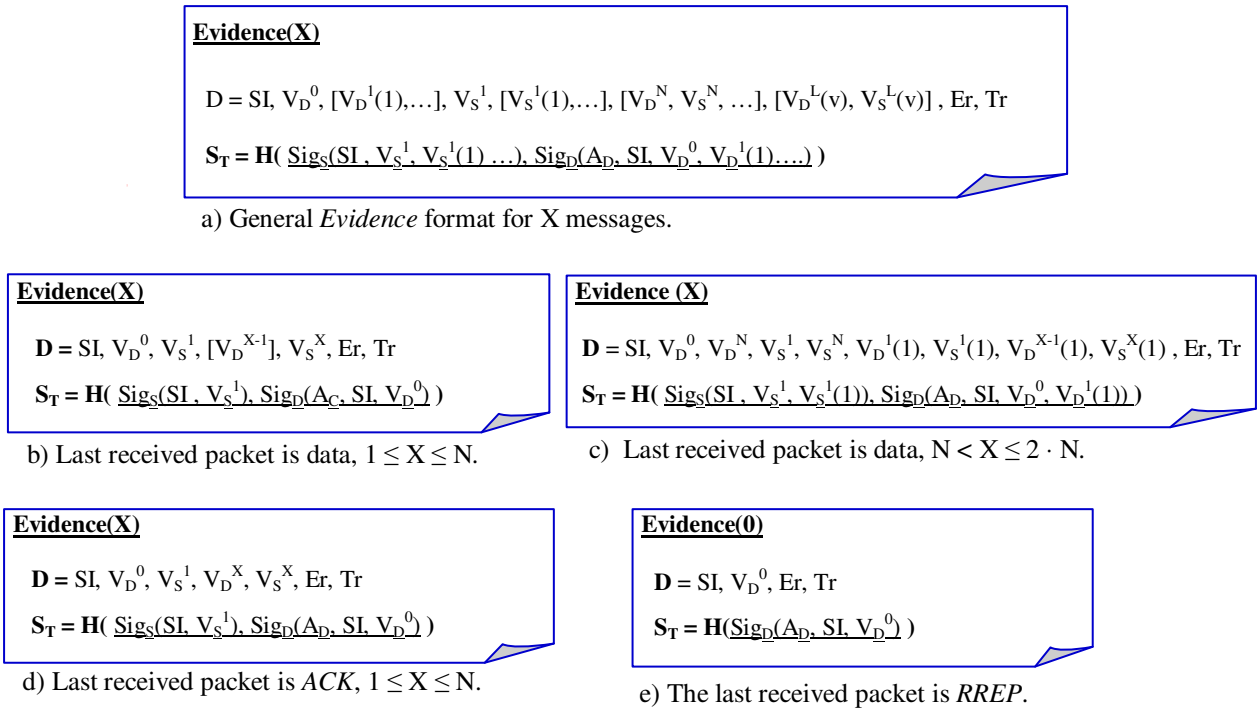


Figure 4.6: The formats of the payment *Evidences*.

Figure 4.6(b) shows the composed *Evidence* when a route is broken during relaying the *X*th data packet and $1 \le X \le N$, i.e., only one hash chain is used in the route. If the route is broken after receiving the first data packet ($X = 1$), the *Evidence* does not have V_D^1 because the *ACK* packet is not received. For $1 < X \le N$, the last hash value received from the destination node

or V_D^{X-1} is attached to the *Evidence*. Since the session is broken before receiving the *ACK* of message *X*, the last released hash value from the destination node is V_D^{X-1} and not V_D^X. If this *Evidence* is composed by a node ID_i, it is clear that ID_i has received *X* messages and relayed *X-1* messages but all the nodes before ID_i in the route have indeed relayed *X* messages. By this way, Tp can compute a trust value for each node to depict the probability that the node relays a packet. The number of delivered messages (*X-1*) can be computed from the number of hashing operations required to obtain V_D^X from hashing V_D^0, and the number of transmitted packets (*X*) can be computed from the number of hashing operations required to obtain V_S^X from hashing V_S^1.

Figure 4.6(c) shows the composed *Evidence* when a route is broken after receiving the *X*th data packet and $N < X \leq 2 \cdot N$, i.e., two hash chains are used. It can be seen that the *Evidence* contains the seed and the root of the first hash chain, the root of the second hash chain, and the last released hash value ($V_D^{X-1}(1)$ and $V_S^X(1)$). Moreover, the two hash chains' roots are included in the source and the destination nodes' signatures. Figure 4.6(d) shows the composed *Evidence* when the last received packet is the *ACK* of the *X*th message. This *Evidence* is a proof of successfully delivering *X* messages. Figure 4.6(e) shows the format of the *Evidence* when the last received packet is *RREP*.

Evidence-aggregation technique: Instead of storing an *Evidence* per session, an aggregated *Evidence* can be computed using onion-hashing technique to prove the node's credibility in a group of sessions. In Figure 4.7, D(i) and $S_T(i)$ are the descriptor and the security token of *Evidence* number i, respectively. Figure 4.7 shows that the aggregated *Evidence* contains one security token that is generated by onion hashing the individual *Evidence*s' security tokens. The onion-hashing technique enables the nodes to aggregate a recently composed *Evidence* to the aggregated *Evidence*, i.e., the *Evidence*s are always stored in aggregated format to reduce their storage area. The technique is called onion hashing because aggregating an *Evidence* adds one hashing layer. However, the *Evidence* aggregation process is irreversible, i.e., the aggregated *Evidence* cannot be split again to individual

Evidences, and thus if Tp requests an *Evidence*, the node has to submit the aggregated *Evidence* that contains the requested one. For example, if *Evidences* E_1, E_2, and E_3 are aggregated to E_C and Tp requests E_2, the node has to submit E_C and Tp has to verify the node's credibility in the three *Evidence*s. Increasing the number of aggregated *Evidence*s reduces the storage area but with more processing overhead, and bandwidth and energy for submitting the aggregated *Evidence*, if an *Evidence* is requested.

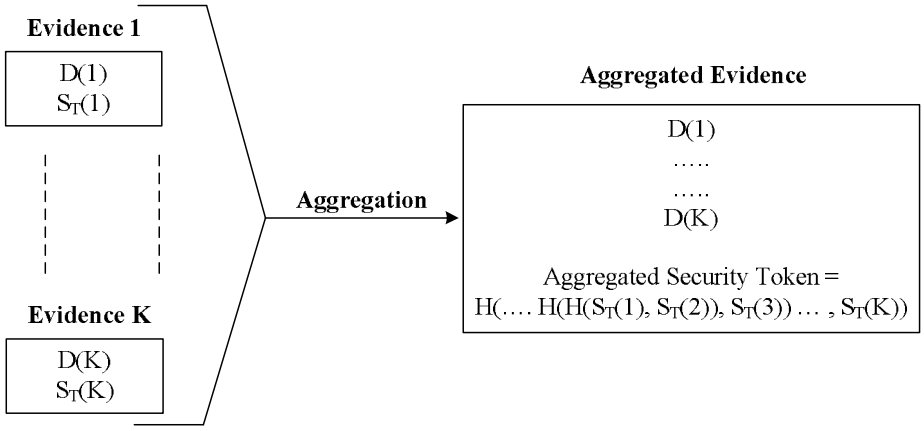


Figure 4.7: Aggregated *Evidence*.

4.3 Classifier

The network nodes periodically submit payment reports to Tp to redeem the payment. As shown in Figure 4.1, the *Classifier* classifies the payment reports to fair or cheating. It first uses the report's unique identifier (SI) to make sure that the report has not been processed before. Then, Tp classifies the fair payment reports to *Incomplete Session Report* (ISR(X)) or *Complete Session Report* (CSR(X)), where X is the messages' number. For ISR(X), the last received packet is the data packet of M_X (or F = 0). The name "incomplete" refers to the fact that the session must be broken because a node does not receive the *ACK* of M_X. For CSR(X), the last received packet is the *ACK* of M_X (or F = 1). The name "Complete" refers to the fact that the session may be complete because the node received the *ACK* of M_X.

As illustrated in Figure 4.8, if the session is broken during relaying the *RREP* packet, the payment reports submitted from the nodes that received the packet are classifies as ISR(0). ISR(0) enables Tp to identify the malicious nodes that frequently drop the *RREP* packets. If the session is broken during relaying the *X*th data packet, the reports submitted from the nodes that received the packet are classified as ISR(X). Submitting ISR(X) by node ID_i does not entail that it has relayed the *X*th data packet just that it has successfully relayed *X*-1 packets and received one, but it is clear that all the nodes before ID_i in the session from ID_S to ID_{i-1} have indeed relayed the *X*th data packet. If the session is broken after receiving the *X*th *ACK*, the reports submitted from the nodes that received the packet are classified as CSR(X).

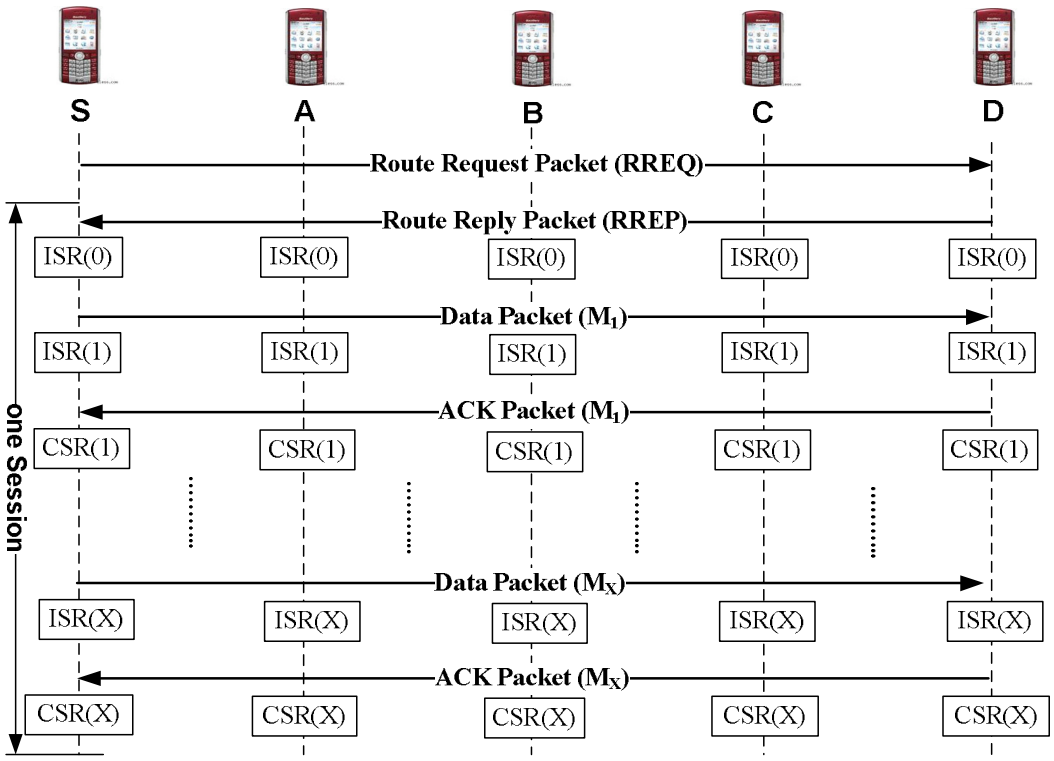
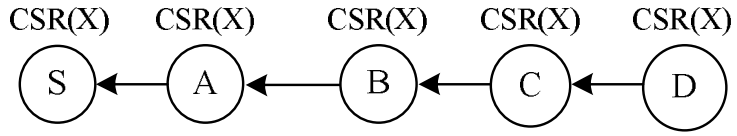


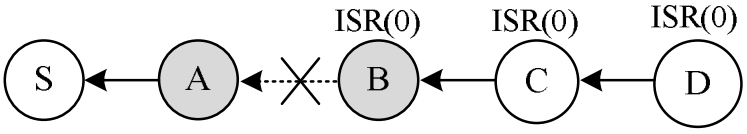
Figure 4.8: The evolution of the session payment report.

The *Classifier* verifies the reports' credibility by matching a node's report with those of the other nodes in the session and classifies a session's reports into fair or cheating. For the

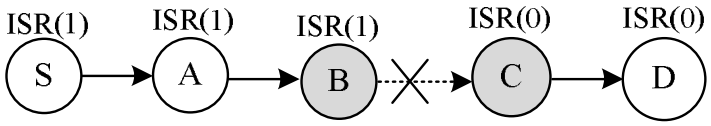
cheating reports, at least one node does not submit the payment report or submits tampered report for rational reasons such as paying less, or for irrational reasons to circumvent the trust system. In the cheating reports, the nodes' reports are inconsistent and show that the reports cannot occur without cheating from at least one node. For the fair reports, there are only five cases shown in Figure 4.9. If a session's reports are consistent with one of these cases, the session's reports are classified as fair, otherwise they are cheating.



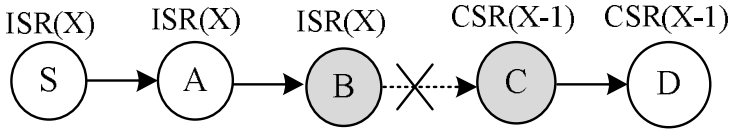
a) Complete session.



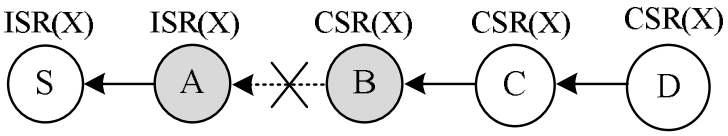
b) Broken session during relaying the *RREP* packet.



c) Broken session during relaying M_1.



d) Broken session during relaying M_X.



e) Broken session during relaying the *ACK* of M_X.

Figure 4.9: The possible cases for fair reports.

4.4 Processing

From Figure 4.1, the objective of this phase is to process the reports to extract the financial and contextual information. The financial information includes who pays whom and how much. The identities of the payers (source and destination nodes) and the payees (intermediate nodes) and the ratio of the payment are included in SI in the payment report. The numbers of the transmitted and delivered messages are included in the source and the destination nodes' reports, respectively. The contextual information reflects the nodes' misbehaviours in terms of packet drop. This information is called contextual because it is not carried in the reports but extracted from the reports' context such as the format and the messages' number. The fair reports can be for complete or broken sessions. A session is complete when the source node transmits its last message, but it is broken when at least one link is broken during the transmission of the data, *ACK*, or *RREP* packets. As illustrated in Figure 4.9(a), a session is complete when all the nodes submit *Complete Session Reports* for the same number of packets or CSR(X). Without loss of generality, Table 4.2 gives numerical examples for fair payment reports. Session number 1 in Table 4.2 gives an example for a complete session. It can be seen that F is one and X is 11 for all the reports, i.e., all the nodes received the *ACK* packet of M_{11}.

For the broken sessions, there are only four possible cases shown in Figure 4.9(b-d). Tp can identify the broken link easily from the reports' format and/or the messages' number. In Figure 4.9(b), if the link between A and B is broken during relaying the *RREP* packet, the nodes B to D submit ISR(0), but the nodes S to A do not submit reports. As shown in Session 2 in Table 4.2, X is zero and F is one for the nodes B to D. Since node A may break the session and does not submit the report to circumvent the trust system, the two nodes in the broken link are accused. In Section 4.6.2, we will discuss how the trust system can precisely differentiate between the honest and the irrational packet droppers. In Figures 4.9(c, d), the link between B and C is broken during relaying the *X*th data packet. If $X = 1$, B and C submit ISR(1) and ISR(0), otherwise they submit ISR(X) and CSR(X-1), respectively. Sessions 3

and 4 in Table 4.2 give numerical examples for these two cases. Since Session 3 was broken at B during relaying the first message, the nodes S to B report messages' number of one and F of zero but the nodes C to D report messages' number of zero and F of one. Session 4 was broken at B during relaying the data packet of M_8 because the nodes S to B report one more message and F of zero. In Figure 4.9(d), the link between A and B is broken during relaying the *ACK* of M_X, so they submit ISR(X) and CSR(X), respectively. In Session 5 in Table 4.2, 11 messages were delivered, but the *ACK* of M_{11} was dropped at B because it is the last node received the *ACK*, i.e., it reports F of one.

Table 4.2: Numerical examples for fair payment reports.

Session №		S	A	B	C	D
1	X	11	11	11	11	11
	F	1	1	1	1	1
2	X	---	---	0	0	0
	F	---	---	1	1	1
3	X	1	1	1	0	0
	F	0	0	0	1	1
4	X	8	8	8	7	7
	F	0	0	0	1	1
5	X	11	11	11	11	11
	F	0	0	1	1	1

4.5 Cheater Identification

Our security strategy is to prevent stealing credits, paying less, or manipulating the honest nodes' accounts. We should also guarantee that an honest node can get its payment even if the other nodes in the session collude. Moreover, instead of requesting the *Evidence*s from all the nodes in the cheating session, Tp should make smart decisions by requesting the *Evidence*s only from the nodes that can reveal the cheaters. The *Evidence*-request rule is as follows. If the end (source and destination) nodes' reports are consistent, the *Evidence*s are requested from the intermediate nodes that claim more payment; else, the *Evidence* is requested from the end node claiming more payment. In other words, the *Evidence*s are

always requested from the nodes claiming more payment because they cannot compose them without undeniable security tag (signature or hash chain element) generated from the cheater. We do not request the *Evidence*s from the end nodes with consistent reports because they can create them if they collude. To verify an *Evidence*'s credibility, Tp creates the *Evidence*'s security token by generating the nodes' signatures and hashing them. The *Evidence* is credible if the resultant hash value is identical to the *Evidence*'s security token. Tp also verifies the source and the destination nodes' hash chains by making sure that V_D^1 and V_S^1 are obtained from hashing V_D^X and V_S^X *X-1* times. The number of received and relayed messages by a node can be computed from the number of hashing operations required to map V_D^X and V_S^X to V_D^1 and V_S^1.

Table 4.3: Numerical examples for cheating payment reports.

Session №		S	A	B	C	D
1	X	6	10	10	10	10
2	X	5	12	12	12	5
3	X	---	4	---	---	---
4	X	9	9	3	8	8
	F	0	0	0/1	1	1
5	X	12	8	8	8	12
6	X	6	6	----	5	5
	F	0	0	----	1	1
7	X	14	14	22	14	14
8	X	7	7	7	7	6
	F	0	0	1	0	1

Without loss of generality, Table 4.3 gives numerical examples for cheating payment reports. For Session 1, the source node can compose a valid *Evidence* if it cheats because the destination node has released V_D^6. Tp can request the *Evidence* from an intermediate node or the destination node, and the source node is cheater if the *Evidence* is correct because the *Evidence* cannot be composed without releasing V_S^{10}. For Session 2, requesting the *Evidence* from an intermediate node can reveal the cheater because the *Evidence* cannot be composed

without releasing V_S^{12} and (V_D^{11} or V_D^{12}). Tp should not request the *Evidence* from the source or the destination nodes because they can compose a valid *Evidence* if they collude. For Session 3, node A cannot compose a valid *Evidence* without the source and the destination nodes' signatures and V_S^4 and (V_D^4 or V_D^3). If A submits a valid *Evidence*, S and D are evicted because they did not report the payment for a session they participated in, but B and C are punished by not rewarding them to discourage un-reporting the payment. We cannot evict the nodes B and C because S and D may exploit that B and C do not sign each message and collude to evict them. This example shows that an honest node can prove its credibility and receive its deserved payment even if the other nodes in the session collude.

For Session 4, the report of node B is inconsistent with those of the other nodes. B may break the session and report less-payment to circumvent the trust system, or the other nodes collude to accuse B. For Session 5, the source and the destination nodes claim sending and receiving more messages and the intermediate nodes claim relaying fewer number of messages. This case may be rare because the attackers' focus will be on stealing credits or paying less. The Tp can clear the payment according to the nodes' reports without requesting evidence, i.e., the source node pays more if it cheats, and the other nodes lose credits if they cheat. In this way, we can achieve our security strategy and discourage cheating. We should not request the *Evidence* from the source and the destination nodes because they can compose valid evidence if they collude. We should not punish the intermediate nodes because they can compose valid *Evidence* for eight messages, and S and D might collude to falsely accuse the intermediate nodes. However, *even if the source and the destination nodes collude, they cannot fabricate Evidence for fake sessions because the intermediate nodes' signatures are required (in A_D) to compose a valid Evidence, which is important to make False-Accusation attack difficult. The only way the attackers can falsely accuse an honest node is by neighboring the node and breaking the session or paying more credits by submitting reports for more messages such as Sessions 4 and 5 in Table 4.3*. For Session 6, node B breaks the session and does not submit the report or the other nodes collude to falsely

accuse B. In this case, the colluders have to neighbor node B to compose a valid A_D that contains B's signature and use A_D to compose the *Evidence*. For Session 7, node B can prove its credibility if its *Evidence* contains V_S^{22} and (V_D^{22} or V_D^{21}). For Session 8, node B claims delivering seven messages but D claims receiving only six messages. If node B is honest, its *Evidence* should have V_D^7.

4.6 Credit-Account and Trust Update

4.6.1 Credit-Account Update

Tp clears the payment of the fair and corrected payment reports according to the charging and rewarding policy discussed in Section 3.3. For the cheating reports, Tp clears the payment in such a way that prevents stealing credits and punishes the cheating nodes to discourage cheating actions. For example, Session 1 in Table 4.3 is cleared for ten messages if S cheat, and Sessions 2 is cleared for twelve messages if S and D cheat. For Session 3, node A is rewarded for four messages, B and C are not rewarded because they do not submit the report, and S and D are charged for four messages. For Sessions 4 to 6, each node is rewarded or charged according to the payment in its report, so the payee that does not submit a report is not rewarded, the payee that submits less-payment report is rewarded less, and the payers that submit more-payment reports are charged more. In this way, *the nodes that submit incorrect payment reports always lose credits*.

4.6.2 Trust Update

As shown in Figure 4.1, Tp updates the nodes' trust values after processing the payment reports and extracting the contextual information. Tp uses a trust system to calculate a trust value for each node. The trust system performs the following two processes: (1) For *Rating Calculation*, a rating is calculated for each node in the session; (2) For *Trust Update*, a node's trust value is updated by aggregating its session rating with its old trust value.

A. Rating Calculation

A node's rating is the probability that the node is the session breaker, so the nodes that are not in a broken link receive positive rating (0) because they cannot be the session breakers. In other words, all the nodes in the complete sessions receive positive ratings, and the nodes that are not in the broken link in a broken session receive positive ratings. For the Sessions 4 to 6 in Table 4.3, all the nodes receive negative ratings (1), i.e., *if an attacker manipulates its payment reports, he loses credits and also receives negative ratings*. If A and B are in a broken link, Tp cannot accuse only A of breaking the session because B may break the session and compose a valid *Evidence* for *X-1* messages instead of *X* messages to circumvent the trust system, i.e., we consider that A and B received *X* messages but relayed only *X-1* messages. *The rationale here is that the nodes that break the sessions more frequently will be accused more and thus suffer from more trust degradation.* Moreover, an honest node can protect its trust values by not involving itself in sessions with a neighbour that frequently drops the packets. The neighbours of the malicious nodes change due to the node mobility and thus the accusations are distributed instead of focusing them on few nodes. Two techniques, called simple and weighted rating, are proposed to calculate the negative ratings of the two nodes in a broken link.

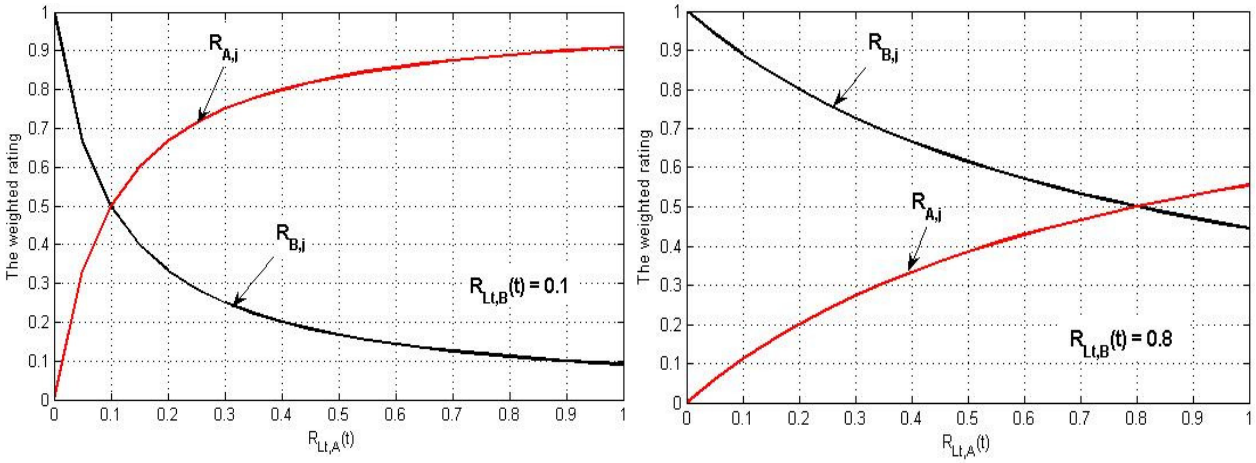
Simple Rating Technique (SRT): The two nodes in a broken link receive equal negative ratings of (1), i.e., the broken link's nodes are equally accused regardless of their session breakage history. *The rational of this technique is that the irrational packet droppers should be involved in much more broken links than the honest nodes to launch effective attacks, so they can be identified because they collect much more negative ratings*. Obviously, the technique is called simple because it requires simple computations and small storage area.

Weighted Rating Technique (WRT): The two nodes in a broken link receive ratings that are proportional to their past session breakage frequency. If the link between nodes A and B is broken in session j, Equation 4.1 is used to calculate the rating of node A ($R_{A,j}$) which is

the ratio of A's long-term reputation value ($R_{Lt,A}(t)$) to the summation of the two nodes' reputation values. $R_{Lt,A}(t)$ is computed from a large number of ratings and reflects the probability that the node breaks a session. By the same way, the rating of B ($R_{B,j}$) is its reputation value to the summation of the two nodes' reputation values, or ($1-R_{A,j}$).

$$R_{A,j} = \frac{R_{Lt,A}(t)}{R_{Lt,A}(t) + R_{Lt,B}(t)} \quad (4.1)$$

$$R_{B,j} = \frac{R_{Lt,B}(t)}{R_{Lt,A}(t) + R_{Lt,B}(t)} \quad (4.2)$$



a) B is an honest node. b) B is an irrational packet dropper.

Figure 4.10: The weighted ratings for two nodes in a broken link.

As shown in Figure 4.10, if A and B have the same reputation value, i.e., $R_{Lt,A}(t) = R_{Lt,B}(t)$, they receive equal negative ratings of 0.5, but the node with worse (higher) reputation value receives more negative rating and vice versa. *The rational of this technique is that the worse-reputation node is more likely the session breaker because it has been involved in more broken links.* The main advantage of the *WRT* is that if honest and malicious nodes are involved in a broken link, they receive low and high negative ratings, respectively, which can improve the trust/reputation system's effectiveness because the malicious nodes' reputation values degrade much faster than those of the honest nodes do. In other words, the malicious nodes cannot cause big reduction in the honest nodes' reputations

but the honest nodes can cause big reduction in the malicious nodes' reputations. In Figure 4.10(a), B is an honest node because its reputation value is low. If A is also honest, e.g., with a reputation value between 0.05 and 0.15, the two nodes receive ratings between 0.4 and 0.6. However, if A is a malicious node, e.g., with a reputation value of 0.8, A and B receive ratings of 0.89 and 0.11, respectively. In Figure 4.10(b), B is a malicious node with a reputation value of 0.8. If A is also malicious with a reputation value close to 0.8, the ratings of the two nodes is around 0.5. In other words, a malicious node receives less and more negative ratings when it neighbours malicious and honest nodes, respectively, so a malicious node can be identified in shorter time when it neighbours more honest nodes because its reputation can be degraded faster. *Due to this property, if the honest nodes' number is larger than that of the malicious nodes, the WRT can accelerate the degradation of the malicious nodes' reputations without the need to know the expected reputation values for honest nodes which may not be easy*, i.e., the well-behaving majority can kick out the misbehaving minority from the network.

B. Trust Update

Using a reputation/trust system is necessary to keep track of a node's long-term behaviour because sessions may be broken normally, e.g., due to the nodes' mobility, or temporarily, e.g., due to the network congestion, but the high frequency of session breakage is an obvious misbehaviour. The reputation system computes a reputation value for each node by accumulating its ratings. A rating is an evaluation to the node's behaviour in one session, but the reputation value is an evaluation to the node's behaviour over a large number of sessions. From Figure 4.11, the reputation system stores a rating window for the latest γ ratings of node A, where $R_{A,j}$ is the rating of A in session number j, and $R_{A,j} \in \{0, 1\}$ and [0, 1] in *SRT* and *WRT*, respectively. After computing a new rating, the rating window is shifted to right to cancel the oldest rating ($R_{A,j}$), and the new one is stored at right. Then, with Equation 4.3, the short-term reputation value ($R_{St,A}(t)$) is calculated by averaging the node's latest γ ratings, which is an evaluation to the node's behaviour in the latest γ sessions.

Finally, with Equation 4.4, the new long-term reputation value ($R_{Lt,A}(t)$) is calculated by aggregating $R_{St,A}(t)$ with the old long-term reputation ($R_{Lt,A}(t-1)$), where $R_{St,A}(t)$, $R_{Lt,A}(t)$ and $\alpha \in [0, 1]$. $R_{Lt,A}(t)$ expresses the probability that A is an irrational packet dropper, i.e., $R_{Lt,A}(t)$ should be large for the irrational packet droppers. α is called the fading factor that determines the given weight to the nodes' past behaviour. The value of α determines how fast the long-term reputation builds up and falls down, i.e., the lower value α has, the faster the long-term reputation is forgotten, and vice versa. To improve the reputation system's effectiveness, α should be greater than α-1 because $R_{Lt,A}(t-1)$ is calculated over more sessions than $R_{St,A}(t)$.

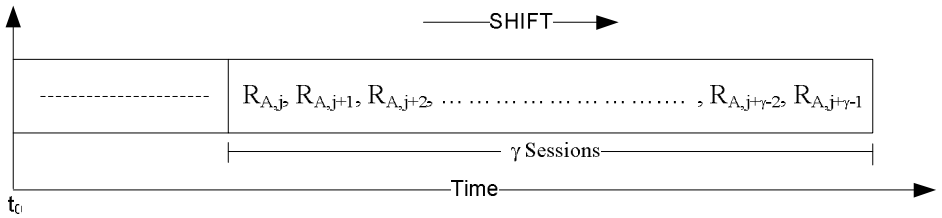


Figure 4.11: The rating window of node A.

$$R_{St,A}(t) = \frac{1}{\gamma} \cdot \sum_{k=1}^{\gamma} R_{A,k} \tag{4.3}$$

$$R_{Lt,A}(t) = \alpha \cdot R_{Lt,A}(t-1) + (1-\alpha) \cdot R_{St,A}(t) \tag{4.4}$$

A node's reputation value is updated by $R_{St,A}(t)$ (the average of the latest γ ratings) instead of only the latest rating (good or bad) to better differentiate between the honest and the malicious nodes. In this way, *the long-term reputations of the honest and the malicious nodes degrade slower and faster when they receive negative ratings because their short-term reputations are smaller and larger, respectively; and the long-term reputations of the honest and the malicious nodes improve faster and slower when they receive positive ratings because their short-term reputations are smaller and larger, respectively*. Moreover, the honest nodes can filter out their negative ratings in two levels: shifting the rating window forgets the node's behaviour in one session, and using α forgets a ratio of the node's past

behaviour.

The notion of trust used in this thesis is defined as the probability that a node will act in a certain way based on the node's past actions. A trust relationship is never absolute, i.e., a node's trust value reflects its ability to perform a specific action. We represent a trust value with a numeric value in the range [0, +1] signifying a continuous range from complete distrust (0) to complete trust (+1). Three trust values, $T_{1,A}(t)$, $T_{2,A}(t)$, and $T_{3,A}(t)$, can be calculated to precisely evaluate node A's behavior. $T_{1,A}(t)$ depicts the probability that A can relay a message successfully. From Equation 4.5, $T_{1,A}(t)$ is the number of relayed messages by A to the number of received messages in the last δ sessions. Obviously, if A drops a large percentage of the messages such as *Black* and *Gray Hole* attackers, $T_{1,A}(t)$ will be very low. $T_{2,A}(t)$ depicts the probability that A does not break a session. Since $R_{LT,A}(t)$ is the probability that node A breaks a session, $T_{2,A}(t)$ is 1- $R_{LT,A}(t)$ as given in Equation 4.6. If A breaks a large percentage of the sessions, e.g., due to high mobility, $T_{2,A}(t)$ will be very low. If $T_{2,A}(t)$ is low, that does not necessarily mean that $T_{1,A}(t)$ is low too, e.g., the nodes that break many sessions but after relaying a large number of messages. $T_{3,A}(t)$ is the probability that node A can relay at least ψ messages in a session. From Equation 4.7, $T_{3,A}(t)$ is the percentage of the sessions that A relayed at least ψ messages in the last δ sessions. $T_{3,A}(t)$ gives information about A's capability of relaying a minimum number of messages in one session. The trust values are calculated only for the last δ sessions, e.g., 100 to 200 sessions, because recent steady behavior is a better predictor for future behavior than behavior observed a long time ago.

$$T_{1,A}(t) = \alpha \cdot T_{1,A}(t-1) + (1-\alpha) \cdot \frac{\text{№ of relayed messages in the last } \delta \text{ sessions}}{\text{№ of received messages in the last } \delta \text{ sessions}} \quad (4.5)$$

$$T_{2,A}(t) = 1 - R_{Lt,A}(t) \quad (4.6)$$

$$T_{3,A}(t) = \alpha \cdot T_{3,A}(t-1) + (1-\alpha) \cdot \frac{\text{№ of sessions node A relayed at least } \psi \text{ messages}}{\delta} \quad (4.7)$$

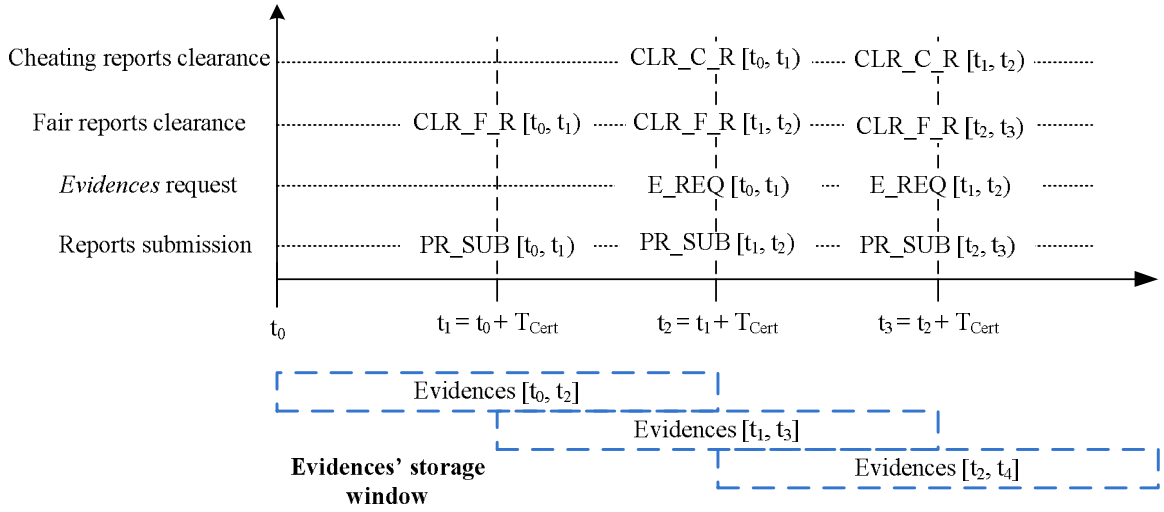


Figure 4.12: The timing of report submission and clearance, and *Evidence* request.

4.6.3 Trust-Update and Payment-Clearance Delay

Unlike receipt-based incentive protocols, clearing the payment (and updating the trust values) once a node submits its payment report is not guaranteed because Tp has to wait until receiving the reports of the other nodes in the session. The payment clearance and trust update delay (P_C) is the elapsed time from a session's occurrence until the payment is cleared and the trust values are updated. Figure 4.12 shows the timing of the payment report submission (PR_SUB), *Evidence* request (E_REQ), clearance of the fair and cheating reports (CLR_F_R and CLR_C_R), and the *Evidence*s' storage window. In the figure, we assume that all the nodes contact Tp every certificate lifetime (T_{Cert}) to simplify our presentation. Node A joined the network at time t_0 and submitted its first report at t_1 for the sessions held in the period $[t_0, t_1)$. At t_1, the fair reports of the sessions held in $[t_0, t_1)$ can be cleared because Tp received all the reports. At t_2, node A submitted its second reports for the sessions held in $[t_1, t_2)$. Tp may request *Evidence*s for the sessions held in $[t_0, t_1)$, and thus the cheating reports for the sessions held in $[t_0, t_1)$ can be cleared. At t_2, node A deletes all the *Evidence*s for the sessions held in $[t_0, t_1)$ because Tp must have cleared them. P_C is bounded by T_{Cert} for the fair

reports and $2 \cdot T_{Cert}$ for the cheating reports because Tp has to wait one extra T_{Cert} for requesting the *Evidence*s. It is obvious that each node has to store the *Evidence*s for $2 \cdot T_{Cert}$.

However, in reality, the nodes may contact Tp at different times and the connection intervals may be variant in the range $(0, T_{Cert}]$ because a connection to Tp may not be available on regular basis. Hence, P_C may be less than T_{Cert}. If a node does not report a session in time T_{Cert}, the session is cleared such as Session 6 in Table 4.3. To estimate the average P_C, we assume that the nodes submit the payment reports when they are involved in a large number of sessions or their certificates are near to expire to reduce the communication overhead. We model this behavior with truncated exponential distribution with parameters Δ and T_{Cert}. T_i is a random variable denoting the time between each two report submissions of node i, where $T_i \in (0, T_{Cert}]$. Equation 4.8 gives the probability that T_i is at most t. A session's payment is cleared and trust values are updated when all the nodes in the session submit their reports. $P_C(R_L)$ is a random variable that denotes the payment clearance delay for a session with R_L nodes, where $P_C(R_L) \in (0, T_{Cert}]$. Equation 4.9 gives the probability that $P_C(R_L)$ is at most t. Equations 4.10 and 4.11 give the probability density function and the average $P_C(R_L)$, respectively.

$$P(T_i \leq t) = \frac{1 - e^{-\Delta \cdot t}}{1 - e^{-\Delta \cdot T_{Cert}}} \tag{4.8}$$

$$P(P_C(R_L) \leq t) = \prod_{i=1}^{i=R_L} P(T_i \leq t) = \left(\frac{1 - e^{-\Delta \cdot t}}{1 - e^{-\Delta \cdot T_{Cert}}}\right)^{R_L} \tag{4.9}$$

$$f(P_C(R_L)) = e^{-\Delta \cdot t} \cdot \Delta \cdot R_L \cdot \frac{(1 - e^{-\Delta \cdot t})^{R_L - 1}}{(1 - e^{-\Delta \cdot T_{Cert}})^{R_L}} \tag{4.10}$$

$$E(P_C(R_L)) = \int_0^{T_{Cert}} t \cdot f(P_C(R_L))\, dt \tag{4.11}$$

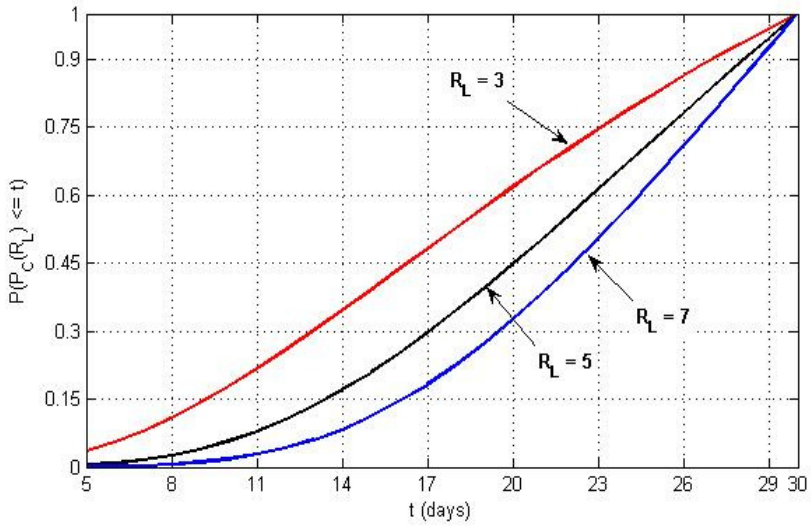


Figure 4.13: $P(P_C(R_L) \leq t)$ VS t at different values of R_L.

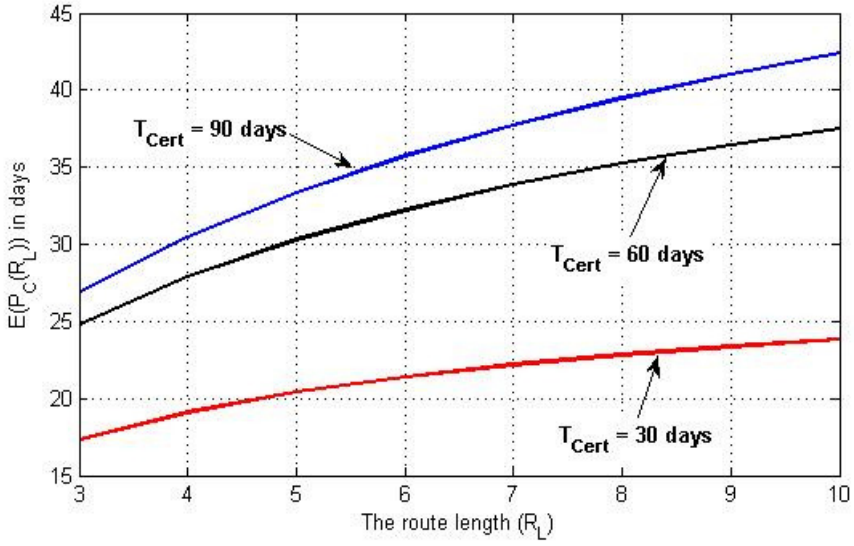


Figure 4.14: The average payment clearance delay at different values of R_L.

Δ is 1/15 in Figures 4.13 and 4.14 and T_{Cert} is 30 days in Figure 4.13. Figure 4.13 shows that the increase of the route length (R_L) decreases the probability of clearing the payment and updating the trust values by time t. Figure 4.14 shows that the average payment clearance delay can be less than T_{Cert}, e.g., at R_L of 5 and T_{Cert} of 60, the average delay is 30 days. This delay is acceptable because the nodes do not need to wait until gaining credits to communicate, i.e., the nodes communicate first and pay later and they can buy credits for real

money. Figure 4.14 shows that shorter T_{Cert} can reduce the delay but with more processing overhead on Tp for renewing the certificates.

4.7 Identification of Irrational Packet-Droppers

A node's state is a conclusion for its behaviour based on the accumulated experience on it (or its reputation value), which can also be an expectation to its behaviour in the future. A node's state space includes three mutually disjoint states: honest or regular node (+1), suspicious or undecided (0), and malicious or irrational packet dropper (-1). From Equation 4.12, the state of node A ($S_A(t)$) is honest if the node's long-term reputation value is below the honest threshold R_h; $S_A(t)$ is malicious if the node's long-term reputation value is above the malicious threshold R_m; otherwise, $S_A(t)$ is suspicious. Moreover, a node is identified as malicious when it spends ω consecutive sessions in the suspicious state because the node receives negative ratings more than the normal rate. A node is also identified as malicious when the difference between the spent times in the honest and the suspicious states is less than β because the node receives positive ratings less than the normal rate. The state transition diagram for a node is shown in Figure 4.15. A suspicious node may be honest but its reputation is degraded temporarily, so instead of taking a harsh reaction by characterizing this node as malicious, the reputation system keeps collecting information about the node's behaviour to figure out whether its misbehaviour is temporary or genuine. If a suspicious node is honest, it should be able to improve its reputation and return to the honest state, but the irrational packet dropper stays some time in the suspicious state before it is transferred to the malicious state. As shown in the figure, a node is transferred directly from the honest to the cheating state without passing through the suspicious state when it commits a clear cheating action such as the source node in Session 1 in Table 4.3.

$$S_A(t) = \begin{cases} +1, & R_{Lt,A}(t) < R_h & \text{(Honest)} \\ 0, & R_h \le R_{Lt,A}(t) \le R_m & \text{(Suspicious)} \\ -1, & R_{Lt,A}(t) > R_m & \text{(Malicious)} \end{cases} \quad (4.12)$$

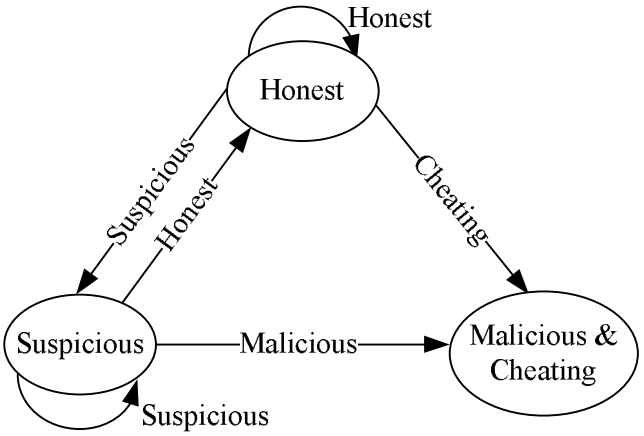


Figure 4.15: A node's state transition diagram.

The threshold R_m can reveal the attackers that break sessions more than the normal rate and the threshold β can reveal the attackers that break a large number of consecutive sessions such as the broken nodes that misbehave after gaining good reputation. Moreover, the threshold ω can reveal the attackers that spend a long time in the suspicious state such as *Gray-Hole* attackers. Since it is impossible to know whether a packet is dropped normally or intentionally, the attackers may drop packets with keeping their reputations above the reputation system's thresholds. If the thresholds are close enough to the normal rate, the reputation system can force the attackers to break sessions at a lower rate than the system's thresholds to avoid eviction, i.e., the system can force the smart attackers to behave in such a way that is not severe threat to the network proper operation. In Section 4.8, we will propose a trust-based routing protocol to establish the routes through the highly-trusted nodes, and thus the honest nodes with relatively low trust values have low chance to participate in routes.

R_h enables an honest node to filter out its negative ratings because the node is identified honest as long as its reputation value is less than R_h. The reputation system tolerates the degradation of an honest node's reputation up to R_m provided that the node improves its reputation and returns to the honest state. Actually, there is an intuitive trade-off between the

required time to detect the malicious nodes and the number of honest nodes that are falsely identified as malicious, which can be controlled by the thresholds. For larger R_m, the reputation system tolerates more misbehaviours and reduces the false accusations but at the expense of longer detection time because the malicious nodes have to break more sessions to be identified. The *Black-Hole* attackers can be identified in short time because their reputation values degrade fast, but it takes longer to identify the *Gray-Hole* attackers because they behave honestly for a while to build up their reputations.

4.8 Trust-Based and Energy-Aware Routing Protocols

Although the nodes having reputation values less than R_h are considered honest, the honest nodes have different packet-drop probabilities, i.e., the nodes with high hardware-capability and low mobility have less packet-drop ratio. In this section, we propose two routing protocols called the *Shortest Reliable Route* (*SRR*) and the *Best Available Route* (*BAR*) to route the packets through the highly-trusted nodes having sufficient energy to minimize the probability of route breakage. The SRR protocol establishes the shortest route that meets the source node's energy and trust requirements, but the destination node selects the best route in the BAR protocol.

Route reliability can be computed from the trust values of the nodes en route to get probabilistic information about the route stability and lifetime, which can be used in route selection. Equation 4.13 gives the probability that a transmitted message will be delivered to the destination node across the intermediate nodes W, X, Y, and Z. In the same way, $T_{2,WXYZ}(t)$ and $T_{3,WXYZ}(t)$ given in Equations 4.14 and 4.15 are the probability that the session will not be broken and the probability that at least ω messages will be transmitted along the route, respectively. Comparing the reliabilities of Routes 1 and 2 in Table 4.4, the low-trust node, such as X in route 2, has very little chance to be involved in a session because it significantly degrades the route reliability. Although the nodes' trust values of Route 3 are

the same as those of Route 1, Route 3 has higher reliability, which demonstrates that shortest routes are preferable. The probability to deliver a message through Route 4 is close to zero because the nodes have very low trust values, which demonstrates the importance of choosing good nodes.

$$T_{1,WXYZ}(t) = T_{1,W}(t) \cdot T_{1,X}(t) \cdot T_{1,Y}(t) \cdot T_{1,Z}(t) \quad (4.13)$$

$$T_{2,WXYZ}(t) = T_{2,W}(t) \cdot T_{2,X}(t) \cdot T_{2,Y}(t) \cdot T_{2,Z}(t) \quad (4.14)$$

$$T_{3,WXYZ}(t) = T_{3,W}(t) \cdot T_{3,X}(t) \cdot T_{3,Y}(t) \cdot T_{3,Z}(t) \quad (4.15)$$

Table 4.4: Numerical examples for route reliability.

Route №	$T_{1,W}(t)$	$T_{1,X}(t)$	$T_{1,Y}(t)$	$T_{1,Z}(t)$	$T_{1,WXYZ}(t)$
1	0.8	0.8	0.8	0.8	0.4096
2	0.8	0.2	0.8	0.8	0.1024
3	0.8	0.8	0.8	----	0.512
4	0.2	0.2	0.2	0.2	0.0016

4.8.1 SRR Routing Protocol

***RREQ Delivery*:** In order to establish an end-to-end route, the source node broadcasts the *Route Request Packet* (*RREQ*) that contains its identity (ID_S) and certificate ($Cert_S$), the destination node's identity (ID_D), Time-To-Live (TTL) or the maximum number of hops, the session establishment time stamp (Ts), the payment-splitting ratio (Pr), the trust requirement (Tr), the energy requirement (Er), and its signature $A_S = Sig_S(ID_S, ID_D, Ts, Pr, Tr, Er)$. The source node is charged the ratio of Pr of the total payment and the destination node is charged the ratio of 1-Pr. Tr is the minimum trust values an intermediate node can have, and Er is the minimum number of messages an intermediate node commits to relay in the session, which is related to the node's residual energy. If a node breaks the route before relaying Er messages, the node's trust values are decreased. The route reliability is bounded by Tr raised to the TTL-th power.

After receiving the *RREQ* packet, a node first checks that the time stamp is within a proper range, TTL is not zero, and its trust values are at least Tr. The node broadcasts the *RREQ* packet after attaching its identity and certificate, signing the packet signature, and decrementing the TTL, as shown in Figure 4.16. This signature authenticates the node and holds it accountable for dropping the next packets, i.e., Tp can make sure that the node has indeed participated in the session. In Figure 4.16, $A_A = Sig_A(A_S)$, $A_B = Sig_B(A_A)$, and $A_C = Sig_C(A_B)$. The certificates enable the nodes to verify the signatures and prove that the nodes are member in the network, which is important to prevent the external attackers from participating in the network. Only the first *RREQ* is broadcasted and the subsequent requests for the same session are discarded. The source node's requirements cannot be achieved if it does not receive the *RREP* within τ_S time period. The source node can send a second round of the *RREQ* after reducing its requirements, or revert to the *BAR* protocol. *The rationale of the SRR protocol is that the node that satisfies the source node's requirements is trusted enough to act as an intermediate nodes.*

---------- RREQ ----------

S $\rightarrow$ * : <**RREQ**, ID_S, $Cert_S$, ID_D, Ts, Pr, Tr, Er, TTL, A_S>

A $\rightarrow$ * : <**RREQ**, ID_S, $Cert_S$, ID_D, Ts, Pr, Tr, Er, **ID_A**, **$Cert_A$**, **TTL-1**, **A_A**>

B $\rightarrow$ * : <**RREQ**, ID_S, $Cert_S$, ID_D, Ts, Pr, Tr, Er, ID_A, $Cert_A$, **ID_B**, **$Cert_B$**, **TTL-2**, **A_B**>

C $\rightarrow$ * : <**RREQ**, ID_S, $Cert_S$, ID_D, Ts, Pr, Tr, Er, ID_A, $Cert_A$, ID_B, $Cert_B$, **ID_C**, **$Cert_C$**, **TTL-3**, **A_C**>

D : <**RREQ**, ID_S, $Cert_S$, ID_D, Ts, Pr, Tr, Er, ID_A, $Cert_A$, ID_B, $Cert_B$, ID_C, $Cert_C$, TTL-3, A_C>

---------- RREP ----------

D $\rightarrow$ C: <**RREP**, SI, $Cert_D$, V_D^0, A_D>

C $\rightarrow$ B: <**RREP**, SI, $Cert_D$, V_D^0, A_D, $Cert_C$>

B $\rightarrow$ A: <**RREP**, SI, $Cert_D$, V_D^0, A_D, $Cert_C$, $Cert_B$>

B $\rightarrow$ A: <**RREP**, SI, $Cert_D$, V_D^0, A_D, $Cert_C$, $Cert_B$, $Cert_A$>

S : <**RREP**, SI, $Cert_D$, V_D^0, A_D, $Cert_C$, $Cert_B$, $Cert_A$>

Figure 4.16: The route establishment packets.

***Route Selection*:** The destination node receives *RREQ* packets for different routes to the source. The first received *RREQ* packet is the shortest route that achieves the source node's requirements. If the verification of the first arrived packet's signature (A_C in Figure 4.16) fails, the destination node verifies the signature of the second packet and so on. In this way, if ID_i manipulates A_i, it cannot prevent establishing the session. The destination node signs the *RREQ* packet's signature to generate the session nodes' authentication code or A_D, e.g., $A_D = Sig_D(A_C, SI, V_D^0)$ in Figure 4.16, and sends the *Route Reply Packet* (*RREP*) containing V_D^0, its certificate, A_D, and the session identifier (SI), where $SI = ID_S, ID_A, ID_B, ID_C, ID_D$, Ts, Pr in Figure 4.16. A_D can authenticate the session nodes with reduced packet overhead because it requires less space than attaching a separate signature for each node. A_D authenticates the destination node's hash chain and links it to the session, and proves the node's approval to pay for the session.

***RREP Delivery*:** Each intermediate node verifies the *RREP* packet's signatures to authenticate the nodes between itself and the destination node. For example, in Figure 4.16, node B authenticates C and D from their signatures in the *RREP* packet and authenticates S and A from the *RREQ* packet signature. This signature verification process is necessary to make sure that A_D is correct, and thus to ensure the *Evidence*'s integrity and protect the payment. Each intermediate node relays the *RREP* packet after adding its certificate. It also saves A_D and V_D^0 to be used in the evidence composition. If a node lies in its residual energy, the route will be broken at this node and thus its trust values degrade. The source and the destination nodes verify the intermediate nodes' certificates to make sure that the nodes can indeed achieve the minimum trust requirements.

4.8.2 BAR Routing Protocol

***RREQ Delivery*:** The *RREQ* packet is the same as that of the *SRR* protocol but Tr is the route reliability field (*R_reliability*) that is initialized to one and $C_m(S)$ is attached instead of Er, where $C_m(S)$ is the expected number of transmitted messages. For the first received *RREQ*

packet, an intermediate node A adds the number of messages it commits to relay ($C_m(A)$), and updates the *R_reliability* by multiplying it with its trust value. Er is the minimum message's number committed by the nodes in the route. Blind flooding generates few routes because each node broadcasts the *RREQ* once, which disables potential better routes. To solve this issue, the *BAR* allows each node to broadcast the *RREQ* more than once if the *R_reliability* or the Er of the recently received packet is greater than the last broadcasted packet. For example, in Figure 4.17, node M receives the first *RREQ* at time t_1 with *R_reliability* $T_{1,AB}$ of 0.3. At t_2, M broadcasts the packet after updating the *R_reliability* to be $T_{1,ABM}$, where $T_{1,ABM} = T_{1,AB} \cdot T_{1,M}$. At t_3, M receives the second *RREQ* packet for the same session with *R_reliability* of $T_{1,NFK}$ that is less than $T_{1,AB}$, so it discards the *RREQ* packet. At t_4, M receives the *RREQ* packet with *R_reliability* of $T_{1,XYZW}$ that is larger than the last broadcasted packet, so it broadcasts the packet at t_5 after updating the *R_reliability*. In this example, we calculate the *R_reliability* only for the first trust value for simplicity. However, it can also be calculated for the other trust values.

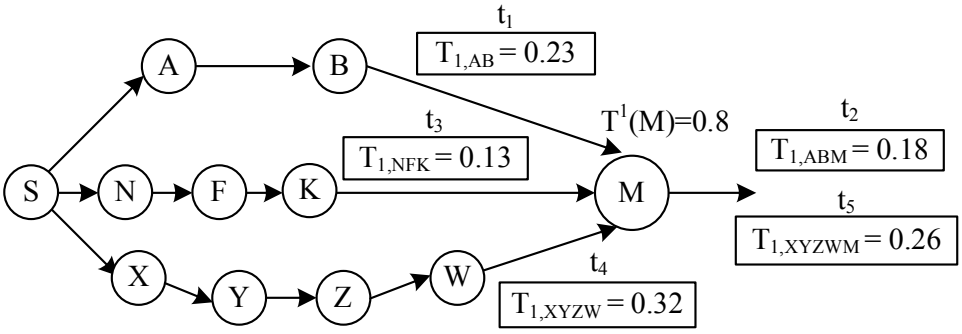


Figure 4.17: Broadcasting the RREQ packets in the BAR routing protocol.

***Route Selection*:** After receiving the first *RREQ* packet, the destination node waits for τ_D time window to keep receiving other *RREQ* packets if there are, and then selects the best available route. First, the destination node excludes the routes with very low reliability. If there are multiple routes with Er at least $C_m(S)$, the destination node selects the most reliable route, otherwise, the destination node establishes multiple routes with a total Er of $C_m(S)$ or more in such a way that reduces the routes' number and maximizes the reliability. For

example, in Figure 4.18, the destination node D receives two possible routes (A, B, C) and (X, Y, Z, W) and chooses the most reliable one, assuming the two routes' Er are $C_m(S)$ or more.

***RREP Delivery*:** This phase is identical to that of the *SRR* routing protocol.

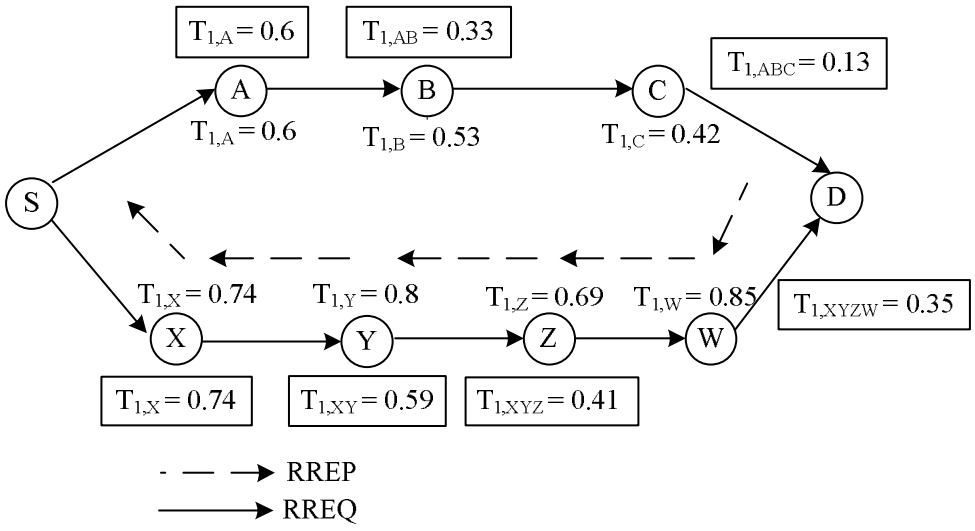


Figure 4.18: Route selection in the *BAR* routing protocol.

Chapter 5

PRIPO: Privacy-Preserving Routing and Incentive Protocol

5.1 Preliminary

***Network Model*:** In this chapter, we propose a **P**rivacy-preserving **R**outing and **I**ncentive Pr**O**tocol called **PRIPO** for hybrid ad hoc wireless network. Unlike the proposed protocol in Chapter 4, all the communication has to be relayed through at least one base station, i.e., we consider only the hybrid communication mode. 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. A mobile node X should register with Tp to get a permanent shared symmetric key K_X and a unique identity ID_X.

***Payment Model*:** 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 *ACK* packet from D. In Chapter 6, we will argue that this rewarding and charging policy can discourage cheating actions and encourage the nodes' cooperation without submitting payment receipts or reports from the nodes.

***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. Location privacy is defined as the ability to prevent other parties from learning one's current and past locations [63], and anonymity is defined as the state of being un-identifiable within a set of subjects called the anonymity set [64].

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 node's 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 the 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.

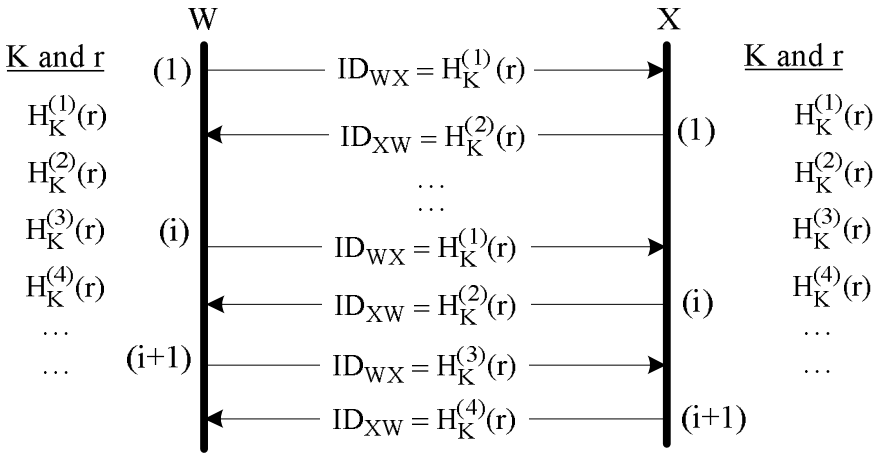


Figure 5.1: Pseudonyms generation technique.

5.2 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 Figure 5.1, 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 $H_K^{(j)}(r)$ refers to the message authentication code resulted from iteratively hashing r j 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., if i is one, each pseudonym is used for one packet.

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' numbers 1 to i in Figure 5.1, 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 releasing $H_K^{(4)}(r)$. The main advantage of this pseudonym generation technique is that the nodes do not have to change their pseudonyms at a given frequency, but this change can be arbitrarily triggered by X or W without losing synchronization. The technique is efficient because a pseudonym generation requires a 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 uses 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 Figure 5.2, 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 the key 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}.

X → Bs→ Tp: <*AREQ*, ID_{XTp}, $(ID_{XTp}, ID_X)K_X$>

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

Bs → X: <*AREP*, ID_{TpX}, $(r, K_{XBs})K_X$>

Figure 5.2: Authentication phase.

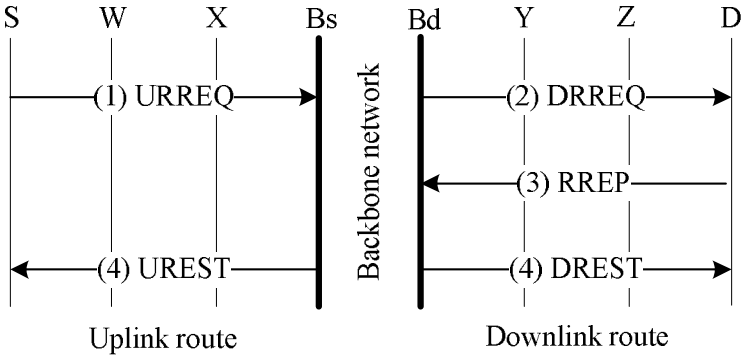


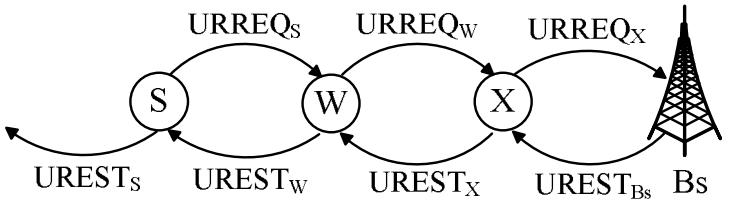
Figure 5.3: Route Establishment phase.

5.3 Route Establishment Phase

As shown in Figure 5.3, 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 *Downlink Route Establishment* (*UREST* and *DREST*) *packets* to establish the uplink and downlink routes, respectively.

URREQ: As shown in Figure 5.4, the *URREQ* packet contains dummy bits called padding (Pad) [65] and the encryption of the source and the destination nodes' real identities, the padding length (P_L), and a unique request identifier (Uni). Uni 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 neighbours 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

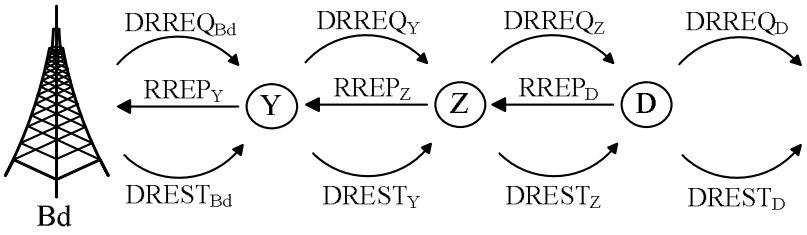
stores Uni in the routing table and drops any further requests with the same identifier to broadcast the request once and avoid routing loops. For the first *URREQ* packet, Bs decrypts the encryption part to know the real identity of the destination node and the padding length, and forwards the request to D.



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>

Figure 5.4: Anonymous uplink route establishment.



DRREQ$_{Bd}$: <Dn$_i$, TTL, (Dn$_i$, ID$_D$, ID$_S$)K$_{BdD}$>
DRREQ$_Y$: <Dn$_i$, TTL-1, (Dn$_i$, ID$_D$, ID$_S$)K$_{BdD}$, ID$_{YBd}$>
DRREQ$_Z$: <Dn$_i$, TTL-2, (Dn$_i$, ID$_D$, ID$_S$)K$_{BdD}$, ID$_{YBd}$, ID$_{ZBd}$>
DRREQ$_D$: <Dn$_i$, TTL-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>

Figure 5.5: Anonymous downlink route establishment.

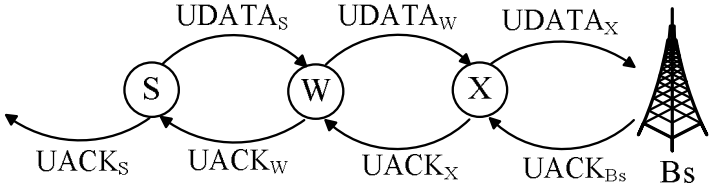
***DRREQ*:** As shown in Figure 5.5, the *DRREQ* packet contains Time-To-Live (TTL), a unique request identifier (Dn_i) that contains the pseudonym shared with D and time stamp, and the real identity of the source node, encrypted with the shared key with D. Bd does not add padding because we do not aim to preserve the base station's location privacy. Each intermediate node adds its pseudonym shared with Bd and broadcasts the packet if TTL is greater than zero. The node stores Dn_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 to know whether it is the destination. This is important because the *DRREQ* packets are received by a large number of nodes.

***DRREP*:** Figure 5.5 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 Figure 5.4, 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 the 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 node.

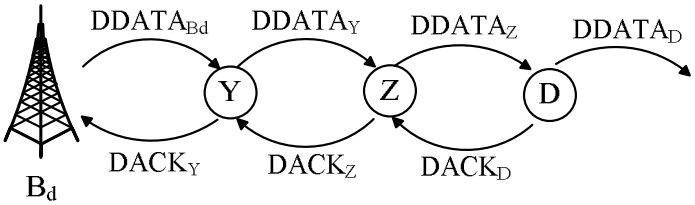
***DREST*:** From Figure 5.5, the format of *DREST* packet is the same as the *UREST* packet.



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$_{XBs}$>

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>

Figure 5.6: Anonymous uplink data transmission.



UDATA$_{Bd}$: <ID$_{BdY}$, (((C, M$_C$, H$_{KDBd}$(M$_C$))K$_{DBd}$)K$_{YZ}$)K$_{BdY}$
UDATA$_X$: <ID$_{YZ}$, ((C, M$_C$, H$_{KDBd}$(M$_C$))K$_{DBd}$)K$_{YZ}$
UDATA$_Y$: <ID$_{ZD}$, (C, M$_C$, H$_{KDBd}$(M$_C$))K$_{DBd}$
UDATA$_D$: <Random>

DACK$_D$:<ID$_{DZ}$, (H$_{KDBd}$(M$_C$))K$_{DBd}$>
DACK$_Z$:<ID$_{ZY}$, ((H$_{KDBd}$(M$_C$))K$_{DBd}$)K$_{ZY}$>
DACK$_Y$: <ID$_{YBd}$, (((H$_{KDBd}$(M$_C$))K$_{DBd}$)K$_{ZY}$)K$_{YBd}$>

Figure 5.7: Anonymous downlink data transmission.

5.4 Data Transfer Phase

As shown in Figure 5.6, 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 Bs. 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 and checks the message integrity, and forwards it to the destination base station.

From Figure 5.7, 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 iteratively encrypted data 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. The uplink and downlink acknowledgement packets (UACK and DACK) are shown in Figures 5.6 and 5.7, respectively. In this way, each intermediate node performs only one encryption or decryption operation but the base stations perform more operation. PRIPO can be used for bidirectional communication without any modification. The overhead of the data packets is only one pseudonym instead of attaching the whole route identities similar to DSR routing protocol [66].

5.5 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.

Chapter 6

Security Analysis

6.1 Security Analysis for ESIP

6.1.1 Defence against Payment Manipulation

Our security objective is to prevent the attackers from achieving gains such as stealing credits or paying less. In our incentive protocol, the charges and rewards are based on payment reports submitted by autonomous nodes, so a node or even a group of colluding nodes may attempt to cheat the protocol to increase their welfare.

For *Free-Calling (or Riding)* attacks, two colluding intermediate nodes in a legitimate session manipulate the session packets to piggyback their data to communicate freely. To thwart this attack, the messages' integrity and authenticity are checked at each node by verifying the message's keyed hash value, and thus the first node after the colluder can detect any addition or modification to the packets and drop them. The *RREQ* and *RREP* packets' integrity and authenticity can be verified at each node by verifying the signatures. Launching *Free Riding* attacks against the *ACK* packets is not possible because each *ACK* contains a hash-chain element that is verified at each node and thus the packet's integrity and

authenticity can be verified at each node. In a malicious attack, node A may manipulate the hash value of D to consume the nodes' resources because the packet will be dropped at D and thus C and D are unfairly accused of dropping the packet. To simplify our presentation in Chapter 4, we consider that a keyed hash value covers only the message, but it should cover the whole packet to thwart this attack. For example, in Figure 4.5, the keyed hash value of node B should be $H_{KSB}(M_X, V_S^X, H_{KSC}(M_X), H_{KSD}(M_X))$, so that node B can stop propagating the packets with incorrect hash values.

It is obvious that the packet overhead will be large for long routes because the source node attaches a keyed hash value for each node in the route. To reduce the packet overhead, the message's keyed hash value can be truncated significantly, e.g., the size of the truncated hash value (η) can be 4 or 5 bytes instead of 16 bytes in HMAC-MD5. This severe hash truncation is secure in our protocol for the following reasons: (1) The packet security lifetime is extremely short, i.e., if an intermediate node does not relay a packet in a short time, the route is considered broken and re-established, so a malicious node does not have long time to run complicated programs to figure out the truncated keyed hash values for the manipulated message; (2) Without knowing the secret key, computing the keyed hash value is not possible; and (3) An attacker has to figure out a keyed hash value for each victim between itself and the other colluder. *Therefore, an attacker has to compute multiple truncated keyed hash values without knowing the keys in a limited time, which is not possible*. What an attacker can do is to replace the truncated hash with a random value, but the probability to hit the correct value is extremely low, e.g., for $\eta = 4$ bytes, the probabilities to hit one and two correct hash values are 0.23×10^{-9} and 0.05×10^{-18}, respectively. Moreover, if the manipulated hash value is not correct, the attacker's neighbour drops the packet and thus the attackers' trust values are degraded.

However, the hash truncation increases the random collision probability, i.e., the corrupted and the original messages have the same truncated keyed hash value. Using birthday paradox, the random collision probabilities for η of 4 and 5 bytes are 1.2×10^{-5} and

7.63×10^{-7}, respectively. In addition, since the message integrity is checked at each node, the probability that the destination node falsely accept a corrupted message as correct is ($n_1 \cdot 1.2 \cdot 10^{-5}$) for η of 4, which is equivalent to the probability that the hash collision occurs in n_1 successive nodes, where n_1 is the number of nodes from the node at which the message is corrupted to the destination node. This probability can be reduced with the increase of η but the packet overhead increases, so η can be dynamic to balance the probability of falsely accepting corrupted message and the packet overhead, i.e., η can be longer for short routes. Moreover, some nodes in a route can have longer η than others, e.g., η can be longer for the destination node to prevent falsely accepting corrupted messages. MD5 is faster and has shorter digest length than SHA-2, but SHA-2 is more collision resistant, so SHA-2 can be used in digital signing operations that require high collision resistance, and MD5 is used to compute the keyed hashes and the hash chain.

For *Double-Rewarding* attack, the attacker attempts to clear a payment report multiple times to be rewarded multiple times for the same session. Tp can thwart the attack and identify the attacker because it uses the report's unique identifier (SI) to check whether the report has been processed before. For *Double-Spending* attack, the attackers attempt to generate identical reports for different sessions to pay once. In ESIP, the reports cannot be repeated because each report contains the identities of the route nodes and time stamp. Even if the attackers establish different routes at the same time, the reports' identifiers are different because at least one intermediate node is different.

For *Evidence-Forgery-and-Manipulation* attack, the attackers attempt to forge an *Evidence* or manipulate a valid *Evidence* to steal credits. This is not possible using secure hash function and public key cryptography because it is not possible to forge or modify the nodes' signatures and to compute the private keys from the public ones. It is also not possible to compute the hash value of the nodes' signatures without computing the signatures, and to compute V_D^X from V_D^{X-1} or V_S^X from V_S^{X-1}. Moreover, if an attacker attaches a random value for an *Evidence's* security token, the probability to hit the correct value is nearly zero,

e.g., this probability is 6.84×10^{-49} with using SHA-1 [67, 68] with digest value of 20 bytes, Tp can identify the attackers when their *Evidence*s' verifications fail. The intermediate nodes can verify the source and the destination nodes' signatures and hash chain, which is important to verify the *Evidence*s and thus secure the payment. If the intermediate nodes cannot verify the payment data, the source node may send packets with invalid payment data.

For *Packet-Replay* attacks, the internal or 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 *ESIP*, a fresh time stamp is used in the *RREQ* packet to establish a route and thus stale *RREQ* packets can be identified and dropped if the time stamp is not within a proper range. For *Impersonation* attack, the attackers attempt to impersonate others to communicate freely or steal credits. This attack is not possible because the nodes use their private keys to sign the packets, and the attackers cannot compute others' private keys. For *Message-Repudiation* attack, the attackers attempt to deny transmitting a message. In *ESIP*, each node can ensure that the intended user has sent a message, but unlike signature-based protocols, it cannot prove that to a third party. However, message non-repudiation is important for other applications such as electronic commerce where a user sends messages to authorize the recipients to perform actions on its behalf. For *Payment-Repudiation* attacks, the attackers attempt to deny initiating a session or the amount of payment so as not to pay. In *ESIP*, the payers cannot deny the payment because the signatures and the hash chains can guarantee the payment non-repudiation.

For *Fake-Intermediate-Node* attack, the attackers may claim that they have non-existent neighbours to make the source and the destination nodes pay more, collect credits to the non-existent neighbour, or falsely accuse the non-existent node of breaking the session or un-submitting the session report. The nodes in a session authenticate themselves to thwart this attack and external attacks [69] that are launched by the external attackers who are not members in the network and the nodes' authentication code (A_D) is included in the *Evidence*, e.g., the external attackers participate in route discovery phase with the intention of dropping

the data packets to launch *Denial-of-Service* attacks. Moreover, the payment is cleared without punishing or rewarding the victim nodes such as Session 6 in Table 4.3. However, the colluding nodes may exchange their cryptographic information to insert non-existent neighbours to collect credits for them without participation in relaying the packets. This attack is a type of the known routing attack called *Route-Lengthening*. First, exchanging the cryptographic information 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, Tp 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 [70] and Ariadne [71] can be implemented in *ESIP*.

For *Destination-Node-Robbery* attack, the source node colludes with some intermediate nodes to steal credits from the destination node by sending bogus messages paid by the source and destination nodes, or fabricating or manipulating evidences. For example, from Figure 2.1, if the source node colludes with K intermediate nodes, the intermediate nodes earn $(X \cdot \lambda \cdot K)$ credits but the source node pays $(X \cdot \lambda \cdot n \cdot Pr)$ for X packets. Obviously, the colluders can achieve gains when $(X \cdot \lambda \cdot K - X \cdot \lambda \cdot n \cdot Pr) > 0$ or $(K - n \cdot Pr) > 0$. In *ESIP*, the colluding nodes cannot fake or manipulate *Evidence*s to steal credits from the destination node because the destination node's signature and hash chain elements are required to compose a valid *Evidence*. Moreover, a session cannot be established and a valid *Evidence* cannot be composed if the destination node is not interested in the communication because its signature is required. The intermediate nodes are rewarded only when the destination node acknowledges receiving correct messages, and thus they do not earn from the bogus messages. For *Denial-of-Report-Submission* attack, the colluding attackers residing close to the base station may attempt to prevent the nodes to submit the payment reports to Tp. First, the nodes can accumulate the reports and submit them to Tp in batch to reduce the communication overhead, e.g., the reports may be submitted every few days, and thus the

nodes can repeatedly try to contact Tp during this period. Second, the nodes transmit the *Report-Submission-Request* packet to Tp to submit the reports. This packet contains the identity of the submitter, time stamp, the payment reports, and the submitter's signature. The signature authenticates the submitter, thwarts packet replay attack, and ensures the packet's integrity. Tp replies with the *Reports-Submission-Confirmation* packet containing Tp's signature for the reports to confirm the reports' submission. The node does not delete the reports before receiving the *Reports-Submission-Confirmation* packet. Third, the nodes may change their cells due to the nodes' mobility, and thus if they cannot submit the reports through one base station, they can submit the reports through others. Finally, the nodes can contact Tp using wired networks such as Internet and Wi-Fi, as explained in Section 3.1.

For *Reduced-Payment* attack, some intermediate nodes may collude with the source and the destination nodes to submit payment reports with less payment. In *ESIP*, even if a group of nodes colludes to reduce an honest node's rewards, the node can prove its credibility and gets its correct payment, such as Sessions 3 and 7 in Table 4.3. In our payment model, the source and the destination nodes can communicate even if they do not have sufficient credits, so for *Payment-Denial* attacks, the attackers may join the network for a short time and leave without paying their debts. Different from the traditional ad hoc networks that can be temporarily established and similar to the current single hop cellular networks, MWN is a long life network where the nodes have long-term relations with the network. The post-paid payment policy has been widely used in many services successfully such as credit cards and cellular networks. In *ESIP*, each node needs a certificate to participate in the network. Issuing a certificate is not free to make changing identity costly. Moreover, similar to the existing single hop cellular networks, Tp stores the personal information of the users, and thus it can take the legal procedures against the users who do not pay. To limit overspending, a node's certificate lifetime is short and the lifetime can depend on the node's available credits at the certificate issuing time and its average credit consumption rate. We can also mix both the pre-paid and the post-paid payment policies to reduce the debt amount, e.g., each node has to

pay some money in advance at certificate renewal.

Without proper charging and rewarding policy, the rational attackers may try to cheat to increase their welfare. Our charging and rewarding policy has been developed to counteract the rational cheating actions and encourage the nodes' cooperation. Particularly, a rational node can exhibit one of the following actions:

1. If the intermediate nodes are rewarded for relaying the messages that do not reach the destination node [34, 35], the colluding nodes can drop a message and relay only the smaller-size security tag (hash chains' elements) that is much shorter than the message to claim the payment for relaying the message with consuming low resources because they can compose a valid *Evidence*. Our payment model encourages the nodes to rely the messages because they are rewarded only when the destination node acknowledges receiving the messages. Moreover, the attackers' trust values degrade when they drop a packet.
2. If the source and the destination nodes are charged only for the successfully delivered messages, the destination node may receive a message but does not send *ACK* so as not to pay, or an intermediate node colludes with the source and the destination nodes to claim that the message does not reach the destination to increase their welfare. To thwart this cheating action, both the source and the destination nodes are charged for the un-delivered messages.

Although the charges are always more than or equal to the rewards, our payment model does not make credits disappear because purchasing credits with real money can compensate the credit loss. The rich nodes that have much more credits than their credit consumption may stop cooperation to save their resources. Tp converts credits to real money to motivate these nodes to cooperate because this conversion reduces the nodes' credits. Due to using post-paid payment policy, the nodes can communicate even if they do not have sufficient credits at the communication time.

6.1.2 Defence against Trust Manipulation and Irrational Packet Drop

The objectives of using trust in route selection are as follows: (1) To foster trust among the nodes by making knowledge about the nodes' past behaviors available; (2) To encourage the nodes to provide high packet-relay success ratios and tell the truth about their residual energy by giving more preference to the highly trusted nodes in route selection; and (3) To punish the nodes that provide low packet-relay success ratios because any loss of trust means loss of potential earnings. Our trust/reputation system can precisely judge the nodes' behaviors because it can monitor the nodes' behaviors over different sessions and long time, but the reputation-based mechanisms [14, 43-47] may not have sufficient time to judge the nodes' real behavior as the period of interaction with any node may be brief due to the nodes' mobility. Moreover, the nodes are motivated not to cooperate in reputation-based mechanisms because packet relay consumes their resource without benefits, but packet-relay is beneficial for the nodes in our protocol to earn credits. Reputation-based mechanisms use thresholds to differentiate between the honest and the malicious nodes, but in our protocol, once a node's trust values fall behind those of the majority of the nodes, the node almost does not participate in routing without the need for determining accurate thresholds.

Reputation and trust systems are susceptible to collusion attacks due to the nature of these systems, e.g., the colluding nodes may fill up an honest node's rating window with negative ratings to evict the node from the network, and fill up their rating windows with positive ratings to avoid eviction. In our trust system, the singular attackers cannot launch the *Trust-Boost* attacks. For *False-Accusation* attack, the attacker has to neighbour the victim node and break the session intentionally to let Tp accuse its neighbour. First, neighbouring a node is not easy due to the nodes' mobility. Second, the attacker is also accused of breaking the session and receives negative rating, which may discourage the attack. Third, frequently launching the attack reduces its effectiveness because the attacker will be less frequently selected in routes due to its low trust. Frequently launching *False-Accusation* attack also reduces the attack's effectiveness in WRT because the honest nodes and the attackers are

offered less and more negative ratings, respectively. Finally, falsely accusing a node does not guarantee that this accusation will be effective because the node can filter out its negative ratings and improve its trust from other sessions.

The impact of small-scale collusion attacks can be mitigated by categorizing ratings by identities. The reputation system can construct neighbour density tables (NDTs) for the negative and positive ratings in the rating windows. The negative rating density of node B in the NDT of node A is the number of negative ratings that B was neighbour to A to the total number of negative ratings in the rating window of A. By the same way, the positive rating density of node B in the NDT of node A is the number of positive ratings that B was neighbour to A to the total number of positive ratings in the rating window of A. In other words, the NDT can show the frequency that B caused positive and negative ratings to A. Obviously, in small-scale collusion attacks, the colluders have much higher densities than those of other nodes. Investigating the NDT for deciding a node's state can improve the reputation system's robustness. For example, in *Reputation-Boost* and *False-Accusation* attacks, few nodes have high densities in a node's positive and negative ratings, respectively, and the node's reputation becomes bad and good with excluding these false ratings, respectively. The NDT can prevent a small number of colluders from falsely improving their reputation values or evicting an innocent node from the network, and thus forces the attackers to collude with a large number of nodes, which is not easy in civilian and large-scale networks [54], [55]. Certainly, if the NDT's densities are flat or dominated by a large number of nodes, the reputation system can have a strong belief about the node's real behaviour.

Several measures can be taken to improve the robustness against large-scale collusion attacks. Clearance fee can be imposed to clear a session's payment to discourage submitting reports for fake complete sessions to launch *Trust-Boost* attack, i.e., to make fabricating sessions by the colluding nodes to boost their trust values expensive. If colluders tamper their payment reports to accuse a victim, they lose credits and defame their reputation such as

Sessions 4 to 6 in Table 4.3. Although the honest nodes may receive negative ratings when they neighbour malicious nodes, the neighbours change due to the nodes' mobility, which can distribute the negative ratings instead of concentrating them on few nodes. Moreover, since dropping the *RREQ* packets is not abuse, an honest node can protect its reputation by not involving itself in sessions with the neighbours that frequently drop packets.

Equation 6.1 gives the probability that a node receives at least a ratio of R_m negative ratings in γ sessions, i.e., the probability of identifying a node as malicious, where P is the probability of receiving a negative rating in a session. Obviously, P should be much larger for the malicious nodes than the honest nodes because they break the sessions more frequently. Figure 6.1 shows that if R_m is chosen in the range [0.35, 0.55], the reputation system can perfectly differentiate between the malicious and the honest nodes. However, if R_m is low, e.g. [0, 0.35), some honest nodes may be falsely identified as malicious, and if R_m is too tolerant, e.g. (0.55, 1], some malicious nodes may not be identified. Thus, R_m can control the tradeoff between the false accusation probability and the malicious nodes' detection probability. The increase of P increases $P_i(\gamma)$ for the same R_m, and thus some honest nodes may be falsely identified as malicious if R_m does not have enough tolerance.

$$P_i(\gamma) = \sum_{k=\gamma.Rm}^{\gamma} \binom{\gamma}{k} \cdot P^k \cdot (1-P)^{\gamma-k} \qquad (6.1)$$

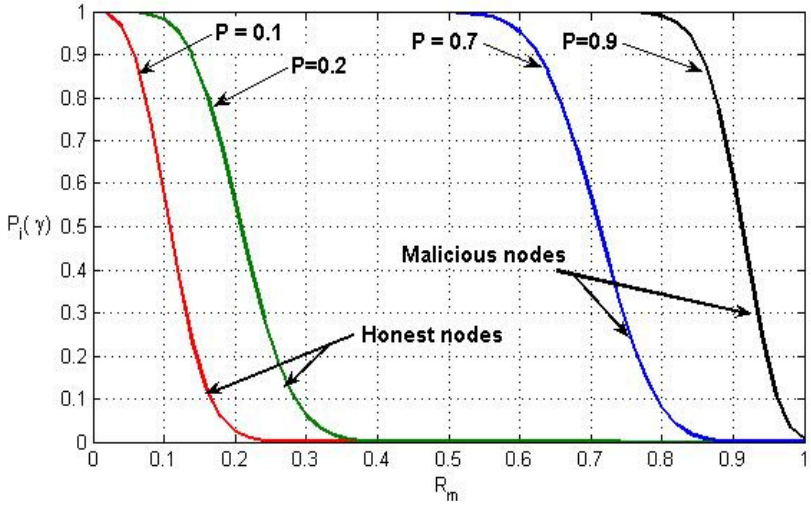


Figure 6.1: The effect of R_m on the reputation system's effectiveness, $\gamma = 50$.

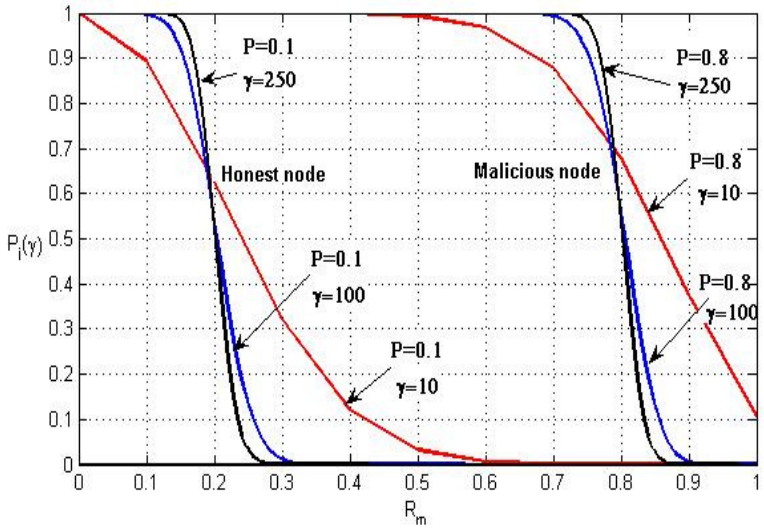


Figure 6.2: The effect of γ on the reputation system's effectiveness.

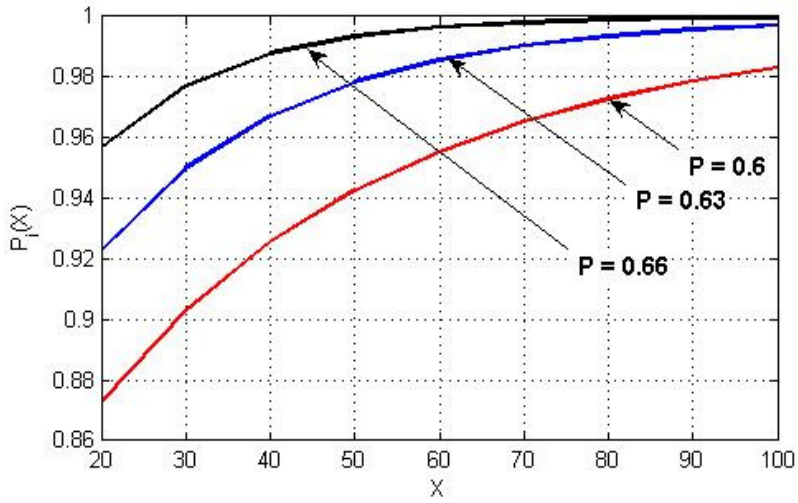


Figure 6.3: The effect of P on $P_i(X)$ for R_m of 0.5.

From Figure 6.2, the honest and the malicious nodes can be identified more precisely with increasing the rating window size (γ) because $P_i(\gamma)$ is less for the honest nodes and more for the malicious nodes, which can reduce the number of honest nodes that falsely identified as malicious and the number of malicious nodes that are not detected. For example, if R_m is in the range [0.3, 0.7], some honest nodes may be falsely identified as malicious and some malicious nodes may not be detected when $\gamma = 10$, but the reputation system can perfectly identify the nodes' behaviours when γ is 100 or 250. From Figure 6.3, the aggressive

attackers that break a large percentage of the sessions, i.e., having large P, can be identified after they participate in a low number of sessions (or in shorter time). For example, the probabilities to identify the malicious nodes after participating in 20 sessions are 0.87, 0.92, and 0.96 for P of 0.6, 0.63, and 0.66 respectively.

In order to evaluate the effectiveness of identifying the malicious nodes, a network simulator is programmed using MATLAB. 35 mobile nodes with 125 m radio transmission range are randomly deployed in a square cell of 1000 m by 1000 m. We adopt the modified random waypoint model [72] to emulate the nodes' mobility. Specifically, a node travels towards a random destination uniformly selected within the network field; upon reaching the destination, it pauses for some time; and the process repeats itself afterwards. The node's speed is uniformly distributed in the range [0, 10] m/s and the pause time is 10 s. The constant-bit-rate traffic source is implemented in each node as an application layer, and the source and destination pairs are chosen randomly. The DSR routing protocol [66] is simulated over an ideal channel, i.e., all the nodes within a transmission range receive the packets correctly. MATLAB is used instead of NS2 because the intention is to compare between the SRT and WRT. The effect of the un-simulated models such as non-ideal channel, channel contention, node buffer, etc, should be the same on the two techniques. Moreover, the reputation system's thresholds can be adjusted to absorb the expected increase of the negative ratings with considering the effect of the un-simulated models. 300 sessions are held in each updating time, packet transmission rate is 0.5 packets/s, and 25 packets are transmitted in each session. A new route is established when the session route is broken. The parameters R_h, ω, β, γ, and α are 0.19, 100, 100, 50, and 0.78, respectively. The initial rating window is the repeat of '00001', and thus the nodes' initial reputation values and states are 0.2 and honest.

In Table 6.1, the number of false-positive nodes is the average number of the honest nodes that are falsely identified as malicious, and the detection time is the average updating times for identifying all the irrational packet droppers. Attack strength 1:X means that the

attackers break one session intentionally and behave normally for X-1 sessions. However, in order to launch effective attacks, the attacker has to break a large percentage of the sessions, i.e., X should be small. When X = 1, the attackers launch *Black-Hole* attacks by dropping all the packets they should relay, otherwise they launch *Gray-Hole* attacks. The simulation results demonstrate that R_m can control the tradeoff between the detection time and the number of false-positive nodes. Less tolerance to the negative ratings (R_m = 0.35) shortens the detection time but increases the number of false-positive nodes. This tradeoff is sharper in SRT because the honest nodes collect more negative ratings. It takes longer to identify the nodes that misbehave less frequently such as the nodes with 1:2 attacking strength because they lose their reputations slowly, but they also harm the network less. The increase of the attackers' ratio increases the number of false-positive nodes because the honest nodes collect more negative ratings due to neighboring more malicious nodes, and the victims could not improve their reputation values with the same rate they degrade. Nevertheless, for R_m = 0.35 and X = 1, when a large ratio of 42.86% (16 nodes) of the nodes behave maliciously, almost no node is falsely accused in the WRT but around 13.25 nodes (37.8%) are falsely accused in the SRT. That is because the honest nodes receive less negative ratings in WRT, i.e., the *WRT can prevent the malicious nodes from degrading the honest nodes' reputations and can better filter out the honest nodes' negative ratings.*

Table 6.1: Simulation Results.

Attack strength	**R_m**	**Attackers' ratio**	**Detection time (in updating times)**		**False-positive nodes' number (γ = 50)**		**False-positive nodes' number (γ = 15)**	
			SRT	**WRT**	**SRT**	**WRT**	**SRT**	**WRT**
1:1	**0.35**	**5%**	1	1.85	2.8	0	8.85	0.16
		42.86%	1.9	4.4	13.35	0.1	16.17	0.56
	0.6	**5%**	2.25	14	0	0	0.9	0
		42.86%	5.3	98	0.85	0	2.85	0
1:2	**0.35**	**5%**	1.15	10.95	3.6	0	11.35	0.72
		42.86%	2	23.65	10	0	12.6	2.24
	0.6	**5%**	38.8	102.85	0	0	1.75	0
		42.86%	95.75	109.8	0.2	0	5.95	0

The number of false-positive nodes can be reduced in the SRT by increasing R_m, e.g., for X = 1 and 42.86% of the nodes are malicious, increasing R_m from 0.35 to 0.6 reduces the number of false-positive nodes from 13.35 (37.8%) to 2.8 (8%). However, the increase of R_m means that the smart attackers can break more sessions with keeping their reputation values above the system thresholds and the detection time increases. *R_m can be less in the WRT, e.g., R_m = 0.35, for small detection time and number of false-positive nodes because the malicious and honest nodes collect more and less negative ratings, respectively comparing with the SRT*. Moreover, the simulation results demonstrate that the increase of the attackers' ratio increases the detection time because some malicious nodes may not participate in sessions during an updating time and the malicious nodes receive less negative ratings due to neighboring more malicious nodes in the WRT. In addition, the number of false-positive nodes increases with reducing γ, which confirms the observation shown in Figure 6.2.

Since the reputation system's thresholds have direct impact on the system's effectiveness, our centralized reputation system can compute the thresholds from the nodes' reputation values and periodically tune them to improve the system's effectiveness. For example, if most of the nodes' reputation values are less than 0.3, R_h can be decided as 0.3 assuming that the majority of the nodes behave honestly. Moreover, since the nodes contact the Tp over discrete times, the detection and eviction times can be reduced with issuing shorter-lifetime certificates to bad-reputation nodes. Investigating the reputation's rate of change ($dR_{LT,i}(t)/dt$) can reduce the detection time of some attackers, e.g., the reputation's rate of change is higher for the *Gray-Hole* attackers than those of the honest nodes.

6.2 Security Analysis for PRIPO

6.2.1 Defence 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, if *Free-Riding* attack is launched by the two colluding nodes A and C, the iterative encryption/decryption operations can thwart the attack because the data sent by A cannot be interpreted by C 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 subsection 6.2.2. For *Packet-Replay* attack, if an 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 neighbour. 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.

Impersonation 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.

To thwart the *Destination-Node-Robbery* attack, 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. Most the existing incentive protocols are vulnerable to this attack because they use post-paid payment policy, i.e., the nodes communicate first and pay later. This attack cannot be launched in *PRIPO* because the base stations know the nodes' total credits from Tp during the authentication phase, and thus they will not allow the node to overspend its credits.

The payment model can encourage the nodes' cooperation and counteract cheating actions without submitting payment receipts or reports 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.
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.

6.2.2 Defence 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 the future. In data transfer phase, if an attacker could link an onion data to a node, this will not help in the future because the onion data will look different even if the same data 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 the 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 the eavesdropping nodes cannot learn the real identities of S and D and their locations in number of hops.

Pseudonym De-synchronization: In Section 5.2, 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, some 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 borders because they are less frequently selected by the routing protocol. The attackers may initiate *RREQ* packets to learn whether a node is still in its neighborhood. The proposed protocol for establishing the uplink route shown in Figure 5.4 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 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

Chapter 7

Performance Evaluations

In this section, simulation results are given to evaluate the overhead cost and the expected network performance of the proposed protocols. First, we evaluate the effect of replacing signatures with hashing operations. Second, we evaluate the reduction in the overhead due to using payment reports instead of receipts. Third, we evaluate the proposed trust-based and energy-aware routing protocols. Finally, the proposed privacy-preserving routing and incentive protocol is evaluated.

7.1 Replacing Signatures with Hashing Operations

7.1.1 Simulation Setup

We use 1024-bit RSA and 1024-bit DSA with signature tags of 128 and 40 bytes, respectively, because the secure private keys should have at least 1024 bits according to NIST guidelines [73]. For the hash functions, we use MD5 and HMAC-MD5 [67] with digest length of 16 bytes and SHA-1 hash function with digest length of 20 bytes [67], [68]. For the

pairing operation, we consider the Tata pairing implementation on MNT curves where G is represented by 171 bits, and the order P is represented by 170 bits. The discrete logarithm in G is as hard as the discrete logarithm in Z_p^* where P = 1024 bits. Network simulator NS2 is used to implement *ESIP* and signature-based incentive protocols that use the public-key operations in each packet.

We simulate multi-hop wireless network by randomly deploying 35 mobile nodes with 125m radio transmission range in a square cell of 1000 m × 1000 m. The Distributed Coordination Function (DCF) of IEEE 802.11 is implemented as the medium access control (MAC) layer protocol. The transmission data rate is 2 Mbits/s. To emulate the node mobility, we adopt the modified random waypoint model [72] with speed and pause times uniformly distributed in the ranges [0, 10] m/s and [0, 50] s, respectively. Specifically, a node travels towards a random destination uniformly selected within the network field; upon reaching the destination, it pauses for some time; and the process repeats itself afterwards. The constant bit rate (CBR) traffic source is implemented in each node, and the source and destination pairs are randomly chosen. All the data packets are 512 bytes and sent at the rate of 2 packets/s. The time stamp and an identity are five and four bytes, respectively. Each simulation is performed 50 runs, and each run is executed for 15 simulated minutes. The averaged results are presented with 95% confidence interval.

In order to estimate the expected processing times of the cryptographic primitives, we have implemented the cryptographic primitives using Crypto++5 library [74] in a laptop with an Intel processor at 1.6 GHZ and 1 GB RAM. From Table 7.1, although the signature tags of the DSA signature scheme are shorter than that of the RSA, these schemes increase the end-to-end delay significantly because the verifying operations performed by the intermediate and the destination nodes are computationally more demanding than the signing operations performed by the source node. Moreover, the energy consumption of the RSA and SHA-1 operations are measured in [75], [76] and the results are given in Table 7.1. The resources of a real mobile node may be less than a laptop, so the results given in Table 7.1 are scaled with

the factor of the five in our simulations to estimate a limited-resource node.

From Table 7.1, we can see that the computational times of the signing and verifying operations are sufficient for 539 and 18 512-byte hashing operations, respectively; and the consumed energy for the signing and verifying operations are sufficient for 1404 and 41 512-byte hashing operations, respectively. In Table 7.2, statistics about the route length and the network connectivity in our simulated network are given. $P(R_L \leq 4)$ is the probability that a route has at most four nodes including the source and destination nodes. The network connectivity is the ratio of the connected routes to the total number of possible routes assuming any two nodes are the source and destination pair. The statistics show that our simulated network is well connected and the route length is acceptable.

Table 7.1: The processing times and energy of the used cryptographic tools.

		Processing time (ms)	**Processing energy (mJ)**
1024-bit RSA	**Signing operation**	15.63	546.5
	Verifying operation	0.53	15.97
1024-bit DSA	**Signing operation**	7.94	313.6
	Verifying operation	9.09	338.02
Pairing operation		4.34 ms	25.5
MD5		8.56 µs/512 bytes	0.302
SHA-1		16.79 Megabytes/s (29 µs/512 bytes)	0.76 µ J/B

Table 7.2: Statistics of the simulated network.

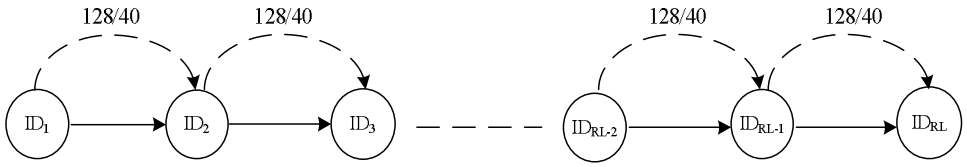
Average network connectivity	**$P(R_L \leq 4)$**	**$P(4< R_L \leq 6)$**	**$P(6 < R_L \leq 8)$**	**$P(8 < R_L \leq 10)$**	**$P(R_L > 10)$**
0.888	0.559	0.297	0.118	0.023	0.0041

7.1.2 Simulation Results

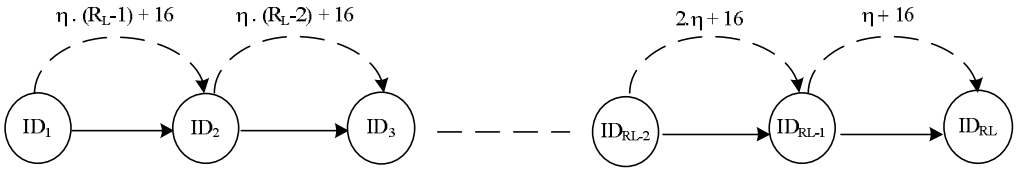
A. Average Packet Overhead

We define the *average security packet-overhead* as the average security related data relayed in all the hops of a route. In Figure 7.1(a), the security packet-overhead of the

signature-based incentive protocols is due to fixed-size and route-length-independent signature, e.g., 40 and 128 bytes for DSA and RSA based protocols, respectively. However, in Figure 7.1(b), the security packet-overhead in ESIP is due to the 16-byte hash chain's element (V_S^X) and the message hash series $HS(M_X)$ with η-byte truncated hash values at $X > 1$. Figure 7.1(b) also shows that the security packet-overhead is reduced by η bytes in each hop because each node drops its hash value. Unlike the signature-based incentive protocols, the security packet-overhead of ESIP depend on the route length (R_L).



a) The hop-by-hop security packet-overhead in RSA/DSA based incentive protocols.



b) The hop-by-hop security packet-overhead of ESIP.

Figure 7.1: The hop-by-hop security packet-overhead of ESIP and signature-based incentive protocols.

Figure 7.2 gives the relation between the average security packet-overhead and the route length for *ESIP*. *The figure shows that even at unrealistic and extreme cases, e.g., R_L = 20 nodes, the average security packet-overhead is less than 55 bytes at η = 4 bytes.* Figure 7.3 gives the equivalent route lengths of signature-based incentive protocol and *ESIP* for the same average security packet-overhead at η = 4 bytes. For example, the routes with six nodes in DSA and RSA based protocols are equivalent to routes with 8 and 15 nodes in *ESIP* for the same average security packet-overhead, respectively. The figure shows that *the average security packet-overhead of ESIP is less than that of the DSA and RSA based protocols when R_L < 13 nodes and R_L < 24 nodes at η = 4, respectively*. Moreover, the security packet-overhead of *ESIP* is less than that of the DSA based incentive protocol when R_L is fewer than

17 and 10 nodes for η of 3 and 5 bytes, respectively. The security packet-overhead of *ESIP* is also less than that of the RSA based incentive protocols when R_L is fewer than 75 and 45 nodes for η of 3 and 5 bytes, respectively. Although the DSA has less signature size than RSA, it results in much more end-to-end packet delay due to its longer verification time, as we will discuss in Section 7.1.2-B.

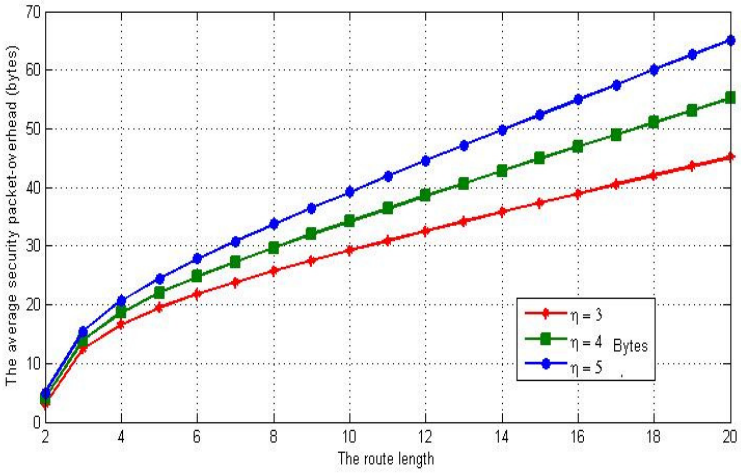


Figure 7.2: The average packet security-overhead in ESIP.

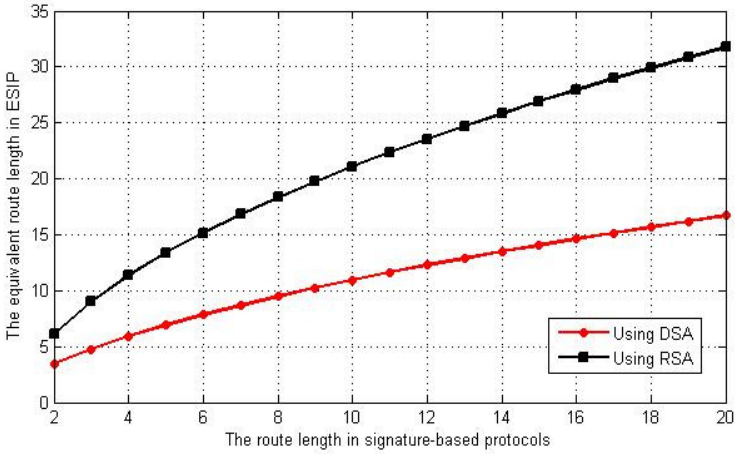
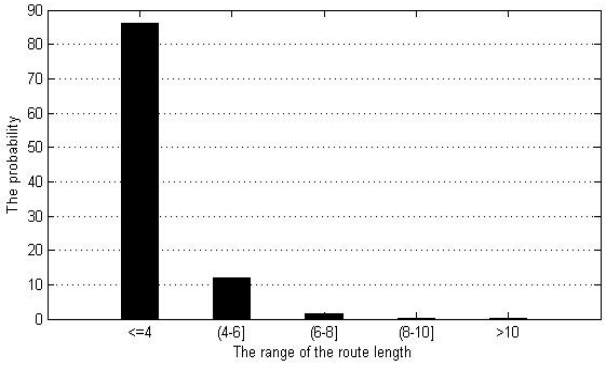


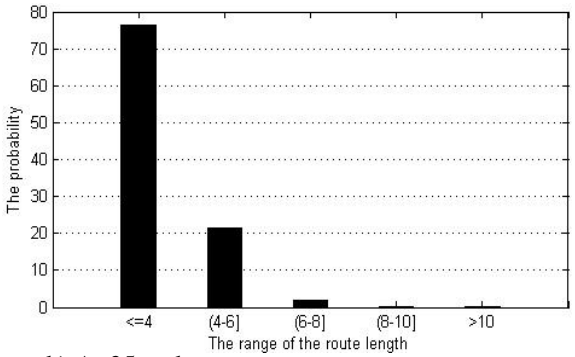
Figure 7.3: The equivalent route lengths for the same security packet-overhead.

Figure 7.4 shows the route length distribution at different number of nodes in the simulated network. At 15 nodes, the network is lightly connected because the average connectivity is 0.66. The network connectivity is measured by the number of connected routes to the total number of routes assuming any two nodes can be the source and

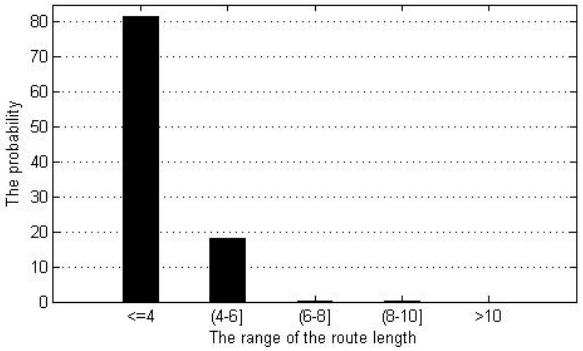
destination pair. As shown in Figure 7.4(a), 86% of the routes have four nodes or fewer, and only 0.0238% of the routes are longer than ten nodes. At 35 nodes, the average network connectivity is 0.99, the probability that a route is shorter than seven is 99.7%, and the probability that a route is longer than ten nodes is 0.0151%. At 50 nodes, the probability that a route is shorter than seven is 98.1%, and the probability that a route is longer than ten nodes is negligible. For dense network with 100 nodes, the probability that a route is shorter than seven is 99.99%, and the probability that a route is longer than ten nodes is negligible. Table 7.3 gives the probability that a route is longer than 13 nodes ($P(R_L > 13)$) at different network parameters. *The conclusion of these results is that the route length is less than 13 nodes with very high probability under realistic network parameters, and thus the expected security packet-overhead of ESIP is less than those of the DSA and RSA based incentive protocols.*



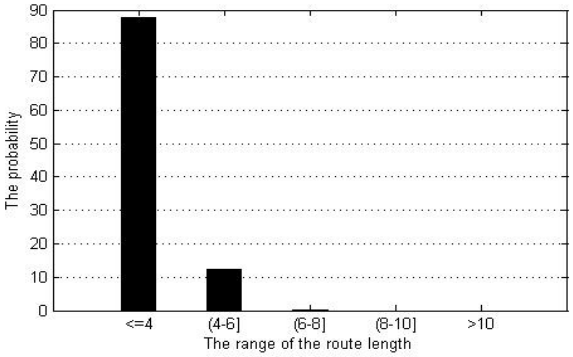
a) At 15 nodes.



b) At 35 nodes.



c) At 50 nodes.



d) At 100 nodes.

Figure 7.4: Route length distribution.

The average packet overhead is the average additional data (in bytes) attached to the

message including the routing and security data. Table 7.4 gives the average packet overhead in *ESIP* and signature-based incentive protocols. The packet overhead using RSA is much longer than DSA due to its longer signature. For the first packet, the average packet overhead of *ESIP* is more than that of the signature-based incentive protocol due to attaching the source node's signature, V_S^N and $HS(M_1)$. However, the packet overhead is less in the next packets because the source node does not attach signatures. The packet overhead of *ESIP* is 1.34 and 1.13 times the overhead of DSA and RSA based incentive protocols, and for a series of two packets, the ratios become 0.98 and 0.68, *so from the second packet, we gain the revenue of the investment in the first packet. Moreover, for a series of 10 packets, the data packet overhead in ESIP is 69% and 33% of that in the DSA and RSA based incentive protocols, respectively*.

Table 7.3: The average connectivity, route length at different network parameters.

Network dimension	Nodes' number	Average connectivity	Average route length	Pr ($R_L > 13$)
800 X 800	15	0.66	3.25	0
	30	0.97	3.66	0
	60	1	3.41	0
1600 X 1600	40	0.2235	3.6892	0.000444
	60	0.5394	5.5683	0.011
	100	0.9531	6.3174	0.0059
2000 X 2000	100	0.5591	7.4081	0.091
	150	0.948	7.7624	0.0539
	200	0.992	7.172	0.01225

Table 7.4: The average data packet overhead (bytes).

		RSA	DSA
Signature-based protocols		143	55
ESIP	**First packet (X = 1)**	161.73	73.73
	Subsequent packets (X > 1)	33.73	

B. Average End-to-End Packet Delay

The required cryptographic operations for *ESIP* and signature-based incentive protocols

are given in Table 7.5, where P, V, S, and H refer to pairing, verifying, signing, and hashing operations, respectively. It can be seen that *ESIP* requires more cryptographic operations in the first data packet, but *ESIP* requires only hashing operations from the second data packet.

Table 7.5: The required cryptographic operations in ESIP and signature-based protocols.

	ESIP			**Signature-based incentive protocols**		
	Source node	**Intermediate node**	**Destination node**	**Source node**	**Intermediate node**	**Destination node**
Data packet (X = 1)	$S + H \cdot R_L$	$2 \cdot V \cdot (R_L\text{-}2) + 2 \cdot H \cdot (R_L\text{-}2)$	$2 \cdot V + 2 \cdot H$	S	$2 \cdot V \cdot (R_L - 2)$	$2 \cdot V$
Data packet (X > 1)	$H \cdot R_L$	$2 \cdot H \cdot (R_L\text{-}2)$	$2 \cdot H$	S	$V \cdot (R_L\text{-}2)$	V

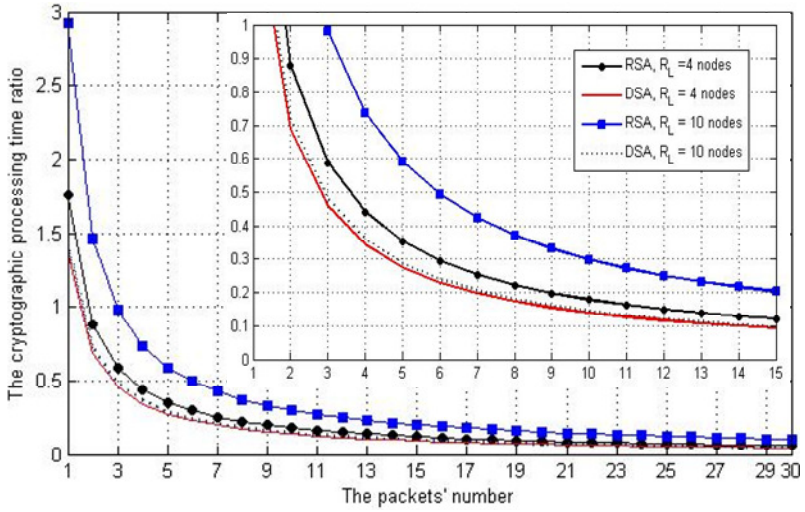


Figure 7.5: The ratio of *ESIP*'s cryptographic delay to that of signature-based incentive protocols.

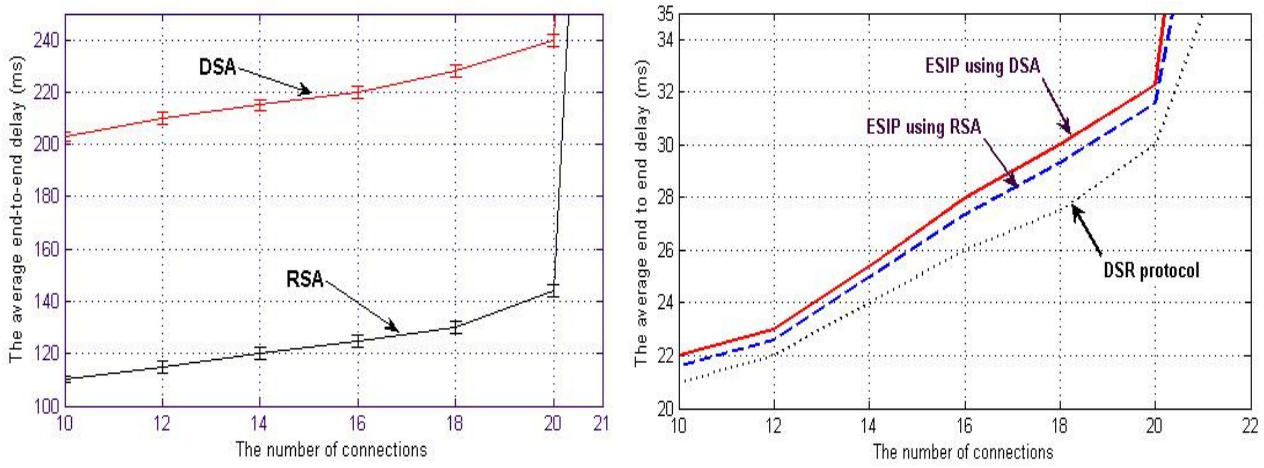
Table 7.6: The average packet series size, and cryptographic time and energy ratios.

Speed (m/s)	**Transmission rate / Nodes' number**	**0.5 packet/sec**					**1 packet/sec**				
		Av. packet series size	**Cryptographic energy ratio**		**Cryptographic delay ratio**		**Av. packet series size**	**Cryptographic energy ratio**		**Cryptographic delay ratio**	
			DSA	**RSA**	**DSA**	**RSA**		**DSA**	**RSA**	**DSA**	**RSA**
[0, 2]	**15**	126.8	0.029	0.033	0.037	0.051	289.7	0.01	0.011	0.011	0.016
	35	134.15	0.015	0.018	0.019	0.027	258	0.012	0.015	0.015	0.022
[0, 10]	**15**	42.55	0.09	0.1	0.117	0.16	94.6	0.05	0.055	0.063	0.084
	35	40.425	0.098	0.11	0.13	0.173	95.4	0.05	0.056	0.064	0.088

The computation-time utility function is the required processing time for *ESIP* to those of the signature-based incentive protocols. For N messages, $(2 \cdot N)$ signatures are generated from the source and destination nodes in signature-based incentive protocols, but only two signatures are generated in *ESIP*. The time utility function versus the number of messages is shown in Figure 7.5. For the first packet and R_L of 4, it can be seen that the time utility functions are 1.4 and 1.75 using DSA and RSA, respectively, and the ratios become 0.68 and 0.88 for two packets. Moreover, *for 13 packets, ESIP* requires *only 10% and 12% of the cryptographic delay in DSA and RSA based protocols, respectively*. That is because the computation time of the hashing operations are negligible comparing to those of the signing and verifying operations, i.e., the time utility functions are the reciprocal of the packets' number. In addition, the simulation results given in Table 7.6 demonstrates that under different network parameters, *the average size of the packet series is greater than 13, and the cryptographic delay in ESIP is incomparable with those of DSA and RSA based incentive protocols.*

The average end-to-end packet delay refers to the average time that packets traverse the network from the source node to the destination node. The end-to-end packet delay is due to propagation, cryptographic, queuing, etc, delays. Figure 7.6 shows the average end-to-end packet delay in *ESIP* and the signature-based incentive protocols at different traffic load expressed in number of connections, and Table 7.7 gives the confidence intervals of Figure 7.6(b). *The simulation results demonstrate that ESIP can significantly reduce the average end-to-end packet delay comparing to the DSA and RSA based protocols because the hashing operations that are computationally free (43 μs per operation) dominate the nodes' operations.* Up to 20 connections, the cryptographic delay dominates the channel contention and queuing delays, but over 20 connections, the delay significantly increases with and without the incentive protocol because the channel contention and queuing delays dominate. Although the DSA has shorter signature than the RSA, it results in longer delay in the signature-based incentive protocols due to its longer verification time, but the DSA increases

the delay very little in *ESIP* because the effect of the long delay of the first packet vanishes with the dominant hashing operations. Hence, *ESIP can be implemented more efficiently using DSA because it has shorter signature and the hashing operations can mitigate the long delay of the first packet.*



a) Signature-based incentive protocols. b) ESIP and original DSR.

Figure 7.6: The average end-to-end packet delay.

Table 7.7: 95% confidence interval (C.I.) for mean.

The number of connections	C. I. for mean	End-to-end delay			Packet delivery ratio		
		ESIP using DSA	ESIP using RSA	DSR	ESIP using DSA	ESIP using RSA	DSR
12	**Upper limit**	23.1	22.8	22.05	99.9	99.99	99.997
	Mean	23.03	22.6	22	99.93	99.95	99.994
	Lower limit	22.96	22.4	21.95	99.9	99.93	99.991
14	**Upper limit**	25.95	25.5	24.21	99.98	99.967	99.989
	Mean	25.4	25	24	99.94	99.96	99.989
	Lower limit	24.85	24.5	23.79	99.9	99.959	99.989
16	**Upper limit**	28.61	27.68	26.2	99.92	99.96	99.988
	Mean	28.01	27.36	26	99.9	99.95	99.985
	Lower limit	27.41	27.04	25.8	99.88	99.94	99.982
18	**Upper limit**	30.21	29.5	28	99.64	99.92	99.994
	Mean	30.02	29.3	27.5	99.6	99.88	99.989
	Lower limit	29.83	28.1	27	99.56	99.84	99.985
20	**Upper limit**	32.39	31.82	29.86	98.56	98.93	99.8
	Mean	32.32	31.6	30	98.5	98.9	99.6
	Lower limit	32.25	31.38	30.14	98.44	98.87	99.4

At high node mobility, Table 7.6 indicates that the average cryptographic delay increases, and Figure 7.7 shows that the end-to-end packet delay increases. That is because the size of the packet series decreases at high node mobility, and thus the effect of the heavy-weight first packet increases. However, the simulation results demonstrate that the overhead of *ESIP* is still incomparable with those of the DSA and RSA based incentive protocols because only the free computation hashing operations are used after the first packet.

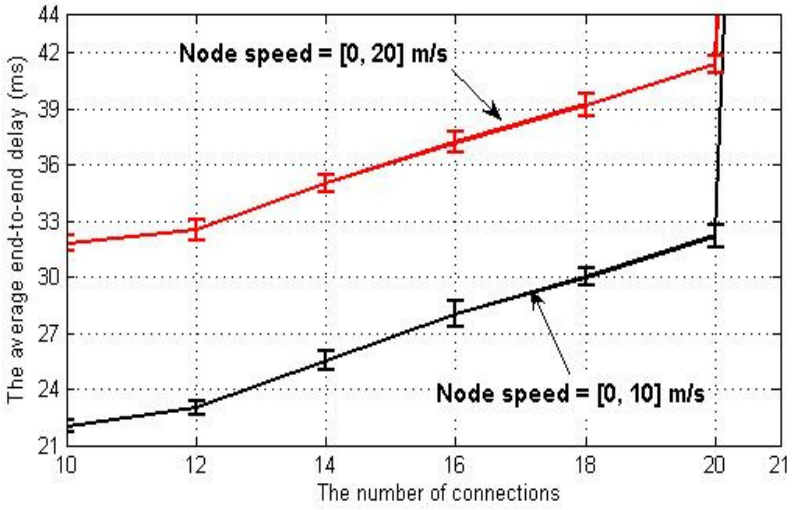


Figure 7.7: The impact of mobility on the end-to-end packet delay.

C. Packet Delivery Ratio

The packet delivery ratio (PDR) refers to the average ratio of messages that successfully delivered to the destination nodes with respect to those generated by the source nodes. Figure 7.8 gives the PDR for *ESIP* and the original DSR at different connections' number, and Table 7.7 gives the confidence intervals. Up to 20 connections, the PDR is quite high (above 98%). Above 20 connections, the PDR starts to decrease because more packets are dropped due to increasing the number of congested nodes and packet collision. Since each node has only 50-packet queue size and increasing the connections' number increases the packet arrival rate, the node is congested and drops the packets once the buffer is full. Moreover, increasing the cryptographic delay causes more congested nodes due to increasing the packet processing (or service) time. Comparing to the original DSR protocol, *ESIP* has a very little effect on the

PDR because the dominant hashing operations require very little computational time.

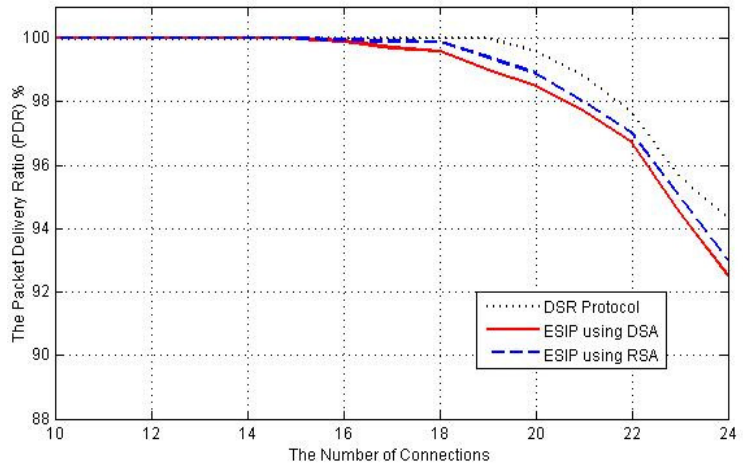


Figure 7.8: The packet delivery ratio.

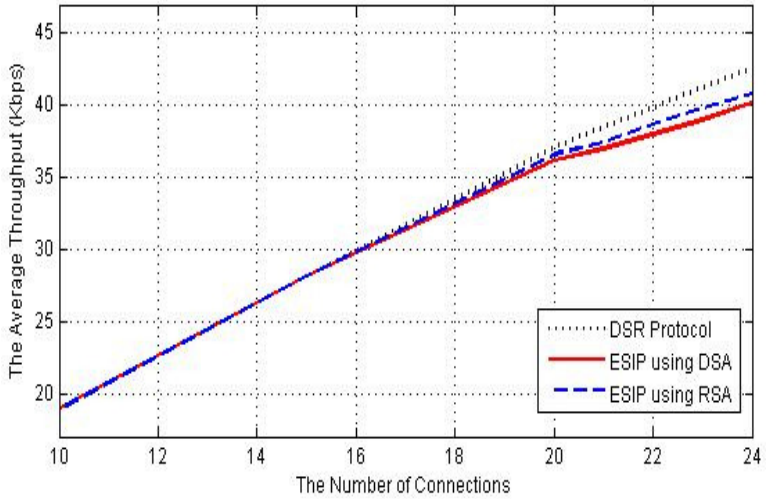


Figure 7.9: The average throughput.

D. Average Network Throughput

The average network throughput is computed by dividing the size of the received data by all the nodes over the simulation time. Since the end-to-end packet delay and the PDR in *ESIP* are close to those of the DSR, it is expected that the throughput of *ESIP* is close to that of the DSR. The simulation results shown in Figure 7.9 demonstrate that *ESIP* has very little effect on the throughput comparing to the original DSR protocol. Up to 20 connections, the throughput increases with increasing the number of connections, but beyond 20 connections,

the increasing rate starts to decrease because the network reaches its capacity, i.e., above 20 connections, the PDR decreases and the end-to-end packet delay increases as discussed in Sections 7.1.2-B and 7.1.2-C, respectively.

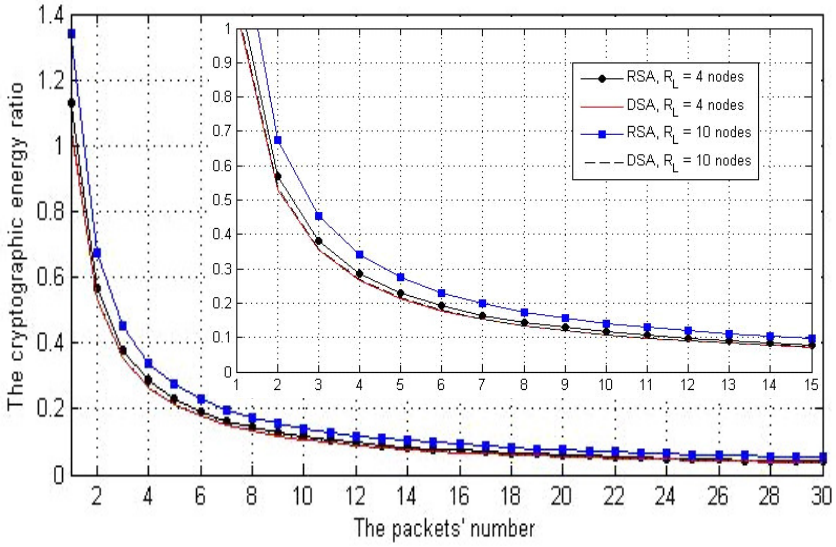


Figure 7.10: The ratio of ESIP's cryptographic energy to that of the signature-based incentive protocol.

E. Energy Consumption

Energy is consumed in relaying the packets and executing the cryptographic operations. As discussed in Sections 7.1.2-A, *ESIP* can reduce the packet overhead with a very high probability. *From Table 7.1, it can be seen that the consumed energy for the hashing operations is incomparable with those of the signing and verifying operations, which supports our approach of replacing the signature with the hashing operations*. Figure 7.10 gives the relation between the ratio of the required cryptographic energy in *ESIP* to that of the DSA and RSA based incentive protocols, and the number of data packets. At $R_L = 4$, *ESIP* requires 1.025 and 1.175 of the consumed cryptographic energy for the DSA and RSA based incentive protocols, respectively for the first packet. However, from the second packet *ESIP* requires less cryptographic energy, e.g., for *10 packets, ESIP requires around 10% of the cryptographic energy consumed in the DSA and RSA based incentive protocols at* $R_L = 4$. In addition, the simulation results given in Table 7.6 demonstrate that the average

cryptographic energy consumed in *ESIP* is incomparable with those consumed in the DSA and RSA based incentive protocols.

7.2 Replacing Receipts with Payment Reports

We simulate a MWN by randomly deploying 50 nodes with 150 m transmission range in a square cell of 1200 m × 1200 m. The constant bit-rate-traffic source is implemented in each node as an application layer, and the source and destination pairs are randomly chosen. To emulate the node mobility, we adopt the random waypoint model with a node speed uniformly distributed in the range [0, 5] m/s and pause time of 20 s. The data transmission rate is 0.5 packet per second. Ts, ID_i, X are five, four, and two bytes, respectively. The simulation results are averaged over 400 runs. We simulate the DSR routing protocol over an ideal and contention-free channel, i.e., all the nodes within transmission range receive packet transmission correctly. MATLAB is used instead of NS2 because the intention is to compare the overhead of the payment reports to the receipts. The effect of the un-simulated models such as non-ideal channel, channel contention, etc, should be the same on the payment-report-based and receipt-based protocols. For DSC [42], each node submits 69% of the receipts to guarantee submitting at least 95% of the receipts when there is no collusion. For PIS, the receipt submission probability is 0.2 to limit a colluder's lifetime to 7 sessions. The hash chain size in DSC and *ESIP* is 100.

7.2.1 Storage Overhead

Using 1024-bit RSA signature scheme and SHA-1 hash function, the average size of receipt, evidence, and report are given in Table 7.8. The receipt size in Sprite is large due to attaching a signature from each end node. The receipt size of DSC and PIS and the *Evidence* size in *ESIP* are much smaller due to hashing the signatures. The receipt size of DSC is larger than that of PIS due to replacing the destination node's signature with hashing operations and thus attaching the root and the last released hash value of the hash chain. *ESIP*'s *Evidence*

size is larger than the receipts in DSC and PIS due to replacing both the source and destination nodes' signatures with hashing operations and thus attaching four hash values. For *ESIP*, DSC, and PIS, 1MB storage area can store up to 7289.9, 10098, and 16425 receipts and evidences, respectively when one hash chain is used in DSC and ESIP. Hashing the nodes' signatures can alleviate the effect of the long RSA signature tag.

For *ESIP*, an *Evidence* size depends on the number of used hash chains (i) because two hash values are attached per hash chain. If the hash chain size is long enough, *ESIP* can generate one fixed-size *Evidence* per session. Table 7.9 gives the statistical distribution of the number of used hash chains. The simulation results demonstrate that more hash chains are used in low node mobility because more packets are transmitted before the route is broken. It can also be seen that the probability of using only one hash chain increases with the increase of Z. Properly choosing Z can reduce the number of used hash chains, which reduces the *Evidence* size and saves the destination node's resources because the unused hash values in a chain should not be used for other sessions to secure the payment. A good Z depends on the average number of transmitted packets before the route is broken, which is related to the packet transmission rate, the node speed, and the expected number of transmitted packets in the session.

Table 7.8: The average receipt, evidence, and payment report size (bytes).

Receipt-based incentive protocols					**Payment-report-based incentive protocols**	
Sprite	**ESIP**	**DSC**	**PIS**	**Express**	**report**	**Evidence**
296.84	43.84 + 80 · i	63.84 + 40 · i	63.84	196	23.84	43.84 + 80 · i

Table 7.9: The statistical distribution for the number of used hash chains.

Smax	**Hash chain size (Z+1)**	**P(i = 1)**	**P(i = 2)**	**P(i = 3)**	**P(i > 3)**
3 m/s	**30**	0.48	0.24	0.11	0.17
	50	0.6	0.28	0.12	0
10 m/s	**30**	0.89	0.11	0	0
	50	0.99	0.01	0	0

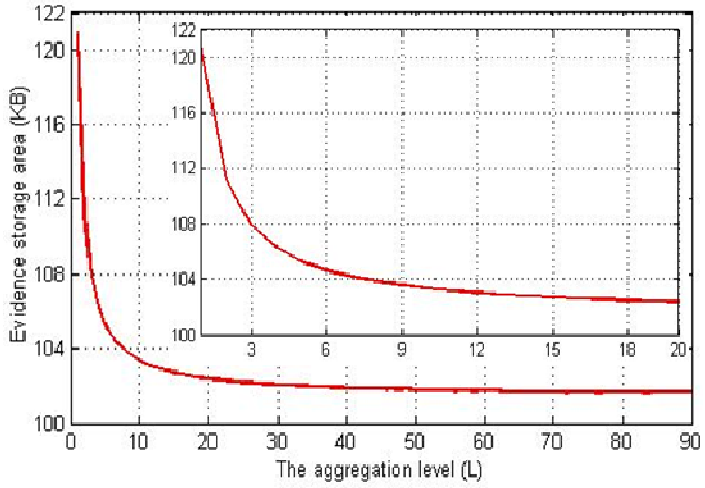


Figure 7.11: The effectiveness of the *Evidence* aggregation technique.

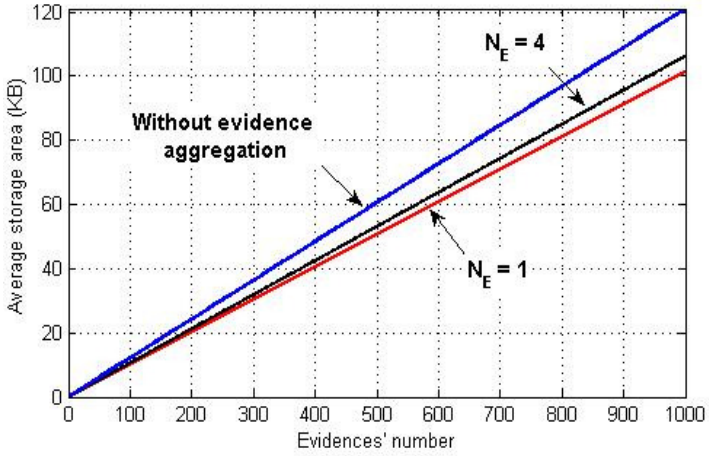


Figure 7.12: The average storage area at different aggregation levels.

Figure 7.11 shows the required *Evidence*s' storage area for 1000 *Evidence*s and at different aggregation level (L) that means that the evidences are stored in 1000/L aggregated *Evidences*. The figure shows that the increase of L over 10 has little effect on the storage area but increases the number of redundant *Evidence*s that are submitted. For example, if L is two, 500 compact *Evidence*s are composed and each one contains two individual evidences, so one redundant evidence is submitted if an *Evidence* is requested; and if L is 1000, all the individual *Evidences* are stored in one compact *Evidence* and thus 999 redundant *Evidence*s are submitted if an *Evidence* is requested. In MWN, the nodes are equipped with limited

energy supplies and the network is characterized by limited bandwidth. The nodes delete the receipts after submission in the receipt-based incentive protocols, but they have to stores the *Evidence*s for some time in *ESIP*. However, the storage area is not the main concern and the more important factors are the bandwidth and energy. Moreover, the capacities of the flash memories continue to rise as per Moore's Law and their costs continue to plummet [77]. Figure 7.12 gives the relation between the number of *Evidence*s and their average storage area at different number of individual *Evidence*s which is aggregated in one compact evidence (or N_E). Without *Evidence* aggregation, the *Evidence*s occupy large storage area, and the storage area is minimized when all the *Evidence*s are aggregated in one compact *Evidence*.

7.2.2 Payment-Report Submission overhead

From Table 7.9, *ESIP* requires submitting only payment report of 23.84 bytes per route. *Evidence*s are submitted only in case of cheating in *ESIP*, but receipts are always submitted in the receipt-based incentive protocols. Table 7.10 gives the amount of payment data submitted to Tp for ten-minute data transmission at different node speed. During the transmission, a new route 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 data packet, and the increase of the packet transmission rate increases the number of receipts significantly due to increasing the number of transmitted packets.

The simulation results indicate that Sprite requires submitting a large number of payment receipts due to generating a receipt per message. Moreover, the increase of the nodes' speed increases the number of receipts because the receipts are generated for the un-delivered packets. For DSC and *ESIP*, a new receipt or payment report is generated when a route is broken or Z messages are transmitted, but a new receipt is generated only when a route is broken in PIS. Table 7.10 indicates that more payment data are submitted at high node mobility because the routes are more frequently broken, i.e., the source node's

messages are transmitted over a larger number of routes. *ESIP* requires submitting more payment data than PIS because only one node has to submit the receipt in PIS and the other intermediate nodes submit the receipt probabilistically, but all the nodes in a route have to submit payment reports in *ESIP*. However, PIS requires two signatures per packet, which consumes the nodes' resources and increases the end-to-end delay.

Table 7.10: The amount of submitted payment reports and receipts (KB).

Protocol / Node speed	Sprite	DSC	PIS	ESIP	
				Payment report	Evidence size
[0, 5] m/s	192.19	2.11	0.56	0.74	4.44
[0, 20] m/s	192.19	5.51	1.22	1.92	11.59

7.2.3 Payment Processing Overhead

Table 7.11 gives the payment processing overhead for a session held for ten minutes, where S, H, and V refer to signing, hashing, and verifying operations, respectively. The table indicates that *ESIP* does not need any cryptographic operations for clearing the payment in case of no cheating, and the *Evidence*s are occasionally processed. The simulation results demonstrate that payment processing overhead in *ESIP* is incomparable with those of the receipt-based incentive protocols.

Table 7.11: The payment processing operations for ten-minute session.

	[0, 5] m/s			[0, 20] m/s		
	S	**H**	**V**	**S**	**H**	**V**
Sprite	0	0	600	0	0	600
DSC	15	307.5	0	39.2	319.6	0
ESIP (fair reports)	**0**	**0**	**0**	**0**	**0**	**0**
ESIP (cheating reports)	2	X+1	0	2	X+1	0
PIS	12.6	306.3	0	38.8	319.4	0

7.3 Evaluation of Trust-Based and Energy-Aware Routing

Packet drop degrades the network performance significantly. The average throughput degrades by 16% to 32% if 10% to 40% of the nodes drop the network packets, and the end-to-end packet delay increases linearly with increasing the attackers' number [14], [15]. Moreover, since a new payment report and *Evidence* are generated when the session is broken and re-established, packet drop increases the payment overhead and exhausts the nodes' resources in re-establishing the broken routes. Route breakage also wastes the end nodes' credits because they pay for the un-delivered messages. In MWNs, a packet cannot reach to its destination if any intermediate node drops the packet, and thus the packet delivery ratio decreases with the increase of the number of packet droppers. Equation 7.1 gives the probability of breaking a session with R_L nodes (or n = R_L-2 intermediate nodes) due to malicious action. P_m is the malicious nodes' ratio, which is equivalent to the probability that an intermediate node is malicious. The packet delivery ratio for a route with R_L nodes ($PDR(R_L)$) is the number of data packets received by the destination node to the total number of packets sent in a route with R_L nodes. In Equation 7.2, $PDR(R_L)$ and $PDR_0(R_L)$ are the average packet delivery ratios with and without the existence of packet droppers, respectively for a route with R_L nodes. *Figure 7.13 shows that a low ratio of the packet droppers such as 20% can reduce the packet delivery ratio by 74% and 60% for sessions with eight and six nodes, respectively.* Moreover, the increase of R_L or P_m increases the session breakage probability and thus degrades the packet delivery ratio.

$$P_b(R_L) = 1 - (1 - P_m)^{R_L - 2} \tag{7.1}$$

$$PDR(R_L) = PDR_0(R_L) \cdot (1 - P_b(R_L)) \tag{7.2}$$

A route may be broken because of node failure, link failure, or node mobility. Establishing routes through stable links has been extensively studied and the proposed

solutions can be integrated to our trust-based and energy aware routing protocol. Our routing protocols aim to reduce route breakage due to the node failure and mobility because a node's trust values depict its failure probability and mobility level. A node's trust value is low when it lies about its residual energy, fails frequently, has high mobility, or drops the messages intentionally. A trust value is a live and real measurement to the node's mobility level and failure probability.

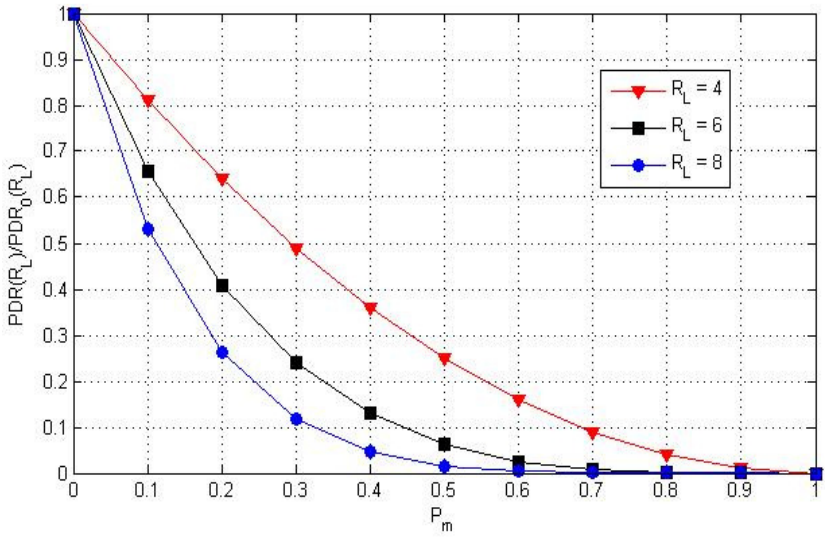


Figure 7.13: The expected drop of the PDR due to the packet droppers.

In route establishment, the nodes in a route use signatures to authenticate themselves to thwart many attacks that can be launched by the external attackers [69] and also to secure ESIP by holding the nodes accountable for their actions. For example, the external attackers that are not members in the network may launch *Resource-Exhaustion* attacks by frequently flooding the network with *RREQ* packets to exhaust the nodes' resources. In this authentication process, each node performs one signing operation and multiple verifying operations, and thus RSA signature scheme may be a proper choice because the verifying operations require much less computational time and energy than those of the signing operations, as indicated in Table 7.1. Moreover, dropping the data packets is more serious

than dropping the *RREP* packets because they are much longer, so the number of the public-key-cryptographic operations and the payment overhead can be significantly reduced if *ESIP* aims to identify only the data packet droppers. In this case, the ISR(0) reports are not submitted, and the nodes can authenticate themselves in the *RREP* packets in order to reduce the number of the public-key-cryptographic operations because the *RREQ* packets are processed by a larger number of nodes.

The SRT and WRT ratings are stored in one and seven bits, respectively, and thus a rating window for 320 ratings requires storage area of 40 and 280 bytes, respectively. Moreover, the storage area can be significantly reduced by making the size of the rating window dynamic. The rating windows can be short for the good-reputation nodes and long for the bad-reputation nodes, e.g., suspicious nodes, to better judge their behaviour.

In our simulation, 70 nodes with 125 m transmission range are randomly deployed in 1000 m by 1000 m. n_L is the number of nodes having low trust values ($T_{1,A}(t)$) that are uniformly distributed in [0.6, 0.995), but 70-n_L is the number of nodes having high trust values ($T_{1,A}(t)$) of 0.995. All the nodes start the simulation with initial energy that is sufficient for relaying 100 messages. The given results are averaged over 30 simulation runs. In each run, 300 communication sessions with randomly chosen source and destination pairs are established. The route is re-established if it is broken before sending 15 messages. The TTL is 10, and the source node's energy and trust requirements in the SRR protocol are 5 and 0.88, respectively. We do not simulate node mobility because it is already included in the trust values, i.e., if $T_{1,A}(t)$ is 0.6, that means that node A drops the messages with the probability of 0.4. As we intend to study the effect of the node selection on the network performance but not the communication interface, our simulation is written in Matlab instead of NS2.

The PDR is a good measurement for route stability. From Figure 7.14, our route-establishment protocols outperform the dynamic source routing (DSR) protocol because unlike the DSR that randomly chooses the nodes, our protocols make informed routing

decisions. Thus, our protocols can establish more stable routes compared to the DSR. We can see that the increase of n_L raises the chance of involving the low-trust nodes in routes in the DSR, but our protocols can avoid these nodes and select the high-trust nodes. Although the BAR outperforms the SRR in the PDR, the BAR has longer route establishment delay because the destination node has to wait to receive different routes.

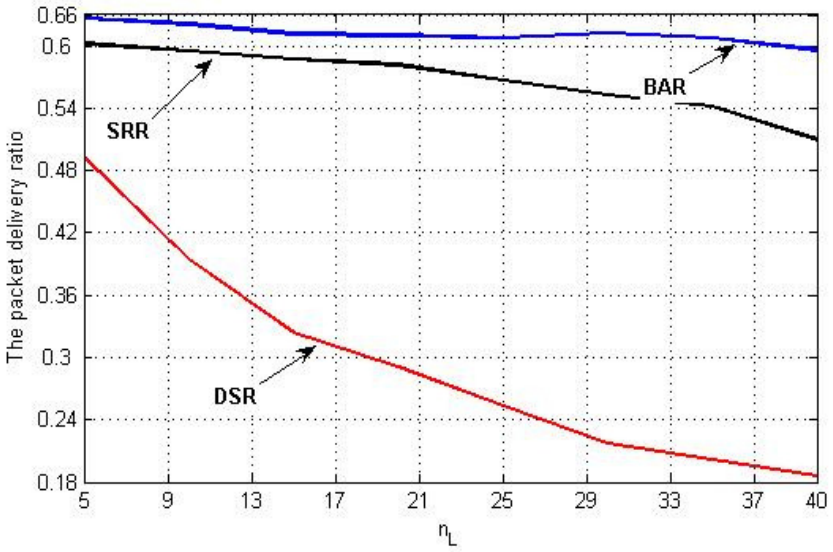


Figure 7.14: The packet delivery ratio VS n_L.

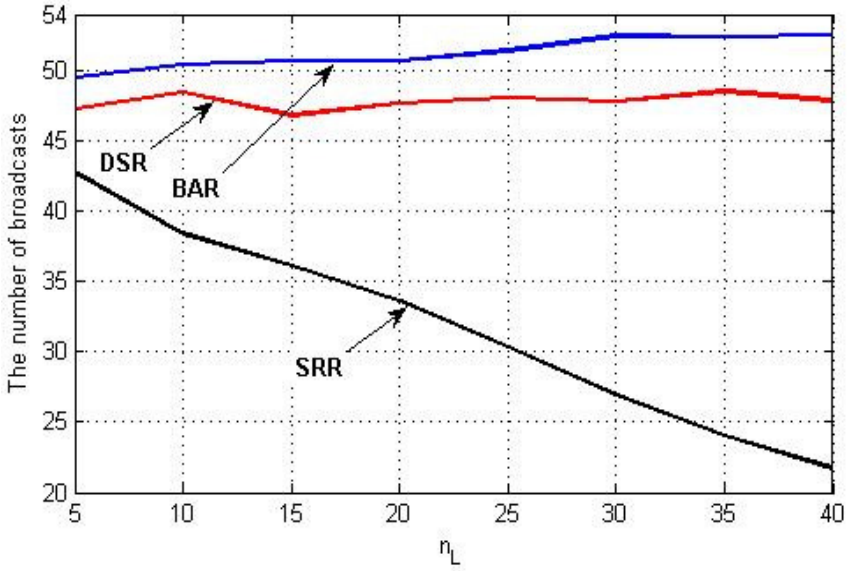


Figure 7.15: The number of broadcasts VS n_L.

Figure 7.15 shows that the number of broadcasts in the DSR and the BAR does not depend on n_L, but the increase of n_L decreases the broadcasts' number in the SRR because more nodes cannot satisfy the source nodes' trust requirement, and thus they do not broadcast the *RREQ* packets. The BAR requires more broadcasts than the DSR because each node broadcasts the *RREQ* only once in the DSR, but some nodes may broadcast a *RREQ* packet more than once in the BAR.

In Figure 7.16, the network connectivity is the number of connected routes to the total number of route establishment trials. We can see that the network connectivity in the DSR and BAR does not depend on n_L, but the increase of n_L decreases the network connectivity in SRR because more nodes cannot satisfy the source node's requirements and thus more routes cannot be established. SRR protocol may not establish a route if the source node's requirements are not adequately determined. In order to increase the probability of establishing a route successfully, the source node can periodically tune its requirements by learning from its past trials.

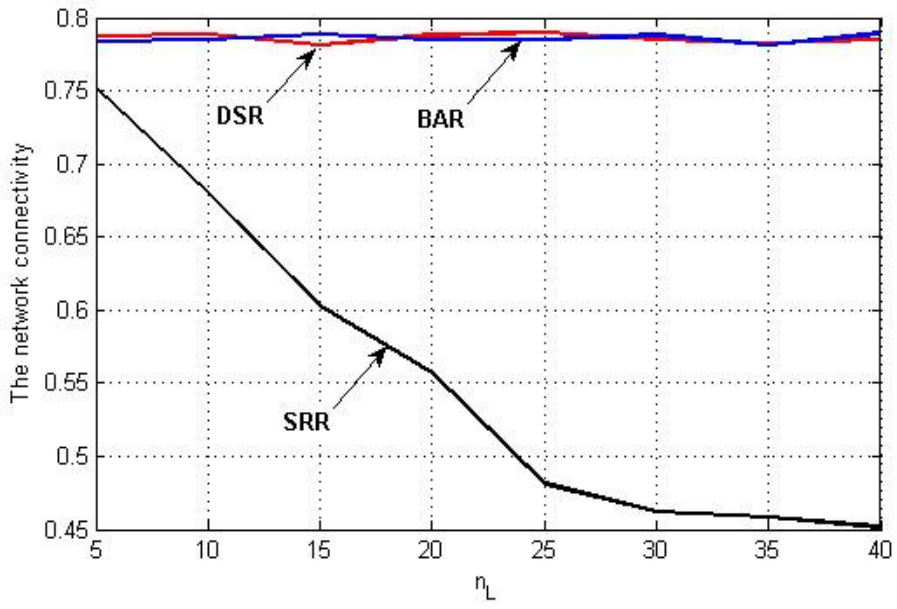


Figure 7.16: The network connectivity VS n_L.

7.4 PRIPO Evaluation

7.4.1 Cryptographic Overhead

To evaluate the computational times of the cryptographic operations used in PRIPO, we have implemented AES symmetric key cryptosystem and SHA-1 (160 bit) hash function using Crypto++ library [74]. The secure key size is at least 128 bits according to NIST [73]. 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 [75] that a hashing operation requires 0.76 μJ/byte and encryption and decryption operations require 1.21 μJ/byte.

7.4.2 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 7.12 indicate that the expected delay is acceptable due to using 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 7.12: Simulation results.

	URREQ	**RREP**	**DREST**	**Data Packet**
Avg. packet length (bytes)	73.68	95.31	170.27	534
Avg. delay (ms)	19.36	21.35	21.24	32.76

Chapter 8

Conclusions and Future Work

In this chapter, we conclude this thesis and discuss our future work.

8.1 Conclusions

In this thesis, we first have proposed a fair, efficient, and secure cooperation incentive protocol for MWNs. The payment model has been developed to implement micropayment efficiently in MWNs to stimulate the nodes' cooperation. Our protocol adopts a fair charging policy by charging both the source and destination nodes when both of them benefit from the communication. For efficient implementation, the proposed incentive protocol limitedly uses the heavy-weight public key cryptography. Instead of generating two signatures from the source and destination nodes to secure the payment, the public-key cryptographic operations are required only for the first packet, and they are replaced with the efficient hashing operations in the next packets. Therefore, for a series of packets, the heavy-weight overhead of the first packet vanishes and the overall overhead converges to that of the lightweight hashing operations.

Moreover, the communication sessions may occur without involving a trusted party. The nodes in a session have to submit the payment data to Tp. Reducing the overhead of the payment data is essential for the practical implementation of an incentive protocol due to the high frequency of low-value transactions. In order to reduce the overhead of submitting and processing the payment data, our protocol is receipt-free. The nodes submit lightweight payment reports containing their alleged charges and rewards for different sessions, and store undeniable security *Evidence*s. A security mechanism has been proposed to identify the fair and the cheating reports. The fair reports can be processed with almost no processing overhead and with acceptable delay. For the cheating reports, the security mechanism requests the *Evidence*s to identify and evict the cheating nodes. In order to reduce the evidences' storage area, the nodes' signatures are hashed instead of storing the long-size signatures. In addition, an *Evidence* aggregation technique has been proposed to reduce the *Evidence* storage area by generating a smaller-size compact *Evidence* for multiple sessions instead of *Evidence* per session. On the other hand, sessions may be broken normally, e.g., due to mobility, or intentionally due to malicious actions. We have proposed a novel trust/reputation system to measure the nodes' packet drop rate by processing the payment reports. The reputation values are used to identify and evict the malicious nodes that intentionally drop the packets. SRT and WRT techniques have been proposed to calculate the nodes' reputation values. SRT offers equal negative ratings to the two nodes in a broken link, but WRT offers more negative rating to the low-reputation node that dropped more packets in the past.

We also have proposed a trust system to measure the nodes' packet-relay probability. The nodes' trust values are attached to their certificates to integrate the nodes' past behaviour in the routing decision-making. We have proposed trust-based and energy-aware route-establishment protocols called SRR and BAR to route the packets through the highly trusted nodes having sufficient energy to minimize the route breakage probability. Our protocols can make intelligent routing decisions based on the nodes' past behaviour and residual energy.

The protocols stimulate the nodes not only to cooperate but also to provide high packet-relay ratio.

Our security analysis and performance evaluations demonstrate that our incentive protocol can secure the payment with very low overhead because the hashing operations dominate the nodes' operations. For a series of two packets, our incentive protocol has lower cryptographic delay and energy than DSA and RSA based protocols, and for a series of 13 packets, the protocol requires around 10% of the cryptographic delay and energy of the DSA and the RSA based incentive protocols. Moreover, the packet overhead is less than that of the DSA and RSA based protocols with very high probability, e.g., for a series of 10 packets, the data packet overhead of our protocol is 70% and 37% of those of the DSA and the RSA based protocol, respectively. Our analysis and simulations also demonstrate that the payment can be cleared with almost no processing overhead and submitting lightweight payment reports while achieving the same security strength as the receipt-based incentive protocols. The simulation results demonstrate that the *Evidence*s can be stored in a compact storage area.

Moreover, WRT technique can precisely identify the irrational packet droppers with negligible false positive ratio because the honest and the malicious nodes receive less and more negative ratings, respectively. The reputation system is secure against small-scale irrational collusion attacks and robust against large-scale collusion attacks because the attackers lose credits and defame their reputations with launching these attacks. The simulation results demonstrate that our route establishment protocol can establish stable routes due to directing the traffic through the highly trusted nodes having sufficient energy, which can significantly improve the packet delivery ratio.

Finally, we have proposed a privacy-preserving routing and incentive protocol for hybrid ad hoc wireless network. Micropayment is used to stimulate the nodes' cooperation without submitting payment receipts or reports. 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 the node cooperation and the user privacy preservation can be securely and efficiently integrated in one protocol.

In summary, the thesis proposes a novel protocol for thwarting the *Packet-Drop* attack, enforcing fairness, discouraging the *Resource-Exhaustion* attack, and routing the packets through the highly trusted nodes having sufficient energy to minimize the session breakage probability and thus maximize the packet delivery ratio. The thesis also proposes an efficient protocol for stimulating the node cooperation and user privacy.

8.2 Future Work

Delay tolerant wireless networks (DTNs) [78-82] are an emerging class of networks characterized by long packet delivery delay and lack of fully connected path between the source and destination nodes. Consequently, packet transmission follows a store-carry-and-forward approach where the mobile nodes acting as packet relays buffer in-transit packets until the next node in the path appears, and so on, until the packets reach their destinations. Many useful applications have been developed for DTNs. Pocket-switched DTNs take advantage of the increasing popularity of the mobile devices equipped with wireless network interfaces to enable a new class of social networking applications. DTNs can be readily deployed at low cost in developing or remote areas. Vehicular DTNs can be used for disseminating safety and location-dependent information. For mobile sensor DTNs, sensors are attached to the nodes, e.g., vehicles, to monitor environment and state of roads, e.g., potholes and black ice. However, the practical implementation of DTNs is questionable because the networks' unique characteristics have made them vulnerable to serious security threats.

The selfish nodes drop other nodes' packets because packet relay consumes their resources without any benefits, and the irrational attackers such as compromised or

malfunctioned nodes launch *Denial-of-Service* attacks by dropping the packets. Other attackers may analyze the network traffics to detect the users' activities and locations. These attacks are severe threat to the network proper operation and users' privacy. *The presence of even a small number of attackers results in repeatedly dropped packets, which may cause network failure.* In the traditional MWNs, the cooperation incentive protocols have been used for thwarting selfishness attacks. The nodes usually submit payment receipts, proofs of packet relay, to a trusted party to charge the source and destination nodes and reward the intermediate nodes credits. However, *the protocols designed for MWNs cannot be used for DTNs because they require establishing end-to-end path before the data transmission occurs*. Moreover, unlike MWNs that transmit one copy of each packet, multiple copies are transmitted in DTNs to enhance the packet-delivery probability, which makes the efficient implementation of the incentive protocols a real challenge [83]. Involving many nodes in packet transmission significantly increases the protocol overhead in terms of the number of receipts and cryptographic operations performed by the nodes.

In addition, in MWNs, an intermediate node is supposed to transmit a packet once receiving it, and thus each node can monitor the transmissions of the successor node in the path to evaluate the node's trust value (or packet-relay probability). The trust values are used to identify the irrational packet droppers and direct the traffic through the highly trusted nodes to maximize the packet-delivery probability. However, *the unique "store-carry-and-forward" packet transmission approach makes using this monitoring technique not possible in DTNs*.

In our future research, we will first investigate the efficient and secure implementation of cooperation incentive protocol in DTNs. In this thesis, we have proposed novel approaches for efficiently implementing an incentive protocol in MWNs, which will help us to devise new approaches for DTNs with considering their unique characteristics. Second, we will study designing a trust system to evaluate the nodes' trust values. In this thesis, we have proposed a novel trust-evaluation approach for MWNs based on processing the incentive

protocol's payment reports, but applying this approach for DTNs is a challenge, e.g., due to the multi-copy packet transmission. Third, we will design a routing protocol to direct the packets through the highly trusted nodes to maximize the packet-delivery probability. Finally, we will investigate using privacy-preserving techniques such as mixers, onion routing and pseudonyms [84-88] to enhance the users' privacy. Integrating these techniques with our trust-based routing and incentive protocols with considering the DTNs' characteristics will be carefully studied.

Bibliography

[1] N. V. Marathe, U. B. Desai, and S. N. Merchant, "Base station selection strategy in multihop cellular networks: A new approach", Proc. of IEEE International Conference on Signal processing, Communications and Networking, pp. 401-404, January 4-6, 2008.

[2] Y. Lin and Y. Hsu, "Multihop cellular: A new architecture for wireless communications", Proc. of IEEE INFOCOM'00, vol. 3, pp. 1273–1282, March 26-30, 2000.

[3] R. Schoenen, R. Halfmann, and B. Walke, "An FDD multihop cellular network for 3GPP-LTE", Proc. of IEEE VTC Spring Conference, Singapore, May 2008.

[4] H. Wu, C. Qios, S. De, and O. Tonguz, "Integrated cellular and ad hoc relaying systems: iCAR", IEEE Journal of Selected Areas in Communications, vol. 19, no. 10, pp. 2105–2115, October 2001.

[5] 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.

[6] 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.

[7] Y. Jiang, M. Shi, X. Shen, and C. Lin, "BAT: A robust signature scheme for vehicular networks using binary authentication tree", IEEE Transactions on Wireless Communications, vol. 8, no. 4, pp. 1974 - 1983, 2009.

[8] X. J. Li, B. C. Seet, and P. H. J. Chong, "Multihop cellular networks: Technology and economics", Computer Networks, vol. 52, pp. 1825–1837, 2008.

[9] S. Bah, R. Glitho, and R. Dssouli, "SIP servlets for service provisioning in multihop cellular networks: High-level architectural alternatives", Proc. of IEEE CCNC, pp. 127-131, 2008.

[10] M. Kubisch, S. Mengesha, D. Hollos, H. Karl, and A. Wolisz, "Applying ad-hoc relaying to improve capacity, energy efficiency, and immission in infrastructure-based

WLANs". Proc. of Kommunikation in Verteilten Systemen, pp. 195-206, Leipzig, Germany, 2003.

[11] R. Schoenen, R. Halfmann, and B. H. Walke, "MAC performance of a 3GPP-LTE multihop cellular network", Proc. of IEEE ICC, pp. 4819- 4824, 2008.

[12] F. Hossain and H. Chowdhury, "Impact of mobile relays on throughput and delays in multihop cellular network", Proc. of the IEE Fourth International Conference on Wireless and Mobile Communications (ICWMC'08), pp. 304-308, 2008.

[13] P. Gupta and P. Kumar, "The capacity of wireless networks", IEEE Transactions on Information Theory, vol. 46, no. 2, pp. 388-404, March 2000.

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

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

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

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

[18] Y. Zhang and Y. Fang, "A fine-grained reputation system for reliable service selection in peer-to-peer networks", IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 8, pp. 1134-1145, August 2007.

[19] S. B. Lee, G. Pan, J-S Park, M. Gerla, and Songwu Lu, "Secure incentives for commercial ad dissemination in vehicular networks," Proc. of MobiHoc'07, September 2007.

[20] R. Lu, X. Lin, H. Zhu, C. Zhang, P.H. Ho and X. Shen, "A novel fair incentive protocol for mobile ad hoc networks," Proc. of IEEE WCNC'08, Las Vegas, Nevada, USA, March 31 - April 3, 2008.

[21] 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.

[22] 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.

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

[24] 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.

[25] Y. Zhang and Y. Fang, "A secure authentication and billing architecture for wireless mesh networks", ACM Wireless Networks, vol. 13, no. 5, pp. 569–582, October 2007.

[26] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, "Analysis of link failures in an IP backbone", Proc. of IMW 2002, ACMPress, Marseille, France, November 2002.

[27] 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.

[28] L. Buttyan and J. Hubaux, "Enforcing service availability in mobile ad-hoc WANs", Proc. of IEEE/ACM international symposium on Mobile Ad Hoc Networking and Computing (MobiHOC'00), pp. 87-96, Boston, Massachusetts, USA, August 11, 2000.

[29] 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.

[30] 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.

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

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

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

[42] M. E. Mahmoud and X. Shen, “DSC: Cooperation incentive mechanism for multi-hop cellular networks”, Proc. of IEEE ICC’09, Germany, June 14-18, 2009.

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

[44] S. Buchegger and J. Boudec, “Performance analysis of the CONFIDANT protocol: cooperation of nodes – fairness in distributed ad hoc networks”, Proc. of IEEE/ACM MOBIHOC’02, pp. 226-236, Switzerland, June 9-11, 2002.

[45] P. Michiardi and R. Molva, “CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks”, Proc. of IFIP CMS’02, pp. 107-121, Portoroz, Slovenia, September 26-27, 2002.

[46] 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.

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

[48] 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.

[49] J. Douceur, “The Sybil attack” Proc. of IPTPS, pp. 251-260, 2002.

[50] 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, 2003.

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

[52] D. Boneh and M. Franklin, “Identity based encryption from the weil pairing”, Proc. of Crypto’01, LNCS, Springer-Verlag, vol. 2139, pp. 213-229, 2001.

[53] 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.

[54] T. Rabin and M. Ben-Or, “Verifiable secret sharing and multiparty protocols with honest majority”, Proc. of ACM symposium on Theory of Computing, pp. 73–85, Seattle, Washington, United States, 1989.

[55] C. Cachin, K. Kursawe, A. Lysyanskaya, and R. Strobl, “Asynchronous verifiable secret sharing and proactive cryptosystems”, Proc. ACM Conference on Computer and Communications Security, CCS02, pp. 88-97, 2002.

[56] 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.

[57] 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.

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

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

[60] 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.

[61] 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.

[62] 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.

[63] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing", IEEE Pervasive Computing, vol. 2, no. 1, pp. 46-55, 2003.

[64] A. Pfitzmann and M. Köhntopp, "Anonymity, unobservability, and pseudonymity - A proposal for terminology", Proc. of Workshop on Design Issues in Anonymity and Unobservability, pp.1-9, 2000.

[65] S. Seys and B. Preneel, "ARM: anonymous routing protocol for mobile ad hoc networks", Proc. of the 20th IEEE International Conference on Advanced Information Networking and Applications (AINA'06), pp. 133-137, 2006.

[66] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks", Mobile Computing, Chapter 5, Kluwer Academic Publishers, pp. 153-181, 1996.

[67] A. Menzies, P. Oorschot, and S. Vanstone, "Handbook of applied cryptography", CRC Press, http://www.cacr.math.uwaterloo.ca/hac, Boca Raton, Fla., 1996.

[68] NIST, "Digital hash standard", Federal Information Processing Standards Publication 180-1, April 1995.

[69] 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.

[70] 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.

[71] 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.

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

[73] National Institute of Standards and Technology (NIST), "Recommendation for key management - part 1: general (revised)", Special Publication 800-57 200, 2007.

[74] W. Dai, "Crypto++ Library 5.6.0", http://www.cryptopp.com, 2009.

[75] 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

[76] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks", IEEE Journal Selected Areas Communication, vol. 24, no. 2, pp.247-260, 2006.

[77] A. Mitra el at, "High-performance low power sensor platforms featuring gigabyte scale storage", Proc. of IEEE/ACM Measurement, Modeling, and Performance Analysis of Wireless Sensor Networks, 2005.

[78] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges", IEEE Communications Surveys & Tutorials, vol. 8, no. 1, pp. 24-37, 2006.

[79] B. J. Choi and X. Shen, "Adaptive asynchronous sleep scheduling protocols for delay tolerant networks", IEEE Transactions on Mobile Computing, to appear.

[80] R. Lu, X. Lin, H. Zhu, and X. Shen, and B. R. Preiss, "Pi: A practical incentive protocol for delay tolerant networks", IEEE Transactions on Wireless Communications, vol. 9, no. 4, pp. 1483-1493, 2010.

[81] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, “SMART: A secure multi-layer credit based incentive scheme for delay-tolerant networks”, IEEE Transactions on Vehicular Technology, vol. 58, issue 8, pp. 4628 - 4639, 2009.

[82] P. Hui, J. Crowcroft, and E. Yoneki, “BUBBLE Rap: Social-based forwarding in delay tolerant networks”, Proc. of MobiHoc’08, Hongkong, China, May 2008.

[83] Y. Li, Y. Jiang, D. Jin, L. Su, L. Zeng, and D. O. Wu, "Energy efficient optimal opportunistic forwarding for delay tolerant networks", IEEE Transactions on Vehicular Technology, in press, published online on 30 August 2010.

[84] R. Lu, X. Lin, H. Zhu, P. H. Ho and X. Shen, “ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications”, Proc. of IEEE INFOCOM’08, Phoenix, USA, April, 2008.

[85] D. Goldschlag, M. Reed, and P. Syverson, “Onion routing for anonymous and private internet connections”, Communications of the ACM, vol. 42, no. 2, pp.84-88, 1999.

[86] R. Dingledine, N. Mathewson, and P. syverson, “Tor: The second–generation onion router”, Proc. of the 13th conference on USENIX Security Symposium (SSYM'04), vol.13, pp. 21-21, 2004.

[87] M. Reiter and A. Rubin, “Crowds: Anonymity for web transactions”, ACM Transactions on Information and System Security, vol. 1, no. 1, pp. 66-92, 1998.

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