Secure and Reliable Routing Protocols for Heterogeneous Multihop Wireless Networks

Mohamed M. E. A. Mahmoud[1], Xiaodong Lin[2], and Xuemin (Sherman) Shen[1], *Fellow, IEEE*

[1]*Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, N2L3G1, Canada*

[2]*Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, Ontario L1H7K4, Canada*

***Abstract*—In this paper, we propose E-STAR for establishing stable and reliable routes in heterogeneous multihop wireless networks. E-STAR combines payment and trust systems with a trust-based and energy-aware routing protocol. The payment system rewards the nodes that relay others' packets and charges those that send packets. The trust system evaluates the nodes' competence and reliability in relaying packets in terms of multi-dimensional trust values. The trust values are attached to the nodes' public-key certificates to be used in making routing decisions. We develop two routing protocols to direct traffic through those highly-trusted nodes having sufficient energy to minimize the probability of breaking the route. By this way, E-STAR can stimulate the nodes not only to relay packets, but also to maintain route stability and report correct battery energy capability. This is because any loss of trust will result in loss of future earnings. Moreover, for the efficient implementation of the trust system, the trust values are computed by processing the payment receipts. Analytical results demonstrate that E-STAR can secure the payment and trust calculation without false accusations. Simulation results demonstrate that our routing protocols can improve the packet delivery ratio and route stability.**

***Index Terms*—**Securing heterogeneous multihop wireless networks, packet dropping and selfishness attacks, trust systems, and secure routing protocols.

1. INTRODUCTION

In multihop wireless networks, when a mobile node needs to communicate with a remote destination, it relies on the other nodes to relay the packets [1]. This multihop packet transmission can extend the network coverage area using limited power and improve area spectral efficiency. In developing and rural areas, the network can be deployed more readily and at low cost. We consider the civilian applications of multihop wireless networks, where the nodes have long relation with the network. We also consider heterogeneous multihop wireless networks (HMWNs), where the nodes' mobility level and hardware/energy resources may vary greatly. HMWNs can implement many useful applications such as data sharing and multimedia data transmission [2]. For example, users in one area (residential neighborhood, university campus, etc) having different wireless-enabled devices (PDAs, laptops, tablets, cell phones, etc) can establish a network to communicate, distribute files, and share information.

In military and disaster-recovery applications, the nodes' behavior is highly predictable because the network is closed and the nodes are controlled by one authority. However, the nodes' behavior is unpredictable in civilian applications for different reasons. The nodes are typically autonomous and self-interested and may belong to different authorities. The nodes also have different hardware and energy capabilities and may pursue different goals. In addition, malfunctioned nodes frequently drop packets and break routes due to faulty hardware or software, and malicious nodes actively break routes to disrupt data transmission.

Since the mobile nodes are battery driven and one of the major sources of energy consumption is radio transmission, selfish nodes are unwilling to lose their battery energy in relaying other users' packets. When more nodes are cooperative in relaying packets, the routes are shorter, the network connectivity is more, and the possibility of network partition is lower. Moreover, since the nodes are equipped with different hardware capability, such as CPU speed and buffer size, the nodes having large hardware resources can perform packet relay more successfully than others. For example, PDAs may not be able to relay packets effectively due to the scarcity of resources. In HMWNs, a route is broken when an intermediate node moves out of the radio range of its neighbors in the route. In addition, some nodes may break routes because they do not have sufficient energy to relay the source nodes' packets and keep the routes connected.

Because of this uncertainty in the nodes' behavior, randomly selecting the intermediate nodes will degrade the routes' stability. It will also endanger the reliability of data transmission and degrade the network performance in terms of packet delivery ratio [3]. Only one intermediate node can break a route, and a small number of incompetent or malicious nodes can repeatedly break routes. When a route is broken, the nodes have to rely on cycles of time-out and route discoveries to re-establish the route. These route discoveries may incur network-wide flooding of routing requests that consume a substantial amount of the network's resources. Breaking the routes increases the packet delivery latency and may cause network partitioning and the multihop communication to fail. Hence, in order to establish stable routes and maintain continuous traffic flow, it is essential to assess the nodes' competence and reliability in relaying packets to make informed routing decisions.

In this paper, we propose E-STAR, a secure protocol for **E**stablishing **STA**ble and reliable **R**outes in HMWNs. E-STAR integrates trust and payment systems with a trust-based and energy-aware routing protocol. The payment system uses credits (or micropayment) to charge the nodes that send packets and reward those relaying packets. Since a trusted party may not be involved in the communication sessions, an offline trusted party (TP) is required to manage the nodes' credit accounts. The nodes compose proofs of relaying packets, called receipts, and submit them to TP. The payment system can stimulate the selfish nodes to relay others' packets to earn credits. It can also enforce fairness by rewarding the nodes that relay more packets such as those at the network center. However, the payment system is not sufficient to ensure route stability. It can stimulate the rational nodes to not break routes to earn credits, but the routes can be broken due to other reasons. Examples for these reasons include low resources, node failure, and malicious attacks.

Trust systems have been used in a wide range of applications, including public key authentication, electronic commerce, supporting decision making, etc [4, 5, 6]. In HMWNs, trust management is essential to assess the nodes' trustworthiness, competence, and reliability in relaying packets. A node's trust value is defined as the degree of belief about the node's behavior, i.e., the probability that the node will behave as expected. The trust values are calculated from the nodes' past behaviors and used to predict their future behavior. For example, there is a strong belief that a node will break a route if it broke a large percentage of routes in the past. Most of the existing trust systems in multihop wireless networks compute a single trust value for each node. However, a single measure may not be expressive enough to adequately depict a node's trustworthiness and competence.

We propose a trust system that maintains multi-dimensional trust values for each node to evaluate the node's behavior from different perspectives. Multi-dimensional trust values can better

Digital Object Indentifier 10.1109/TPDS.2013.138

predict the node's future behavior, and thus help make smarter routing decisions. In our trust system, the nodes that frequently drop packets, break routes, or are not active in relaying packets have low trust values. Moreover, for the efficient implementation of the trust system, TP computes the trust values by processing the payment receipts. A node's trust values are attached to its public-key certificate to be used in making routing decisions.

We develop two trust-based and energy-aware routing protocols, called the *Shortest Reliable Route* (SRR) and the *Best Available Route* (BAR). Our goal is to establish stable routes to reduce the probability of breaking them due to the following reasons: (1) lack of energy: an intermediate node may not have sufficient energy to relay the source node's packets and keep the route connected; and (2) node behavior: the nodes may break routes due to malicious action, malfunction, low hardware resources, etc. SRR protocol establishes the shortest route that can satisfy the source node's requirements including energy, trust, and route length. For BAR protocol, the destination node may learn multiple routes and establishes the most reliable one. Our analytical results demonstrate that E-STAR can secure the payment and trust calculation without false accusations. The simulation results demonstrate that our routing protocols can improve the packet delivery ratio due to establishing stable routes.

The main benefits of integrating the payment and trust systems with the routing protocol can be summarized as follows. First, it fosters trust among the nodes by making knowledge about the nodes' past behavior available. Relaying packets by unknown nodes entails a certain element of risk, so a source node needs to trust the nodes that relay its packets. Second, this integration can deliver messages through reliable routes and allow the source nodes to prescribe their required level of trust. Third, it can punish the nodes that break routes by giving more preference to the highly-trusted nodes in route selection, and thus in earning credits. Fourth, the integration of the payment and trust systems with the routing protocol can punish the nodes that report incorrect energy capability. This is because the routes will be broken at these nodes and their trust values will degrade. Finally, a node may use a greedy strategy: never earn too much unneeded credits and stop relaying others' packets after earning sufficient credits. The integration of the payment and trust systems not only stimulates the nodes to cooperate in relaying packets to earn credits, but also stimulates the wealthy nodes to cooperate to maintain good trust values. This is because the nodes lose trust over time if they do not cooperate. By this way, in addition to payment, trust is another incentive for cooperation.

The main contributions of this paper can be summarized as follows. (1) E-STAR integrates payment and trust systems with the routing protocol with the goal of enhancing route reliability and stability; (2) we propose a multi-dimensional trust system based on processing the payment receipts; (3) E-STAR stimulates the nodes not only to relay others' packets even if they have many credits, but also to stabilize the routes and report their energy capability truthfully to increase their chance to participate in future routes; and (4) we propose trust-based and energy-aware routing protocols to establish stable routes. Unlike most of the existing schemes that aim to identify and mitigate the malicious nodes, E-STAR aims to identify the good nodes and select them in routing.

The remainder of this paper is organized as follows. Section 2 reviews the related works. Section 3 gives the network and adversary models. Section 4 presents E-STAR. Security analysis and performance evaluations are given in Sections 5 and 6, respectively. Conclusions are drawn in Section 7.

2. RELATED WORKS

2.1 Reputation-based Schemes

Reputation-based schemes [3] attempt to identify the malicious nodes that drop packets with a rate more than a pre-defined threshold in order to avoid them in routing. When a node $\mathcal{N}_\text{A}$ sends a packet to the next node in the route ($\mathcal{N}_\text{B}$) to relay to $\mathcal{N}_\text{C}$, $\mathcal{N}_\text{A}$ has to overhear the channel to check whether $\mathcal{N}_\text{B}$ forwards the packet. If $\mathcal{N}_\text{A}$ does not overhear the packet transmission, it assumes that $\mathcal{N}_\text{B}$ has dropped the packet. Each node measures the frequency by which the other nodes drop packets in terms of reputation values. $\mathcal{N}_\text{A}$ increases the reputation value of $\mathcal{N}_\text{B}$ when it observes a packet transmission; otherwise, it decreases the reputation value of $\mathcal{N}_\text{B}$. Once the reputation value degrades to a threshold, $\mathcal{N}_\text{A}$ identifies $\mathcal{N}_\text{B}$ as malicious.

However, there are a number of situations at which monitoring by overhearing the medium does not work: (1) when a node $\mathcal{N}_\text{B}$ relays a packet to $\mathcal{N}_\text{C}$, it is possible that $\mathcal{N}_\text{A}$ cannot overhear the transmission due to another concurrent transmission in $\mathcal{N}_\text{A}$'s neighborhood; and (2) if $\mathcal{N}_\text{B}$ is closer to $\mathcal{N}_\text{A}$ than $\mathcal{N}_\text{C}$, $\mathcal{N}_\text{B}$ could save its energy and circumvent the scheme by adjusting its transmission power to be overheard by $\mathcal{N}_\text{A}$ but less than the required power for reaching the true recipient $\mathcal{N}_\text{C}$. To eliminate using the channel overhearing technique, two-hop ACK technique has been proposed in [7]. $\mathcal{N}_\text{A}$ accuses its neighbor $\mathcal{N}_\text{B}$ of dropping a packet, if $\mathcal{N}_\text{A}$ does not receive an ACK packet from the two-hop-away node $\mathcal{N}_\text{C}$.

Reputation-based schemes suffer from false accusations where some honest nodes are falsely identified as malicious. This is because the nodes that drop packets temporarily, e.g., due to congestion, may be falsely identified as malicious by its neighbors. In order to reduce the false accusations, the schemes should use tolerant thresholds to guarantee that a node's packet dropping rate can only reach the threshold if the node is malicious. However, this increases the missed detections where some malicious nodes are not identified. Moreover, tolerant threshold enables the nodes with high packet dropping rate to participate in routes, and enables the malicious nodes to circumvent the scheme by dropping packets at a rate lower than the scheme's threshold. When a node's reputation value is above the threshold, it does not have incentive to relay packets because it does not bring more utility.

Reputation-based schemes may identify the black-hole attackers that drop all the packets they are supposed to relay. However, they are less effective in detecting the gray-hole attackers that drop a portion of the packets. There is an unavoidable tradeoff between missed detections and false accusations. This is because determining an optimal threshold that can precisely differentiate between the honest and the malicious nodes is a challenge, especially in HMWNs. Using a threshold to determine the trustworthiness of a node is not effective in HMWNs because the nodes' packet-dropping rates vary greatly. Therefore, these schemes cannot guarantee route stability or reliability in HMWNs.

2.2 Payment Schemes

Payment (or incentive) schemes use credits (or micropayment) to encourage the nodes to relay others' packets [8 - 10]. Since relaying packets consumes energy and other resources, packet relaying is treated as a service which can be charged. The nodes earn credits for relaying others' packets and spend them to get their packets delivered. In Sprite [8], for each message, the source node signs the identities of the nodes in the route and the message. Each intermediate node verifies the signature and submits a signed receipt to TP to claim the payment. However, the receipts overwhelm the network because one receipt is composed for each message. To reduce the receipts' number, PIS [9] generates a fixed size receipt per route regardless of the number of messages.

In ESIP [10], the payment scheme uses a communication protocol that can transfer messages from the source node to the destination with limited use of the public key cryptography operations. Public key cryptography is used for only one packet and the efficient hashing operations are used in next packets. Unlike ESIP that aims to transfer messages efficiently, E-STAR aims to establish stable and reliable routes. Although the proposed communication protocol in [10] can be used with E-STAR, we use a simple protocol due to space limitation and to focus on our contributions. In [11], payment is used to thwart the rational packet-dropping attacks, where the attackers drop packets because they do not benefit from relaying packets. A reputation system is also used to identify the irrational packet-dropping attackers once their packet-dropping rates exceed a threshold.

2.3 Trust Systems

Theodorakopoulos et al. [12] analyze the issue of evaluating the trust level as a generalization of the shortest-path problem in an oriented graph, where the edges correspond to the opinion that a node has about other node. The main goal is to enable the nodes to indirectly build trust relationships using exclusively monitored information. In [13], Velloso et al. have proposed a human-based model which builds a trust relationship between nodes in ad hoc network. Without the need for global trust knowledge, they have presented a protocol that scales efficiently for large networks.

In [14], Lindsay et al. have developed an information theoretic framework to quantitatively measure trust and model trust propagation in ad hoc networks. Trust is a measure of uncertainty with its value represented by entropy. The evidence collected for malicious and benign behaviors are probabilistically mapped by following a modified Bayesian approach. The probabilistic estimate of Bayesian approach is then mapped to entropy. In [15], a secure routing protocol with quality of service support has been proposed. The routing metrics are obtained by combing the requirements on the trustworthiness of the nodes and the quality of service of the links along a route.

Table 1: Comparison between E-STAR and reputation-based/payment schemes.

	Reputation-based schemes	Payment schemes	E-STAR
The nodes are motivated to behave well	No	Yes	Yes
Cooperation	Enforced	Stimulated	Stimulated
Enforce fairness	No	Yes	Yes
Establishing stable routes	To some extent	No	Yes
Reputation/trust management technique	Overhearing the wireless medium	------	Processing the payment receipts
Trust values	A single trust value	------	Multi-dimensional trust values
False accusations and missed detections	Yes	No	No

2.4 Comparison

Different from reputation-based schemes that aim to identify the malicious nodes, E-STAR does not suffer from false accusations because it aims to identify the competent nodes in packet relaying and select them in routing. Once a node's trust values fall behind those of the majority of the nodes, the node will almost not participate in routing without the need for determining good thresholds. Moreover, the reputation-based schemes cannot enforce fairness because they force the nodes to relay others' packets without any benefits. For example, although the nodes situated at the network center relay much more packets than those at the periphery, they are not compensated. The nodes may relay packets to avoid punishment, but do not monitor their neighbors to save their resources. This will degrade the scheme's effectiveness. It is nearly impossible to verify how the nodes compute the reputation values, which enables the attackers to launch effective false-accusation/reputation-boost attacks in undetectable way by spreading false bad/good reputation values.

As the period of interaction with any node may be brief due to the nodes' mobility, the reputation-based schemes may not have sufficient time to precisely assess the nodes' behavior. However, E-STAR can evaluate the nodes' trust values accurately because it can monitor/evaluate the nodes' behavior over different times and sessions. The payment schemes alone are not sufficient for establishing stable routes that require selecting the nodes that behaved well in the past and have sufficient energy. Relaying others' packets and submitting the receipts are beneficial in E-STAR for earning credits, but relaying others' packets and performing the monitoring in reputation-based schemes consumes the nodes' resources without direct benefits. Table 1 summarizes the main differences between E-STAR and reputation-based/payment schemes.

3. System Models

3.1 Network Model

The considered HMWN has mobile nodes and offline trusted party (TP) whose public key is known to all the nodes. The mobile nodes have different hardware and energy capabilities. The network is used for civilian applications, its lifetime is long, and the nodes have long relation with the network. Thus, with every interaction, there is always an expectation of future reaction. Each node has a unique identity and public/private key pair with a limited-time certificate issued by TP. Without a valid certificate, the node cannot communicate nor act as an intermediate node. TP maintains the nodes' credit accounts and trust values. Each node contacts TP to submit the payment receipts and TP updates the involved nodes' payment accounts and trust values. This contact can occur via cellular networks or Internet.

3.2 Adversary Model

The adversaries have full control on their nodes. They can change the nodes' normal operation and obtain the cryptographic credentials. They may attempt to attack the payment system to steal credits, pay less, or communicate for free. Some adversaries may report incorrect energy capability to increase their chance to be selected by the routing protocol, e.g., to earn more credits. The adversaries may also attempt to attack the trust system to falsely augment their trust values to increase their chance to participate in routes. They may try to defame other nodes' trust values. Attackers may launch denial-of-service attacks by breaking the communication routes intentionally. When a node $\mathcal{N}_B$ receives packets from $\mathcal{N}_A$ to forward to the next node in the route, $\mathcal{N}_B$ drops the packets and keeps silent to let $\mathcal{N}_A$ believe that $\mathcal{N}_B$ is out of transmission range and the link between them is broken. These attacks may be launched by compromised, malfunctioned, or low-resource nodes.

The mobile nodes are probable attackers but TP is fully secure. The nodes are autonomous and self-interested and thus motivated to misbehave, but TP is run by an operator that is interested in ensuring the network secure operation.

4. The Proposed E-STAR

Fig. 1 shows that E-STAR has three main phases. In *Data Transmission* phase, the source node sends messages to the destination node. In *Update Credit-Account* and *Trust Values* phases, TP determines the charges and rewards of the nodes and updates the nodes' trust values. Finally, in *Route Establishment* phase, trust-based and energy-aware routing protocol establishes stable communication routes.

4.1 Data Transmission Phase

Let the source node $\mathcal{N}_S$ send messages to the destination node $\mathcal{N}_D$ through a route with the intermediate nodes $\mathcal{N}_X$, $\mathcal{N}_Y$, and $\mathcal{N}_Z$. The route is established by the routing protocols that will be discussed in Subsection 4.3. For the *i*th data packet, $\mathcal{N}_S$ computes the signature $\xi_S(i) = \{H(H(m_i), ts, R, i)\}K_{S+}$ and sends the packet $<R, ts, i, m_i, \xi_S(i)>$ to the first node in the route. R, ts, and m_i are the concatenation of the identities of the nodes in the route ($R = ID_S, ID_X, ID_Y, ID_Z, ID_D$ in Fig. 2), the route establishment time stamp, and the *i*th message, respectively. *H*(d) is the hash value resulted from hashing the data d using the hash function *H(). {d}*K_{S+} is the signature of d with the private key of $\mathcal{N}_S$. The purpose of the source node's signature is to ensure the message's authenticity and integrity and secure the payment by enabling TP to ensure that $\mathcal{N}_S$ has sent *i* messages. Each intermediate node verifies ξ_S(i) and stores ξ_S(i) and $H(m_i)$ for composing the receipt. It also removes the previous ones (ξ_S(i-1) and $H(m_{i-1})$) because ξ_S(i) is enough to prove transmitting *i* messages. Signing $H(m_i)$ instead of m_i can reduce the receipt size because the smaller-size $H(m_i)$ is attached to the receipt instead of m_i.

The destination node generates a one-way hash chain by iteratively hashing a random value h_S *S* times to obtain the hash chain $\{h_S, h_{S-1},\ldots, h_1, h_0\}$, where $h_{i-1} = H(h_i)$ for $1 \le i \le S$ and h_0 is called the root of the hash chain. The node signs h_0 and *R* to authenticate the hash chain and links it to the route, and sends the signature to the source node in route establishment phase. In order to acknowledge receiving the message m_i, the destination node sends ACK packet containing the preimage of the last released hash chain element or h_i. Each intermediate node verifies the hash chain element by making sure that h_{i-1} is obtained from hashing h_i, and saves h_i for composing the receipt and removes h_{i-1}. The underlying idea is that ξ_S(i) and h_i are undeniable proofs for sending and receiving *i* messages, respectively.

Each node in the route composes a receipt and submits it when it has a connection to TP to claim the payment and update its trust values. A receipt is a proof for participating in a route and sending, relaying, or receiving a number of messages. A receipt contains *R*, *ts*, *i*, $H(m_i)$, h_0, h_i, C_m, and an undeniable cryptographic token for preventing payment manipulation. C_m is data that depends on the used routing protocol, such as the number of messages the intermediate nodes commit to relay. The cryptographic token contains the hash value of the last source node's signature and *Auth_Code*. *Auth_Code* is the authentication code that authenticates the hash chain and the intermediate nodes to hold them accountable for breaking the route. More details about C_m and *Auth_Code* will be given in Subsection 4.3. If *i* messages are delivered, the format of the receipt is <*R*, *ts*, *i*, $H(m_i)$, h_0, h_i, C_m, $H(\xi_S$(i), *Auth_Code*)>. ξ_S(i) and *Auth_Code* are hashed to reduce the receipt's size.

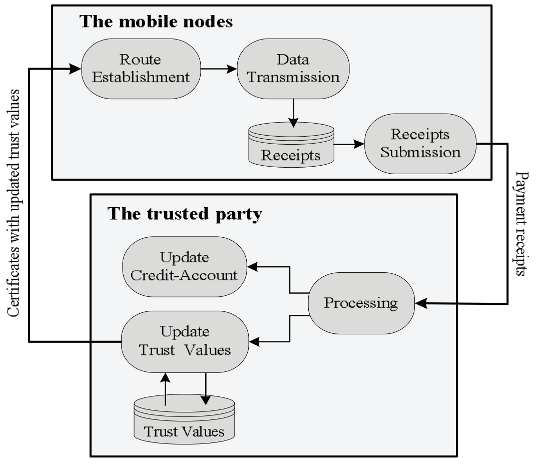


Fig. 1: The architecture of E-STAR.

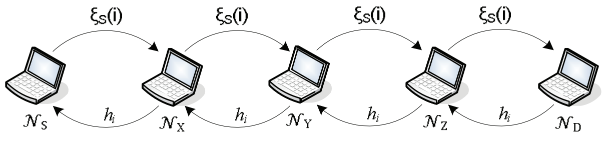


Fig. 2: The exchanged cryptographic tokens during data transmission.

4.2 *Update Credit Account and Trust Values Phase*

Once TP receives a receipt, it first checks if the receipt has been processed before using its unique identifier (*R*, *ts*). Then, it verifies the credibility of the receipt by computing the nodes' signatures ($\xi_S(i)$ and *Auth_Code*) and hashing them. The receipt is valid if the resultant hash value is identical to the receipt's cryptographic token. TP verifies the destination node's hash chain by making sure that hashing h_i *i* times produces h_0. TP clears the receipt by rewarding the intermediate nodes and debiting the source and destination nodes. The number of sent messages (*i*) is signed by the source node and the number of delivered messages can be computed from the number of hashing operations to obtain h_0 from h_i.

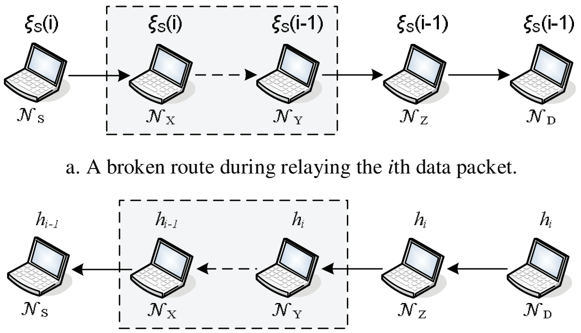
The notion of trust used in this paper is defined as the degree of belief, the expectation, or the probability that a node will act in a certain way in the future based on the node's past behavior [6]. Trust values are calculated from the past behavior to predict the expected future behavior. For instance, people will not assign critical jobs to someone with a record of failure since there is a good reason to believe that he will not get the job done properly. Similarly, if a node has broken a large percentage of routes in the past, there is a strong belief that this node will break routes with high probability in the future, and thus the routing protocol should avoid it. The trust values are computed to depict the nodes' reliability and competence in relaying packets.

Considering trust in routing decisions is essential in HMWN that is characterized by uncertainty in the nodes' behavior because they are autonomous and self-interested. A trust relationship is never absolute, but it is *context-dependent* in the sense that a node's trust value depicts its ability to perform a specific action. For example, Alice may trust Bob to repair her computer but she may not trust Bob to repair her car. Trust is also *dynamic* or *time-sensitive*, so TP has to periodically evaluate the nodes' trustworthiness, i.e., a trust value at time *t* may be different from its value at another time *t'*. In order to capture the dynamicity of trust, it should be expressed as a continuous value rather than binary or even discrete. Also, a continuous variable can represent uncertainty better than a binary variable.

The underlying idea of computing the nodes' trust values is that a packet is relayed by a node if a successor node in the route reports receiving the packet. The possession of ξ_S(i) by an intermediate node $\mathcal{N}_Y$ entails that all the nodes between $\mathcal{N}_S$ and $\mathcal{N}_Y$ have indeed relayed *i* packets. Similarly, the possession of h_i by $\mathcal{N}_Y$ entails that it has relayed *i* delivered packets and received the *i*th ACK, but all the nodes between $\mathcal{N}_D$ and $\mathcal{N}_Y$ have indeed forwarded the ACK packet. For example, in Fig. 3(a), $\mathcal{N}_D$ received *i-1* packets because it submits the source node's signature for *i-1* packets, and thus the intermediate nodes relayed *i-1* packets. It is obvious that the link between $\mathcal{N}_X$ and $\mathcal{N}_Y$ is broken because they submit receipts for *i* and *i-1* messages, respectively. TP should not accuse only $\mathcal{N}_X$ of breaking the route because $\mathcal{N}_Y$ may break the route but submits ξ_S(i-1) instead of ξ_S(i) to circumvent TP. In other words, $\mathcal{N}_X$ and $\mathcal{N}_Y$ received *i* packets but relayed only *i-1* packets. $\mathcal{N}_Z$ in Fig 3(a) received *i*-1 packets and relayed all of them because $\mathcal{N}_D$ submits ξ_S(i-1). In Fig 3(b), $\mathcal{N}_Y$ and $\mathcal{N}_X$ are accused of breaking the route during relaying the *i*th ACK packet because they submit h_i and h_{i-1}, respectively. A route is not broken if all the nodes submit ξ_S(i) and h_i.

It is fair to increase the trust values of the nodes that are not in broken links, because they relayed packets truthfully. Conversely, the trust system decreases the trust values of the two nodes in a broken link. However, the nodes that frequently break routes will be distinguishable by aggregating the nodes' behaviors in different sessions. The rationale here is that *the nodes that break routes more frequently (because of genuine behavior) are accused more and suffer from more trust degradation*. Moreover, since our protocols use relative and not absolute trust values, the nodes that frequently break routes will be distinguishable because their trust values will be less than those of the majority of the nodes. Since neighbors change due to the nodes' mobility, the accusations are distributed and will not be focused on few nodes.

A node can protect its trust values by not involving itself in routes with a neighbor that frequently breaks routes or has low trust values. Additionally, we are sure that the nodes that are not in a broken link did not break the route, which coincides with our objective of identifying good nodes.



a. A broken route during relaying the *i*th data packet.

b. A broken route during relaying the *i*th ACK packet.

Fig. 3: Evaluating the nodes' trust values.

Our trust system adopts multi-dimensional trust management framework in which the notion of trustworthiness is further classified into several attributes (or dimensions). Each attribute can indicate to what extent the node will conduct one specific action. We use multi-dimensional trust values instead of one trust value to precisely predict the nodes' future behavior. The trustworthiness of a node $\mathcal{N}_k$ is assessed in n-dimensional vector of numeric values $\tau_k = [\tau_k^{(1)}, \tau_k^{(2)}, \ldots, \tau_k^{(n)}]$, where $\tau_k^{(i)}$ stands for the *i*-th dimension of the trustworthiness of $\mathcal{N}_k$. Each dimension $\tau_k^{(i)}$ corresponds to one action $\beta_k^{(i)}$. $\tau_k^{(i)}$ depicts the probability that $\mathcal{N}_k$ will conduct $\beta_k^{(i)}$ in an appropriate manner, and thus the higher the value of $\tau_k^{(i)}$ is, the more likely $\mathcal{N}_k$ will conduct $\beta_k^{(i)}$. $\tau_k^{(i)}$ can be assigned any real value in the range of [0, 1] signifying a continuous range from complete distrust (0) to complete trust (+1), i.e., $\tau_k^{(i)} \in [0, 1]$, $\forall\, i \in \{1, 2, \ldots, n\}$.

$\tau_k^{(1)}$ depicts the probability that $\mathcal{N}_k$ will relay a packet successfully. From Eq. 1, $\tau_k^{(1)}$ is the total number of packets that are relayed by $\mathcal{N}_k$ to the total number of incoming packets to be relayed by $\mathcal{N}_k$ in the last ω sessions. Obviously, if $\mathcal{N}_k$ drops a large portion of packets, $\tau_k^{(1)}$ will be very low. The trust value $\tau_k^{(2)}$ depicts the probability that $\mathcal{N}_k$ will not break a route. From Eq. 2, $\tau_k^{(2)}$ is the ratio of routes $\mathcal{N}_k$ did not break in the last ω sessions. If $\tau_k^{(2)}$ is low, this does not necessarily mean that $\tau_k^{(1)}$ is low, e.g., the nodes that break routes after relaying a large number of packets.

$\tau_k^{(3)}$ is the probability of relaying at least δ packets in a session. From Eq. 3, $\tau_k^{(3)}$ is the percentage of sessions that $\mathcal{N}_k$ relayed at least δ packets in the last ω sessions. $\tau_k^{(3)}$ depicts the node's ability to keep a route connected for a minimum number of packets. This trust value should be low for the high-mobility nodes and the nodes that participate only in short sessions. The trust values are calculated only from the last ω sessions, e.g., 50 to 100 sessions, because the recent steady behavior is a better predictor for future behavior than behaviors observed long time ago. However, considering only the most recent behavior (very small ω) can yield a distorted picture of the nodes' behavior as few observed instances are not enough to measure the trend of the behavior.

Since there is a stronger belief in the trust values that are computed from recent sessions, $\tau_k^{(4)}$ given in Eq. 4 depicts how $\mathcal{N}_k$ was recently active in participating in sessions, or to what extent the other trust values are fresh, i.e., computed from recent sessions. $\tau_k^{(4)}$ is the total number of sessions $\mathcal{N}_k$ participated in, in the last period t over a normalizing factor ($\mathcal{M}$) that depicts the maximum number of sessions a trusted node should participate in, in t. Note that the maximum value of $\tau_k^{(4)}$ is 1, and thus $\tau_k^{(4)} = 1$ if $\mathcal{M} <$ the number of sessions $\mathcal{N}_k$ participated in, in t. Obviously, $\tau_k^{(4)}$ decreases over time if $\mathcal{N}_k$ does not participate in sessions. The centralized trust system can determine a good value for $\mathcal{M}$ by observing the maximum number of sessions the active nodes participate in, in period t. Even if the value of $\mathcal{M}$ is not optimal, $\tau_k^{(4)}$ is still valuable measure because the trust values are relative and not absolute.

Humans are able to know each other better as time goes by and the same idea applies here. We can trust more the older nodes that spent more time in the network than the new nodes that joined the network recently, because TP had enough time to assess their behavior. The basic idea is to use the time the nodes spent in the network as a metric in selecting the intermediate nodes. The certificate of node $\mathcal{N}_K$ ($Cert_K$) contains, its identity (ID_K) and public key ($K_{K\text{-}}$), certificate issuing time (t_i), the time $\mathcal{N}_K$ joined the network (t_j), the certificate expiration time (t_e), the trust values (τ_k), and the TP's signature. The certificate of $\mathcal{N}_K$ has the following format:

$$\text{TP} \rightarrow \mathcal{N}_K\text{: } Cert_K = ID_K, t_e, t_j, t_i, K_{K\text{-}}, \tau_K, \{H(ID_K, t_e, t_j, t_i, K_{K\text{-}}, \tau_K)\}K_{TP+}$$

$$\tau_k^{(1)} = \frac{\text{№ of packets that are relayed in the last } \omega \text{ sessions}}{\text{Total № of incoming packets in the last } \omega \text{ sessions}} \tag{1}$$

$$\tau_k^{(2)} = 1 - \frac{\text{№ of sessions broken by } \mathcal{N}_k \text{ in the last } \omega \text{ sessions}}{\omega} \tag{2}$$

$$\tau_k^{(3)} = \frac{\text{№ of sessions that } \mathcal{N}_k \text{ relayed at least } \delta \text{ packets}}{\omega} \tag{3}$$

$$\tau_k^{(4)} = \frac{\text{№ of sessions } \mathcal{N}_k \text{ paricipated in in the period } t}{\mathcal{M}} \tag{4}$$

$$\tau_{WXYZ}^{(i)} = \tau_W^{(i)} \times \tau_X^{(i)} \times \tau_Y^{(i)} \times \tau_Z^{(i)} \tag{5}$$

Trust values are used to decide which nodes to select/avoid in routing. Since a trust value depicts the probability that the node conducts an action, route reliability can be computed using its nodes' trust values to give probabilistic information about the route stability and lifetime. This information is very useful for establishing stable routes and selecting proper routes that can satisfy the source nodes' requirements. Eq. 5 gives the route reliability of a route with intermediate nodes $\mathcal{N}_W$, $\mathcal{N}_X$, $\mathcal{N}_Y$, and $\mathcal{N}_Z$, $\forall\, i \in \{1, 2, 3, 4\}$, to depict the probability that the action $\beta_k^{(i)}$ will be conducted. For example, if i = 1, Eq. 5 gives the probability that a packet will reach the destination node through the intermediate nodes $\mathcal{N}_W$, $\mathcal{N}_X$, $\mathcal{N}_Y$, and $\mathcal{N}_Z$. Similarly, $\tau_{WXYZ}^{(2)}$ and $\tau_{WXYZ}^{(3)}$ give the probability that the route will not be broken and the probability that at least δ packets will be transmitted through the route, respectively.

In order to clarify the importance of using the route reliability in routing decisions, numerical examples are given in Table 2. From cases one and two, the low-trusted node such as $\mathcal{N}_X$ in case two, nearly does not have any chance to participate in routes because it significantly degrades the route reliability. Although the nodes of cases one and three have the same trust values, the route reliability of case three is larger, which demonstrates that the shortest routes are inherently preferable. If i = 1, the probability of delivering a packet through the route of case four is nearly zero because the nodes have very low trust values. This case demonstrates the importance of using trust in routing decisions. With i = 2 and comparing cases one and four, it is obvious that choosing a good route is important for route stability. From cases one and five, a longer route with trusted nodes is preferable than a shorter route with malicious nodes. From cases three and six, once the trust value of $\mathcal{N}_Y$ slightly decreases beyond the other nodes' trust values, the route reliability decreases, and thus the node's chance to participate in routes decreases.

Since a connection to TP may not be available on a regular basis, the receipts may be submitted after some time, and thus the trust values may be updated after some delay. This is acceptable because: (1) the routing protocol is sensitive to any degradation in the trust values; and (2) the nodes' behavior is repetitive, i.e., for a normal node the probability of breaking a route is fixed.

Table 2: Numerical examples for route reliability.

Case	$\tau_W^{(i)}$	$\tau_X^{(i)}$	$\tau_Y^{(i)}$	$\tau_Z^{(i)}$	$\tau_{WXYZ}^{(i)}$
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
5	0.8	0.2	0.8	----	0.128
6	0.8	0.8	0.75	----	0.48

$$\mathcal{N}_S \rightarrow *\text{:}\ \ \text{RREQ}, D = (ID_S, ID_D, H_{max}, ts, T_r, E_r), \{D\}K_{S+}, Cert_S$$

$$\mathcal{N}_X \rightarrow *\text{:}\ \ \text{RREQ}, D, \underline{\boldsymbol{ID_X}}, \underline{\boldsymbol{\{\{D\}K_{S+}\}K_{X+}}}, Cert_S, \underline{\boldsymbol{Cert_X}}$$

$$\mathcal{N}_Y \rightarrow *\text{:}\ \ \text{RREQ}, D, ID_X, \underline{\boldsymbol{ID_Y}}, \underline{\boldsymbol{\{\{\{D\}K_{S+}\}K_{X+}\}K_{Y+}}}, Cert_S, Cert_X, \underline{\boldsymbol{Cert_Y}}$$

$$\mathcal{N}_Z \rightarrow *\text{:}\ \ \text{RREQ}, D, ID_X, ID_Y, \underline{\boldsymbol{ID_Z}}, \underline{\boldsymbol{Sig = (\{\{\{\{D\}K_{S+}\}K_{X+}\}K_{Y+}\}K_{Z+})}}, Cert_S, Cert_X, Cert_Y, \underline{\boldsymbol{Cert_Z}}$$

a. The format of RREQ packet.

$$\mathcal{N}_D \rightarrow \mathcal{N}_Z\text{: RREP}, R = (ID_S, ID_X, ID_Y, ID_Z, ID_D), h_0, \{Sig, h_0\}K_{D+}, Cert_D$$

$$\mathcal{N}_Z \rightarrow \mathcal{N}_Y\text{: RREP}, R, h_0, \{Sig, h_0\}K_{D+}, Cert_D, \underline{\boldsymbol{Cert_Z}}$$

$$\mathcal{N}_Y \rightarrow \mathcal{N}_X\text{: RREP}, R, h_0, \{Sig, h_0\}K_{D+}, Cert_D, Cert_Z, \underline{\boldsymbol{Cert_Y}}$$

$$\mathcal{N}_X \rightarrow \mathcal{N}_S\text{: RREP}, R, h_0, \{Sig, h_0\}K_{D+}, Cert_D, Cert_Z, Cert_Y, \underline{\boldsymbol{Cert_X}}$$

b. The format of RREP packet.

Fig. 4: The format of RREQ and RREP packets in the SRR routing protocol.

4.3 Route Establishment Phase

In this section, we present two routing protocols called the *Shortest Reliable Route* (SRR) and the *Best Available Route* (BAR). SRR establishes the shortest route that can satisfy the source node's trust, energy, and route-length requirements, but the destination node selects the best route in the BAR protocol. The routing protocols have three processes: i) *Route Request Packet* (RREQ) delivery; ii) Route selection; and iii) *Route Reply Packet* (RREP) delivery.

4.3.1 The SRR Routing Protocol

To establish a route to the destination node $\mathcal{N}_D$, the source node $\mathcal{N}_S$ broadcasts RREQ packet and waits for RREP packet. The source node embeds its requirements in the RREQ packet, and the nodes that can satisfy these requirements broadcast the packet. The destination node establishes the shortest route that can satisfy the source node's requirements. The rationale of the SRR protocol is that *the node that satisfies the source node's requirements is trusted enough to act as a relay*. The protocol is useful to establish a route that avoids the low-trusted nodes.

RREQ: As shown in Fig. 4(a), the RREQ packet contains the packet type identifier "RREQ", the identities of the source and destination nodes (ID_S and ID_D), the maximum number of intermediate nodes (H_{max}), the time stamp of the route establishment (*ts*), the trust and energy requirements (T_r = [$\tau^{(1)}$, $\tau^{(2)}$, $\tau^{(3)}$, $\tau^{(4)}$] and E_r), and the source node's signature and certificate. H_{max} can limit the propagation area of the packet and *ts* can ensure the freshness of the request. The trust requirements are the minimum trust values an intermediate node can have, e.g., if T_r = [0.7, 0, 0, 0], the first trust value of the intermediate nodes should be at least 0.7 regardless of the other trust values. E_r is the minimum number of packets an intermediate node should relay in the session, which is related to the node's available battery energy. If a node breaks the route before relaying E_r messages, its trust values will decrease. The minimum route reliability is bounded by $\tau^{(i)}$ raised to the H_{max}-th power.

Each intermediate node ensures that it can satisfy the source node's trust/energy requirements, the current time is within a proper range of *ts*, and the number of intermediate nodes is fewer than H_{max}. It also verifies the packet's signature(s) using the public keys extracted from the nodes' certificates. These verifications are necessary to ensure that the packet is sent and relayed by legitimate nodes and the nodes can satisfy the trust requirements because their trust values are signed by TP. The intermediate node signs the packet's signature forming a chain of signatures of the nodes that broadcast the packet. This signature authenticates the intermediate node and proves that the node is the certificate holder and thus the attached trust values belong to the node. The signature also enables the trust system to make sure that the intermediate nodes have indeed participated in the route to hold them accountable for breaking the route. Finally, the intermediate node broadcasts the packet after adding the signature chain and its identity and certificate. If a node receives the same request packet from different nodes, it processes only the first packet and discards the subsequent packets.

Route Selection: If there is a route that can satisfy the source node's requirements, the destination node receives at least one RREQ packet. The destination node composes the RREP packet for the route traversed by the first received RREQ packet, and sends it to the source node. This route is the shortest one that can satisfy the source node's requirements. The source node's requirements cannot be achieved if it does not receive the RREP packet within a time period. It can initiate a second RREQ packet but with more flexible requirements, e.g., by increasing H_{max} and/or decreasing E_r and T_r, or revert to the BAR protocol.

RREP: As shown in Fig. 4(b), the RREP packet contains the packet type identifier "RREP", the identities of the nodes in the route (R), h_0, the destination node's certificate, and the nodes' authentication code ($Auth\_Code$ = $\{Sig, h_0\}K_{D+}$), where *Sig* is the signature chain of the RREQ (Sig = $\{\{\{\{D\}K_{S+}\}K_{X+}\}K_{Y+}\}K_{Z+}$). h_0 is the root of the hash chain created by the destination node by iteratively hashing a random value h_S S times, where $h_{i-1} = H(h_i)$. The destination node's signature authenticates the hash chain and links it to the session. It also authenticates the destination node and proves to TP that $\mathcal{N}_D$ has indeed participated in the session. *Auth_Code* can authenticate the nodes with less packet space than attaching separate signatures. Each intermediate node verifies *Auth_Code*, adds its certificate, and relays the packet as shown in Fig. 4(b). It also stores h_0, *Auth_Code*, and C_m = <H_{max}, T_r, E_r> to be used for composing the receipt.

From the RREQ packet, each intermediate node can authenticate the source node and the in-between intermediate nodes, and from the RREP packet, each intermediate node can authenticate the destination node and the in-between nodes. For example, in Fig. 4, $\mathcal{N}_Y$ can authenticate $\mathcal{N}_S$ and $\mathcal{N}_X$ from the signature chain ($\{\{D\}K_{S+}\}K_{X+}$) attached to the RREQ, and authenticate $\mathcal{N}_D$ and $\mathcal{N}_Z$ from the signature ($\{Sig, h_0\}K_{D+}$) attached to the RREP. In order to reduce the number of verifications, $\mathcal{N}_Y$ stores the signature chain $\{\{\{D\}K_{S+}\}K_{X+}\}K_{Y+}$ and decrypts $\{Sig, h_0\}K_{D+}$ until it obtains $\{\{\{D\}K_{S+}\}K_{X+}\}K_{Y+}$. By this way, each node in the route performs one signature and $2(R_L$-1) verifications to verify the nodes' authentication code and certificates, where R_L is the number of nodes in the route including the source and destination nodes. Verifying the *Auth_Code* is important to ensure the receipt's integrity to secure the trust/payment systems. The source node verifies the *Auth_Code* and the nodes' certificates to make sure that the nodes satisfy its trust requirements and the intended destination node was reached, then it starts data transmission.

4.3.2 The BAR Routing Protocol

RREQ: As shown in Fig. 5, the RREQ packet contains ID_S, ID_D, *ts*, H_{max}, the source node's certificate and signature (Sig_S), and the number of messages it needs to send (E_r(S)). For the first received RREQ packet, an intermediate node $\mathcal{N}_X$ broadcasts the packet after attaching its identity and certificate, the number of messages it commits to relay ($E_r(X)$). Unlike the SRR protocol, $E_r(X)$ can be fewer than $E_r(S)$. $\mathcal{N}_X$ also signs the concatenation of $E_r(X)$ and the signature received in the RREQ packet. $E_r(X)$ not only depends on the available battery energy in $\mathcal{N}_X$, but also on other factors such as the cooperation strategy (or the node's willingness for relaying packets) and the link quality and stability. For example if the links between $\mathcal{N}_X$ and its two neighbors in the route are unstable, it can decrease its $E_r(X)$ to decrease the probability of breaking the route. The nodes are motivated to report correct energy commitments to avoid breaking the route and thus degrading their trust values.

$\mathcal{N}_S \rightarrow *$: RREQ, $D = (ID_S, ID_D, H_{max}, ts, E_r(S))$, $Sig_S = \{D\}K_{S+}$, $Cert_S$

$\mathcal{N}_X \rightarrow *$: RREQ, D, $\underline{\boldsymbol{ID_X}}$, $\underline{\boldsymbol{E_r(X)}}$, $\underline{\boldsymbol{Sig_X = \{Sig_S, E_r(X)\}K_{X+}}}$, $Cert_S$, $\underline{\boldsymbol{Cert_X}}$

$\mathcal{N}_Y \rightarrow *$: RREQ, D, ID_X, $E_r(X)$, $\underline{\boldsymbol{ID_Y}}$, $\underline{\boldsymbol{E_r(Y)}}$, $\underline{\boldsymbol{Sig_Y = \{Sig_X, E_r(Y)\}K_{Y+}}}$, $Cert_S$, $Cert_X$, $\underline{\boldsymbol{Cert_Y}}$

$\mathcal{N}_Z \rightarrow *$: RREQ, D, ID_X, $E_r(X)$, ID_Y, $E_r(Y)$, $\underline{\boldsymbol{ID_Z}}$, $\underline{\boldsymbol{E_r(Z)}}$, $\underline{\boldsymbol{Sig_Z = \{Sig_Y, E_r(Z)\}K_{Z+}}}$, $Cert_S$, $Cert_X$, $Cert_Y$, $\underline{\boldsymbol{Cert_Z}}$

Fig. 5: The format of RREQ packet in the BAR routing protocol.

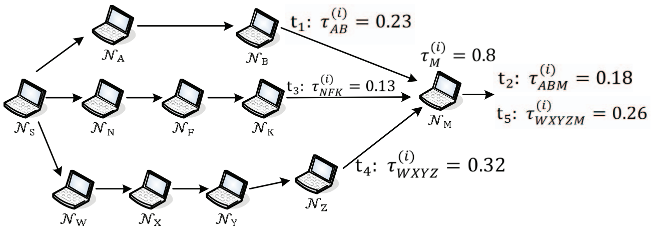


Fig. 6: Broadcasting the RREQ packet in the BAR routing protocol.

Blind RREQ flooding generates few routes because each node broadcasts the packet once, which disables potential better routes. To solve this issue, BAR allows each node to broadcast the RREQ more than once if the route reliability or lifetime of the recently received packet is greater than the last broadcasted packet. The route lifetime is the minimum number of packets the intermediate nodes commit to relay, e.g., if the commitments of the intermediate nodes are $E_r(X) = 10$, $E_r(Y) = 8$, and $E_r(Z) = 17$, the route lifetime is 8 packets.

In Fig. 6, $\mathcal{N}_M$ receives the first RREQ at time t_1 with route reliability ($\tau_{AB}^{(1)}$) of 0.23. At t_2, $\mathcal{N}_M$ broadcasts the packet, where the updated route reliability is $\tau_{ABM}^{(1)} = \tau_{AB}^{(1)} \times \tau_M^{(1)} = 0.18$. At t_3, $\mathcal{N}_M$ receives the second RREQ for the same request but it discards the packet because $\tau_{NFK}^{(1)}$ is less than $\tau_{AB}^{(1)}$. At t_4, $\mathcal{N}_M$ receives RREQ packet with $\tau_{WXYZ}^{(1)}$ that is larger than the route reliability of the last broadcasted packet ($\tau_{AB}^{(1)}$), so it broadcasts the packet at t_5. In this example, we consider only the route reliability of the first trust value for simplicity, but the other trust values can also be considered using weighting factors. The source node can attach the weighting vector $[\alpha_1, \alpha_2, \alpha_3, \alpha_4]$ to the RREQ, where $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$. Node $\mathcal{N}_M$ calculates the total route reliability as follows ($\alpha_1 \times \tau_{ABM}^{(1)} + \alpha_2 \times \tau_{ABM}^{(2)} + \alpha_3 \times \tau_{ABM}^{(3)} + \alpha_4 \times \tau_{ABM}^{(4)}$), and broadcasts a RREQ packet if its total route reliability is larger than that of the last broadcasted packet. The source node can also add some conditions to the RREQ packet such as the minimum time a node has been in the network.

To reduce the number of RREQ broadcastings, when an intermediate node receives a RREQ, it introduces a Wait Period to collect subsequent packets, if any, traveling through different routes and then selects some. It selects the most reliable route having at least lifetime of $E_r(S)$; and if this route does not exist, it selects multiple RREQ packets with at least total lifetime of $E_r(S)$ in such a way that reduces the RREQ packets' number and maximizes the reliability.

Route Selection: After receiving the first RREQ packet, the destination node waits for a while to receive more RREQ packets if there are. Then, it selects the best available route if a set of feasible routes are found. If there are multiple routes with lifetimes at least $E_r(S)$, the destination node selects the most reliable route, otherwise, it establishes multiple routes with at least total lifetime of $E_r(S)$ in such a way that reduces the routes' number and maximizes the reliability. The destination node should not select multiple routes with common node(s) (if possible) to disallow one node to break the routes.

RREP: This phase is identical to that of the SRR routing protocol, but in RREP packet, Sig is the signature chain in the RREQ packet, i.e., $Sig = \{\{\{\{D\}K_{S+}, E_r(X)\}K_{X+}, E_r(Y)\}K_{Y+}, E_r(Z)\}K_{Z+}$, and the nodes' energy commitments ($E_r(S)$, $E_r(X)$, $E_r(Y)$, $E_r(Z)$) are attached. Similar to SRR, each node stores h_0, *Auth_Code*, and C_m for composing the receipt, where $C_m = E_r(S), E_r(X), E_r(Y), E_r(Z), H_{max}$.

5. Security Analysis

Securing the payment and trust calculation are based on the following well known cryptographic properties: (1) forging or modifying a signature without knowing the private key is infeasible; (2) deriving the private keys from the public ones is infeasible; (3) computing the hash value of a signature without computing the signature is infeasible; and (4) computing the hash function's input from its output is infeasible. The hash function is unidirectional in the sense that it is feasible to compute $H(X)$ from X, but it is infeasible to compute X from $H(X)$. These cryptographic properties are used to enable TP to make sure that the source, intermediate, and destination nodes have indeed participated in a route and to verify the number of transmitted, received, and relayed messages by each node. They also enable the intermediate nodes to compose valid receipts and verify them.

The number of transmitted messages by a node is undeniable because it is signed by the node and the number of received messages by the destination node is also undeniable because it is infeasible to compute h_i from h_{i-1}. The number of received/relayed messages by an intermediate node cannot be manipulated because the node cannot modify the source node's signatures or manipulate the hash chain. Digital signature and hash function are used in such a way that can make the receipts undeniable, unmodifiable, and unforgeable. The source/destination nodes cannot deny initiating the session and the amount of payment, and the intermediate nodes cannot manipulate/forge receipts to steal credits.

The autonomous mobile nodes are stimulated and not forced to participate in routes and relay other nodes' packets by using their devices. However, frequently dropping packets and breaking routes are an abuse due to disrupting the network proper operation. In black-hole attack, the attackers involve themselves in routes with the intention of dropping all the packets they are supposed to relay. In Gray-hole attack, the attackers selectively drop packets instead of dropping all the packets. Gray-hole attack is more difficult to detect in HMWNs because the nodes' packet-dropping rates vary greatly. E-STAR can effectively exclude the black-hole and gray-hole attackers from routing once their trust values become less than the other nodes' trust values. Even if the gray-hole attackers drop a low portion of packets, they have little chance to participate in routes. This is because the honest nodes that do not drop packets intentionally will have higher trust values and will be selected by the routing protocol.

Greedy nodes may overload themselves by participating in many sessions simultaneously to collect more credits. This behavior will create congested nodes that drop packets once their buffers are full. E-STAR can discourage this behavior because it reduces these nodes' trust values.

E-STAR uses multiple levels of trust as follows. (1) the micropayment stimulates the nodes to behave well to earn credits; (2) using certificates signed by a trusted party enables the nodes to ensure that the other nodes are members in the network; and (3) the trust values computed by the trust system is a trust level that is based on evaluating the nodes' behaviors. Trust values are usually computed from the nodes' past behaviors to predict their future behavior because the nodes' behavior is usually repetitive. Continuously evaluating the nodes' trust values is necessary to track the change in the nodes' behavior. Some nodes may change their behavior after gaining high trust values due to faulty hardware or software or malicious action.

In E-STAR, once a node's trust values drop beyond those of the other nodes, it has little chance to participate in routes, as discussed in Case 6 in Table 2. In reputation-based schemes, it is important but difficult to determine good threshold and initial reputation values. These values will have direct impact on the schemes' effectiveness in terms of false accusation and missed detection ratios. Nevertheless, E-STAR does not use threshold and determining good initial trust values is not problematic because of using relative (not absolute) trust metrics in route selection. BAR selects the most reliable route regardless of the absolute value of the nodes' trust values. In SRR, a source node can reduce its trust requirement if route discovery fails.

Since the behavior of the newly joined nodes is unknown, these nodes will not be involved in a large number of routes until they build up good trust by behaving well in the routes they participate in. The nodes with good past behavior are more trusted than those with unknown behavior. The newly joined nodes will be selected when the source and destination nodes have limited options, or when they report high energy capability, so they will build up their trust values slowly. This coincides with the meaning of trust, i.e., a node cannot be trusted before showing a clear trustful behavior.

Packets can be dropped due to non-malicious reasons such as temporary bad channel or network congestion. The effect of this on the nodes' trust values can be neutralized by computing the trust values based on the nodes' behavior in a number of sessions. It is not reasonable to assume that some nodes will suffer from the non-malicious packet drop more than others over a number of sessions.

For trust boost attack, the attackers attempt to falsely boost their trust values to increase their chance to participate in routes and thus earn more credits. For false accusation attack, the attackers aim to falsely accuse victim nodes of breaking the routes to degrade their trust values. Thwarting trust boost and false accusation attacks launched by large-scale colluding nodes has been studied in Web-related trust systems [17]. It is shown that false accusation attacks can be avoided by concealing the real identities of the intermediate nodes. In [18], statistical filtering algorithms have been proposed to filter out the false accusations by excluding or giving low weight to the presumed unfair ratings based on analyzing the rating values. The assumption is that unfair ratings can be recognized from their statistical properties. Since eBay [19] charges a fee for each transaction, trust boost attack would be expensive. Fortunately, these techniques can be used with our trust system more effectively because it is difficult to obtain multiple identities comparing to Web applications that provide free access to users via simple registration process.

Singular attackers cannot launch trust boost attacks in E-STAR. In order to make fabricating sessions by colluding nodes to boost their trust expensive, clearance fee can be imposed to clear the payment of a session. For false accusation attack, the attacker has to neighbor the victim node and break the route to let TP accuse its neighbor. First, neighboring the victim node may not be easy due to the nodes' mobility. Second, the attacker is also accused of breaking the route, which may discourage launching 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 values. Finally, falsely accusing a node of breaking a route does not guarantee that this accusation will be effective because the node can improve its trust values from participating in other routes.

If a malicious node can create several fake identities, the trust management system suffers from the Sybil attack [20] and the attackers can launch effective attacks. For the newcomer attack, the attackers can change their identities when their trust values degrade if they can easily register as new users [21]. Hence, the attackers can easily remove their bad history by registering as new users. The newcomer attack can also improve the effectiveness of false accusation attacks. In order to make the newcomer and Sybil attacks expensive and ineffective, TP can impose some fees for registering a new user or changing an identity. Moreover, the nodes' initial trust values should not be high when they first join the network, and the old nodes that have spent long time in the network should be trusted more.

For on-off attack, the attackers take advantage of the fact that bad behavior can be compensated with good behavior. They first behave well until gaining high trust value. Then, they alternate their behaviors between misbehaving and well-behaving with keeping their trust values high [22]. This attack is possible when the trust system increases/decreases the trust values with the same amount when good/bad behaviors are observed. Our trust system can thwart this attack because the trust values are computed based on the ratio of the good observations in the last ω sessions.

To secure the routing protocol, E-STAR can satisfy the following requirements: (1) valid routing packets (RREQ and RREP) cannot be fabricated; (2) invalid routing packets cannot be propagated in the network; (3) valid routing packets cannot be altered in transit without detection, except according to the normal functionality of the protocol; (4) routing loops cannot be formed by malicious action; (5) stale routing packets are not accepted by the nodes or propagated in the network; and (6) unauthorized nodes' packets are not accepted by the authorized nodes.

For unauthorized participation, the nodes that are not members of the network cannot act as source, intermediate, or destination nodes. For spoofed RREQ, since only the source node can sign with its own private key, the attackers cannot spoof other nodes to establish routes under their identities. RREQ packets will not be accepted by the nodes if their signatures are invalid. This can thwart many external attacks launched by unauthorized nodes, such as RREQ flooding attack. For alteration of routing packets, an adversary may attempt to manipulate the RREQ packets, e.g., to provide wrong data to increase its chance to be involved in routes to earn more credits or break the routes. Since routing packets are signed, manipulating the routing information, the trust/energy requirements, and the maximum number of hops of an in-transit packet would be detected by the intermediate nodes along the route, and the altered packet would be subsequently discarded. This attack cannot prevent establishing a route because flooding the RREQ packets enables the destination nodes to receive multiple routes. Also, dropping the RREP packets cannot prevent establishing the route because the destination node can send a new RREP packet for a different route if the data transmission does not start on a route.

In route establishment, the nodes that report incorrect trust values can be detected because the trust values are signed by TP. The nodes cannot manipulate their trust values because they cannot forge the TP's signature. For destination node impersonation attack, the attacker attempts to send RREP packet to let the source node believe that it communicates with the destination node. This is infeasible in E-STAR because the destination nodes sign the RREP packets to ensure that only the destination node can respond to the RREQ packet. For the RREQ flooding attack launched by internal attackers, since the source nodes sign the RREQ packets, the attackers can be identified in an undeniable way. The network nodes can ignore a node's packets when it sends a large number of RREQ packets in a short time. For route lengthening attack, in E-STAR, elongating a route by inserting non-existing nodes to the RREQ packet requires signing the packet with the private keys of these nodes. It also decreases the chance of selecting the route because the route reliability decreases, as discussed in Table 2. More security analyses are given in Appendix A.

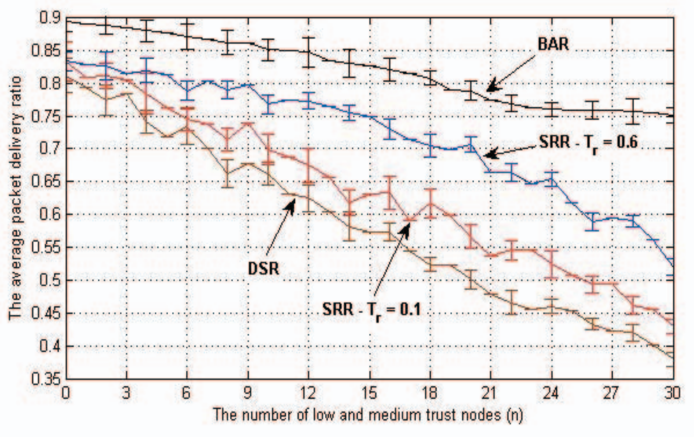


Fig. 7: E-STAR can improve the packet delivery ratio due to selecting good intermediate nodes.

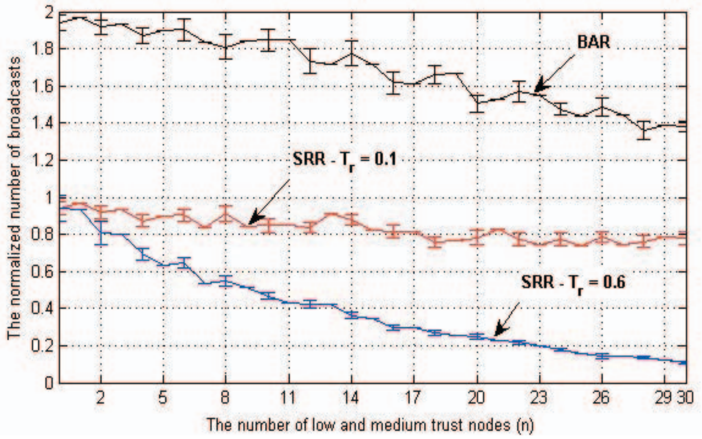


Fig. 8: SRR generates fewer RREQ broadcasts because the nodes that cannot satisfy the source node's requirements do not broadcast the packets.

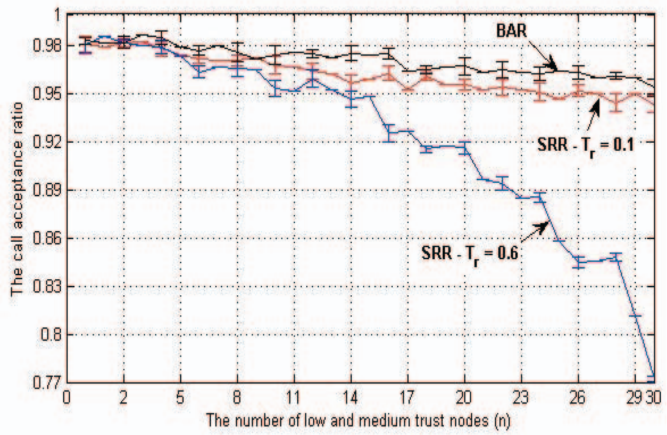


Fig. 9: Routes are not established if the source node's trust requirement is not well selected in SRR.

6. Performance Evaluations

We simulate a heterogeneous multihop wireless network by randomly deploying 55 nodes in an area of 1000 m × 1000 m. n is the number of nodes having low and medium trust values. The number of nodes having high trust values is 55-n and their trust values are uniformly distributed in [0.8, 1). The number of nodes having low trust values is $\lfloor 0.67 \times n \rfloor$ and their trust values are uniformly distributed in [0, 0.3). The number of nodes having medium trust value is $\lceil 0.33 \times n \rceil$ and their trust values are uniformly distributed in [0.3, 0.8). A node with a trust value of 0.6 breaks routes with the probability of 1 - 0.6 = 0.4. By this way, the trust values can be used to simulate the variety in the nodes' lack of resources and malicious actions.

The radio transmission range is 125 m, and all the nodes start the simulation with the same initial energy that is sufficient for relaying 100 messages. The data packet size is 512K bytes, $H_{\max}$ is eight. In SRR protocol, the trust requirement (T_r) is 0.1 and 0.6, and the energy requirement is the number of messages the source node needs to send. A digital signature generation time of 8.5 ms and verification time of 0.5 ms and hashing time of 29 µs were simulated by adding them to the processing time of the packets. These values were obtained by measuring the signing and verifying times of the 1024-bit RSA digital signature algorithm and the hashing time of SHA-1 hash function by using the Crypto++5 library [23] and 750 MHz processor. The size of SHA-1 hash value is 20 bytes and the RSA signature size is 128 bytes. The energy consumptions of the RSA signing and verifying operations and SHA-1 hashing operations are measured in [24]. The signing and verifying operations consume 546.5 mJ and 15.97 mJ, respectively. The SHA-1 hash function consumes 0.76 mJ per byte.

The given results are averaged over 100 simulation runs and presented with 95% confidence interval. In each run, 30 communication sessions with randomly chosen source and destination pairs are established. The route is re-established if it is broken before sending 12 messages. The constant bit rate (CBR) traffic model is employed for all the connections. To simulate the node mobility, we use the random waypoint mobility model [25]. Each node travels to a randomly selected location at a configured speed and then pauses for some time, before choosing another random location and repeating the same steps. The pause time is 2s and the nodes' speed is uniformly distributed in [0, 1] m/s.

The packet delivery ratio (PDR) is the total number of packets received by the destination nodes to the total number of packets sent by the source nodes. From Fig. 7, E-STAR outperforms the dynamic source routing (DSR) protocol [16] in the packet delivery ratio. DSR enables the low-trusted nodes and the nodes having low energy to repeatedly participate in routes and break them because it randomly chooses the intermediate nodes. Conversely, E-STAR establishes more stable routes by selecting reliable intermediate nodes and therefore it delivers packets more successfully. Although DSR reestablishes a route each time it is broken, the new route still includes low-trusted nodes with a high probability, and thus fails again.

When we compare our protocols with DSR, we actually compare between two strategies: informed routing decisions and randomly selecting intermediate nodes. DSR randomly selects intermediate node, but our protocols make informed routing decisions by selecting the nodes that behaved well in the past and have enough energy. Therefore, improvement techniques proposed for DSR such as route recovery schemes can also be used with our protocols.

We can see that the packet delivery ratio of DSR significantly degrades as the number of low-trust nodes increases due to involving these nodes in routes more frequently. For SRR, the increase of T_r can increase the packet delivery ratio due to selecting more trusted nodes, but as we will discuss later the probability of establishing routes decreases. At T_r = 0.1, the increase of n decreases PDR because more low-trust nodes participate in routes. However, the reduction in PDR at T_r = 0.6 is mainly due to the messages the source nodes could not send because they did not find routes with this trust requirement. In BAR, the increase of the low-trust nodes has little effect on PDR because it can avoid these nodes and select nodes with good trust values and sufficient energy. Moreover, we can see that at n = 0 and at few low-trust nodes, the packet delivery ratios in SRR and BAR are higher than that of DSR, because they select the nodes having sufficient energy.

Fig. 8 shows the number of RREQ broadcast transmissions in E-STAR to this of the DSR at different values of *n*. The Wait Period at each node is 20ms in BAR. We can see that the normalized number of broadcasts in SRR is always less than one because the nodes that cannot satisfy the energy or trust requirements do not broadcast the RREQ packets. At $T_r = 0.6$, the number of broadcasts is less because more nodes cannot satisfy the trust requirements and thus do not broadcast RREQ packets. For BAR, the normalized number of broadcasts is always above one because a node may broadcast a RREQ packet more than once, but in DSR each node broadcasts a RREQ packet at most once.

In Fig. 9, the call acceptance ratio is the ratio of times a route is established after sending a RREQ packet. We can see that the call acceptance ratio in BAR nearly does not depend on *n*. However, the increase of *n* decreases the call acceptance ratio in SRR because more nodes cannot satisfy the trust requirement, and thus more routes cannot be established. At $T_r = 0.6$, the call acceptance ratio significantly decreases with the increase of *n* because more nodes cannot satisfy the trust requirement.

In Fig. 10, the normalized route lifetime is the average route lifetime in E-STAR to that of DSR. The route lifetime is the number of packets sent in one route before it is broken. Route lifetime is a good measure for route stability. Since the normalized route lifetime is always more than one, E-STAR can establish more stable routes comparing to DSR. At $n > 12$, SRR with $T_r = 0.6$ may establish more stable routes but as indicated in Fig. 9, the likelihood of establishing a route decreases as *n* increases. More simulation results are given in Appendix A.

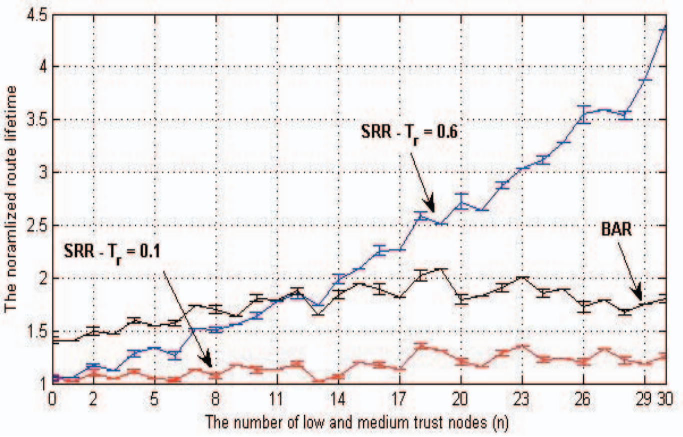


Fig. 10: The route lifetime in E-STAR is more than that in DSR because of establishing more stable routes.

7. Conclusion

We have proposed E-STAR that uses payment/trust systems with trust-based and energy-aware routing protocol to establish stable/reliable routes in HMWNs. E-STAR stimulates the nodes not only to relay others' packets but also to maintain the route stability. It also punishes the nodes that report incorrect energy capability by decreasing their chance to be selected by the routing protocol. We have proposed SRR and BAR routing protocols and evaluated them in terms of overhead and route stability. Our protocols can make informed routing decisions by considering multiple factors, including the route length, the route reliability based on the nodes' past behavior, and the route lifetime based on the nodes' energy capability. SRR establishes routes that can meet source nodes' trust/energy requirements. It is useful in establishing routes that avoid the low-trust nodes, e.g., malicious nodes, with low overhead. For BAR, destination nodes establish the most reliable routes but with more overhead comparing to SRR. The analytical results have demonstrated that E-STAR can secure the payment and trust calculation without false accusations. Moreover, the simulation results have demonstrated that E-STAR can improve the packet delivery ratio due to establishing stable routes.

References

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

[2] C. Chou, D. Wei, C. Kuo, and K. Naik, "An efficient anonymous communication protocol for peer-to-peer applications over mobile ad-hoc networks", IEEE Journal on Selected Areas in Communications, vol. 25, no. 1, January 2007.

[3] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks", Proc. of IEEE/ACM MobiCom'00, pp. 255–265, Boston, MA, August 6-11, 2000.

[4] X. Li, Z. Li, M. Stojmenovic, V. Narasimhan, and A. Nayak, "Autoregressive trust management in wireless ad hoe networks", Ad hoc & Sensor Wireless Networks, vol. 16 Issue 1-3, pp. 229-242 2012

[5] G. Indirania, K. Selvakumara, "A swarm-based efficient distributed intrusion detection system for mobile ad hoc networks (MANET)", International Journal of Parallel, Emergent and Distributed Systems, 2013.

[6] H. Li and M. Singhal, "Trust management in distributed systems", IEEE Computers, vol. 40, no. 2, pp. 45-53, February 2007.

[7] K. Liu, J. Deng, and K. Balakrishnan "An acknowledgement-based approach for the detection of routing misbehavior in MANETs", IEEE Transaction on Mobile Computing, vol. 6, no. 5, pp 536–550, May 2007.

[8] S. Zhong, J. Chen, and R. Yang, " Sprite: a simple, cheat-proof, credit based system for mobile ad-hoc networks", Proc. of IEEE INFOCOM'03, vol. 3, pp. 1987-1997, San Francisco, CA, March 30- April 3, 2003.

[9] M. 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.

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

[11]M. Mahmoud and X. Shen, "An integrated stimulation and punishment mechanism for thwarting packet drop in multihop wireless networks", IEEE Transactions on Vehicular Technology, vol. 60, no. 8, pp. 3947-3962, 2011.

[12]G. Theodorakopoulos and J. S. Baras, "On trust models and trust evaluation metrics for ad hoc networks", IEEE Journal on Selected Areas in Communications, vol. 24, no. 2, pp. 318-328, February 2006.

[13]P. Velloso, R. Laufer, D. Cunha, O. Duarte, and G. Pujolle, "Trust management in mobile ad hoc networks using a scalable maturity-based model", IEEE Transactions on Network and Service Management, vol. 7, no. 3, pp. 172–185, September 2010.

[14]S. Lindsay, Y. Wei, H. Zhu and K. Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks", IEEE Journal on Selected Areas in Communications, vol. 24, no. 2, pp. 305–317, 2006.

[15]M. Yu and K. Leung, "A trustworthiness-based QoS routing protocol for wireless ad hoc networks", IEEE Transactions on Wireless Communications, vol. 8, no. 4, pp. 1888–1898, April 2009.

[16]D. Johnson, D. Maltz, and J. Broch, "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks", In C. Perkins, editor, Ad Hoc Networking, chapter 5, pp. 139-172. Addison-Wesley, 2001.

[17]A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision", Decision Support Systems, vol. 43, no. 2, pp. 618-644, 2007.

[18] A. Withby, A. Jøsang, and J. Indulska, "Filtering out unfair ratings in Bayesian reputation systems", The Icfain Journal of Management Research, vol. 4, no. 2, pp. 48-64, 2005.

[19] P. Resnick and R. Zeckhauser, "Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system", Proc. of NBER workshop on empirical studies of electronic commerce, 2000.

[20] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: Analysis and defenses", Proc. of the third International Symposium on Information Processing in Sensor Networks (IPSN), 2004.

[21] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara, "Reputation systems", Communications of the ACM, vol. 43, no. 12, pp. 45–48, 2000.

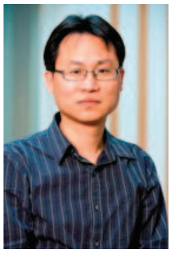
[22] N. Bhalaji and A. Shanmugam, "Reliable routing against selective packet drop attack in DSR based MANET", Journal of Software, vol. 4, no. 6, pp. 536-543, August 2009.

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

[24] 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, Mar./Apr. 2006.

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

Mohamed M. E. A. Mahmoud received the Ph.D. degree (April 2011) in electrical and computer engineering from University of Waterloo, Waterloo, Ontario, Canada. He is currently a Postdoctoral Fellow with the Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo. His research interests include wireless network security, privacy-preserving schemes, anonymous and secure routing protocols, trust and reputation systems, cooperation incentive mechanisms, and cryptography.



Xiaodong Lin (S'07-M'09-SM'12) received the Ph.D. degree in information engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1998 and the Ph.D. degree (with Outstanding Achievement in Graduate Studies Award) in electrical and computer engineering from the University of Waterloo, Waterloo, Ontario, Canada, in 2008. He is currently an assistant professor of information security with the Faculty of Business and Information Technology, University of Ontario Institute of Technology (UOIT), Oshawa, Ontario, Canada. His research interests include wireless network security, applied cryptography, computer forensics, and software security. Dr. Lin was the recipient of Best Paper Awards of the IEEE International Conference on Computer Communications and Networks (ICCCN 2009) and the IEEE International Conference on Communications (ICC 2007) - Computer and Communications Security Symposium.

Xuemin (Sherman) Shen received the BSc degree from Dalian Maritime University, China, in 1982, and the MSc and PhD degrees from Rutgers University, Camden, New Jersey, in 1987 and 1990, respectively, all in electrical engineering. He is currently a professor and the university research chair with the Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Ontario, Canada. He is the author or coauthor of three books and more than 400 papers and book chapters on wireless communications and networks, control, and filtering. He serves as the editor-in-chief for peer-to-peer networking and applications and an associate editor for Computer Networks, ACM/Wireless Networks, and Wireless Communications and Mobile Computing. He has also served as a guest editor for ACM Mobile Networks and Applications. His research focuses on mobility and resource management in interconnected wireless/wired networks, ultra wideband wireless communications networks, wireless network security, wireless body area networks, and vehicular ad hoc and sensor networks. He is a registered professional engineer in the Province of Ontario and a distinguished lecturer of the IEEE Communications Society. He received the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premier's Research Excellence Award in 2003 from the Province of Ontario, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. He served as the tutorial chair for the 2008 IEEE International Conference on Communications, the technical program committee chair for the 2007 IEEE Global Telecommunications Conference, the general cochair for the 2007 International Conference in Communications and Networking in China, and the 2006 International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, and the founding chair for IEEE Communications Society Technical Committee on P2P Communications and Networking. He also serves as a founding area editor for the IEEE Transactions on Wireless Communications and an associate editor for the IEEE Transactions on Vehicular Technology and the KICS/IEEE Journal of Communications and Networks. He has also served as a guest editor for the IEEE Journal on Selected Areas in Communications, IEEE Wireless Communications, and the IEEE Communications Magazine. He is a fellow of the IEEE.