Research Article

Swarm Intelligent Power-Aware Detection of Unauthorized and Compromised Nodes in MANETs

Shabana Mehfuz¹ and M. N. Doja²

¹ Department of Electrical Engineering, Faculty of Engineering and Technology, Jamia Millia Islamia, New Delhi 110025, India

² Department of Computer Engineering, Faculty of Engineering and Technology, Jamia Millia Islamia, New Delhi 110025, India

Correspondence should be addressed to Shabana Mehfuz, mehfuz.shabana@yahoo.com

Received 2 July 2007; Revised 14 October 2007; Accepted 15 November 2007

Recommended by Terence Soule

Characteristics of mobile ad hoc networks (MANETs) such as lack of central coordination, mobility of hosts, and limited availability of resources make quality of service (QoS) provisioning very challenging. Limited resource availability such as battery power and insecure medium is one of the major QoS issues to be dealt with. In this paper, we have proposed a new secure power-aware ant routing algorithm (SPA-ARA) for mobile ad hoc networks that is inspired from ant colony optimization (ACO) algorithms which are a swarm intelligent technique. In this algorithm, we have introduced a new metric, *next-hop availability*, which is a combination of two metrics. It maximizes path availability and minimizes travel time of packets, and therefore it offers a good balance between selection of fast paths and a better use of network resources. The protocol also incorporates a trust model which helps in detection of unauthorized and compromised nodes in MANETs.

Copyright © 2008 S. Mehfuz and M. N. Doja. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

MANETs are zero-configuration, self-organizing, and highly dynamic networks formed by a set of mobile hosts connected through wireless links. These networks can be formed on the fly, without requiring any fixed infrastructure. As these are infrastructureless networks, each node should act also as a router. These characteristics of MANETs such as lack of central coordination, mobility of hosts, limited availability of resources, and insecure medium make quality of service (QoS) provisioning very challenging [1]. QoS is usually defined as a set of service requirements that need to be met by the network while transporting a packet stream from a source to its destination(s) [2]. Providing routes that are stable based on route statistics could potentially reduce the communication disruption time too. This can be achieved by incorporating quality of service (QoS) metrics such as battery life and security measures into the routing decisions as opposed to choosing a shortest path. Efficient resource management mechanisms are required for optimal utilization of this scarce resource, that is, battery power. Also, MANETs are susceptible to attacks such as eavesdropping, spoofing, denial of service, message distortion, and impersonation. Without so-

phisticated security mechanisms, it is very difficult to provide secure communication guarantees.

The “swarm intelligence” paradigm is an approach to routing in distributed networks of cooperative agents, inspired by the process by which swarms of ants converge to the optimal route to a food source by progressively reinforcing the successful paths using pheromone secretions. ACO and S-ACO algorithms [3] have originated from the above-mentioned paradigm and are desirable in the context of ad hoc networks.

In this paper, we have proposed a new secure power-aware ant routing algorithm (SPA-ARA) for MANETs. It solves the routing problem and manages the network resources from the point of view of achieving fair resources usage across the network node. At the same time, it also detects the unauthorized and compromised nodes in the network. It combines the hard security mechanisms with a trust model and provides security to the network against various internal and external attacks.

The remainder of the paper is organized as follows. Section 2 briefly discusses some special characteristics of MANETs’ routing problem. In Section 3, we describe ACO principles and ant routing algorithms for MANETs. Section 4

discusses the power-aware routing protocols. Security issues are discussed in Section 5. Section 6 describes the proposed algorithm. Section 7 presents simulation. In Section 8, brief conclusions are given.

2. ROUTING ISSUES IN MANETs

In mobile ad hoc networks, routing protocols are challenged with establishing and maintaining multihop routes in the face of mobility, bandwidth limitation, and power constraints. Many routing algorithms have been designed for the mobile ad hoc networks. These algorithms can be classified into two groups: table-driven routing (such as DSDV, CGSR, GSR, FSR, HSR, WRP, etc.) [4, 5], and source-initiated on-demand routing (such as AODV, DSR, TORA, ABR, SSR, etc.) [4, 6]. Both groups need large uncontrolled overheads to solve the routing problem. The number of routing packets increases dramatically as the network size increases. This large routing overhead affects the scalability of the network and the network performance since it uses a significant part of the wireless bandwidth and the node's limited energy and processing power. In addition, most of these algorithms are optimizing only one parameter, which in most cases is the number of hops. In these algorithms, little work has been done on the energy consumed during the routing. What they concerned mostly is to find the route to make the messages reach the destination fast. In such process, some nodes may be overused and will use out their energy quickly. So, the network may be partitioned.

These protocols are highly optimized to spread new routing information quickly as condition changes, requiring more rapid and often more frequent routing protocol interaction between nodes that is typical in a traditional network. Thus, expensive and cumbersome security mechanisms are not embedded into ad hoc routing protocols since they can delay or prevent such exchanges of routing information, leading to reduced routing effectiveness, and may consume excessive network or node resources, which may lead to denial-of-service attacks through the routing protocol [7]. Sanzgiri et al. [8] have detailed security threats against ad hoc routing protocols, specifically examining AODV and DSR.

3. ANT COLONY OPTIMIZATION

The ACO algorithms have been inspired by the behavior of the real ant colony. The algorithm can find the optimum solution by generating artificial ants. As the real ants search their environment for food, the artificial ants search the solution space. The probabilistic movement of ants in the system allows the ants to explore new paths and to re-explore the old visited paths. The strength of the pheromone deposit directs the artificial ants toward the best paths, and the pheromone evaporation allows the system to forget old information and avoid quick convergence to suboptimal solutions. A number of proofs for the convergence to the optimum path of the ACO can be found in [9, 10]. The ant's behavior as well as pheromone evaporation has been implemented in an algorithm called Simple-ACO (S-ACO) [3]. S-ACO is a didactic tool to explain the basic mechanism underlying ACO algo-

FIGURE 1: Connected graph $G = (N, A)$; N = number of nodes; A = number of arcs. Arrow shows the path where ant travels from source to destination.

rithms. Some of the important concepts in S-ACO have been explained as follows.

Probabilistic forward ants and solution construction

S-ACO ants can be thought of as having two working modes: forward and backward. They are in forward mode when they are moving from nest toward the food, and they are in backward mode when they are moving from the food back to their nest. Once an ant in forward mode reaches its destination, it switches to backward mode and starts its travel back to source. In S-ACO forward mode, ants build a solution by choosing probabilistically the next node to move to among those in the neighborhood of the graph node at which they are located. Given a graph $G = (N, A)$, where N is the set of $n = |N|$ nodes and A is the set of undirected arcs connecting them, two nodes $i, j \in N$ are neighbors if there exists an arc $(i, j) \in A$ (see Figure 1). The probabilistic choice is biased by pheromone trails previously deposited on the graph by other ants. Forward ants do not deposit any pheromone trails previously deposited on the graph by other ants. Forward ants do not deposit any pheromone while moving. This, together with deterministic backward moves, helps in avoiding the formation of loops.

Deterministic backward ants and pheromone update

The use of an explicit memory allows an ant to retrace the path it has followed while searching the destination node. Moreover, S-ACO ants improve the system performance by implementing loop elimination. In practice, before starting to move backward on the path they memorized while searching the destination node (i.e., the forward path), S-ACO ants eliminate any loops from it. While moving backward, S-ACO ants leave pheromone on the arcs they traverse.

Pheromone updates based on solution quality

In S-ACO, the ants memorize the nodes they visited during the forward path, as well as the cost of the arcs traversed if the graph is weighted. They can therefore evaluate the cost of the solutions they generate and use this evaluation to modulate the amount of pheromone they deposit while being in backward mode. Making pheromone update, a function of

the generated solution quality can help in directing future ants more strongly toward better solutions. In fact, by letting ants deposit a higher amount of pheromone on short paths, the ant's searching is more quickly biased toward the best solutions.

Pheromone evaporation

In real ant colonies, pheromone intensity decreases over time because of evaporation. In S-ACO, evaporation is stimulated by applying an appropriately defined pheromone evaporation rule. The artificial pheromone decay can be set to a constant rate. Pheromone evaporation reduces the influence of the pheromones deposited in the early stages of the search, when artificial ants can build poor quality solutions.

3.1. Why ant colony optimization algorithm suits to MANETs

The simple ant colony optimization metaheuristic shown illustrates different reasons why this kind of algorithms could perform well in mobile multihop ad hoc networks. We will discuss various reasons by considering important properties of mobile ad hoc networks.

(1) Dynamic topology

This property is responsible for the bad performance of several routing algorithms in mobile multihop ad hoc networks. The ant colony optimization metaheuristic is based on agent systems and works with individual ants. This allows for a high adaptation to the current topology of the network.

(2) Local work

In contrast to other routing approaches, the ant colony optimization metaheuristic is based only on local information; that is, no routing tables or other information blocks have to be transmitted to neighbors or to all nodes of the network.

(3) Link quality

It is possible to integrate the connection/link quality into the computation of the pheromone concentration, especially into the evaporation process. This will improve the decision process with respect to the link quality. It is here important to notice that the approach has to be modified so that nodes can also manipulate the pheromone concentration independent of the ants, that is, data packets. For this, a node has to monitor the link quality.

(4) Support for multipath

Each node has a routing table with entries for all its neighbors, which contains also the pheromone concentration. The decision rule, to select the next node, is based on the pheromone concentration on the current node, which is provided for each possible link. Thus, the approach supports multipath routing.

3.2. Ant routing algorithms for MANETs

Early work on MANETs consisted primarily of applying the traditional approaches to routing in wired networks, such as distance vector or link state algorithms, and to the more volatile network environments experienced in ad hoc networks. While many optimizations to these algorithms exist, each of them is primarily concerned with finding the minimum hop route from source to destination.

The ant routing algorithms use artificial ant colony in the optimization as S-ACO or S-ACO metaheuristic. The ant finds, maintains, and optimizes the best paths. The ant-colony-based routing algorithm (ARA) [11] employs a forward/backward ant technique, which is very close to blind flooding and is still used for path discovery. This algorithm works in an on-demand way, with ants setting up multiple paths between source and destination at the start of a data session. ARA is made up of two phases which are route discovery and route maintenance. During route discovery, a forwarding ANT (FANT) is propagated throughout the network (similar to an RREQ). At each hop, each node calculates a pheromone value depending on how many hops the FANT has taken to reach them. The nodes then forward the FANT to their neighbors. Once the destination is reached, it creates a backward ANT (BANT) and returns it to the source. When the source receives the BANT from the destination node, a path is determined and data packet dissemination begins. To maintain each route, each time a data packet travels between intermediate nodes, the pheromone value is increased. Otherwise, the pheromone value is decreased over time until it expires. To repair a broken link, the nodes firstly check their routing table; if no route is found, they inform their neighbors for an alternate route. If the neighbors do have a route, they inform their neighbors by backtracking. If the source node is reached and no route is found, a new route discovery process is initiated. The advantage of this strategy is that the size of each FANT and BANT is small, which means that the amount of overhead per control packet introduced in the network is minimized. However, the route discovery process is based on flooding, which means that the protocol may have scalability problems as the number of nodes and flows in the network grows. Probabilistic emergent routing algorithm (PERA) [12] works in an on-demand way, with ants being broadcast toward the destination (they do not follow pheromone) at the start of a data session. Multiple paths are set up, but only the one with the highest pheromone value is used by data with the other paths being available for backup. Ants-based routing in large-scale mobile ad hoc networks [13] is another ACO algorithm in which all nodes should be aware of their location and other nodes locations. The algorithm protocols work in two layers: logical (upper) layer and lower layer. The algorithm divides the network into regions and considers each region as logical router. Each logical router's table is distributed on the region's nodes satisfying some properties such as redundancy. The ant routing algorithm is working in the logical layer. This proposed algorithm is complex because it needs means to determine the locations of all the nodes. In addition, in each layer, forwarding protocol is needed. Also, other ACO routing algorithms

TABLE 1: Summary of traditional and ACO methods.

Approach	Protocols
Traditional	<i>Table-driven</i> : destination-sequenced distance vector (DSDV), cluster-head gateway switch routing (CGSR), wireless routing protocol (WRP), global state routing (GSR), optimized link state routing (OLSR), fisheye state routing (FSR), landmark routing (LMAR)
	<i>On-demand</i> : ad hoc on-demand distance vector (AODV), dynamic source routing (DSR), temporally ordered routing algorithm (TORA), associativity-based routing (ABR), signal stability routing (SSR)
ACO	Ant-colony-based routing algorithm (ARA), probabilistic emergent routing algorithm (PERA), ants-based routing in large-scale mobile ad hoc networks

have been proposed for MANETs in [14, 15]. Table 1 shows a summary of traditional and ACO methods. In general, however, most of all these algorithms move quite far away from the original ACO routing ideas trying to obtain the efficiency needed in MANETs, and many of them are not very different from single-path on-demand algorithms. None of ACO algorithms for MANET, mentioned above, has taken up power-aware routing neither incorporated security measures in routing, which are the two most major QoS issues in MANETs.

4. POWER-AWARE ROUTING IN MANETs

As already discussed in Section 2, the traditional routing algorithms (e.g., DSDV, CGSR, GSR, FSR, HSR, WRP, etc. in [4, 5] and AODV, DSR, TORA, ABR, SSR, etc. in [4, 6]) lack power-aware routing. The ACO algorithms which have been proposed for MANETs [11–15] mostly employ a number of hops as the only optimization parameter, and they are not concerned with the depleting battery power which is a very major issue involved in MANET routing.

Two classes of power-saving strategies for ad hoc networks have been identified: local strategies, which typically operate on small time scales, and global strategies, which operate on longer time scales [16]. Local strategies employ the transmission power control approach, which reduces the active communication energy by adjusting each node's radio power just enough to reach the receiving node but not more than that. Another approach used by local strategy that saves the inactive energy is sleep/power-down mode, which switches off the node when there is no data to transmit or receive. Global strategies mainly employ load distribution approach where the primary focus is to balance the energy usage among the nodes to maximize the network lifetime by avoiding overutilized nodes when selecting a routing path. An overview of some recent power-aware protocols has been given below.

Power-efficient routing protocols include the work of Singh et al. [17] who investigated the use of power-aware metrics in the calculation of shortest paths. These metrics describe the power required for transmitting and receiving a packet on a link, so as to minimize the end-to-end power requirements for routing. This proposal did not take into account the remaining energy in the nodes, and it can result in a severe drain of energy in the batteries of the nodes on the least-cost route.

Other proposals overcame this problem by using battery lifetime information. Toh [18] proposed a new metric, which calculates the sum of the inverse of the remaining battery capacities of the nodes on the path. In addition, Toh proposed the min-max algorithm to maintain a fair use of resources by avoiding the use of nodes with the least remaining battery capacity in the network. Li et al. [19] proposed an algorithm (denoted by max-min zPmin) that computes the paths with minimal energy consumption while maximizing the minimal residual power of the network.

Power-aware source routing (PSR) [20] is similar to DSR, but the destination calculates the link cost based on the remaining battery capacity and transmission power of the nodes. The drawback of this approach is that the destination needs to wait some time after the arrival of the first route request, so as to receive more than one possible route, and then selects the one with the minimum cost.

5. SECURITY ISSUE IN MANETs

As already discussed earlier, security in MANET is one of the important QoS issues. It is an essential component for basic network functions like packet forwarding and routing. Network operations can be easily jeopardized if countermeasures are not embedded into basic network functions at the early stages of their design. The threats to the basic functions of any network such as routing have the most immediate effects on the QoS. Unlike networks using dedicated nodes to support basic functions like packet forwarding, routing, and network management, in ad hoc networks those functions are carried out by all available nodes. This very difference is at the core of the security problems that are specific to ad hoc networks. As opposed to dedicated nodes of a classical network, the nodes of an ad hoc network cannot be trusted for the correct execution of critical network functions.

When more security features are introduced into the network, in parallel, along with the enhanced security strength, there are the ever increasing computation, communication, and management overheads too. Consequently, network performance of the security solutions, in terms of scalability, service availability, robustness, and so on, becomes an important concern in a resource-constrained ad hoc network. While many contemporary proposals focus on the security vigor of their solutions from the cryptographic standpoint, they leave the network performance aspect largely unaddressed. In fact, both dimensions of security, that is, strength and network performance, are equally important, and achieving a good tradeoff between two extremes is one fundamental challenge in security design for MANETs.

TABLE 2: Security attacks classification.

Passive attacks	Eavesdropping, traffic analysis, monitoring
Active attacks	Jamming, spoofing, modification, replaying, DoS

TABLE 3: Attacks on protocol stack.

Layer	Attacks
Application layer	Repudiation, data corruption
Transport layer	Session hijacking, SYN flooding
Network layer	Wormhole, black hole, Byzantine, flooding, resource consumption, location disclosure attacks
Data link layer	Traffic analysis, monitoring, disruption MAC (802.11), WEP weakness
Physical layer	Jamming, interceptions, eavesdropping
Multilayer attacks	DoS, impersonation, replay, man-in-the-middle

5.1. Attacks in MANETs

The attacks in MANET can roughly be classified into two major categories, namely, passive attacks and active attacks, according to [21]. A passive attack obtains data exchanged in the network without disrupting the operation of the communications, while an active attack involves information interruption, modification, or fabrication, thereby disrupting the normal functionality of a MANET. Table 2 shows the above-mentioned classification of security attacks against MANET.

The attacks can also be classified into two external attacks and internal attacks, according to the domain of the attacks. Some papers call them outsider and insider attacks [22]. External attacks are carried out by nodes that do not belong to the domain of the network. Internal attacks are from compromised nodes, which are actually part of the network. Internal attacks are more severe when compared with outside attacks since the insider knows valuable and secret information and possesses privileged access rights. Attacks can also be classified according to network protocol stacks. Table 3 shows an example of a classification of security attacks based on protocol stack.

Nodes that perform active attacks with the aim of damaging other nodes by causing network outage are considered to be malicious (also referred to as compromised), while nodes that just drop the packets they receive with the aim of saving battery life for their own communications are considered to be selfish [23, 24]. A selfish node affects the normal operation of the network by not participating in the routing protocols or by not forwarding packets. In addition, a compromised node may use the routing protocol to advertise itself as having the shortest path to the node whose packets it wants to intercept as in the so-called black hole attack [25, 26].

Compromised nodes can interrupt the correct functioning of a routing protocol by modifying routing information and by fabricating false routing information. Recent research studies have also brought up a new type of attack that goes under the name of wormhole attack [27, 28]. In the latter, two compromised nodes create a tunnel (or wormhole) that

is linked through a private connection, and thus they bypass the network. This allows a node to short-circuit the normal flow of routing messages creating a virtual vertex cut in the network that is controlled by the two attackers [29, 30].

On the other hand, selfish nodes can severely degrade network performance and eventually partition the network by simply not participating in the network operation. Compromised nodes can easily perform integrity attacks by altering protocol fields in order to subvert traffic, denying communication to legitimate nodes and compromising the integrity of routing computations in general. Spoofing is a special case of integrity attacks whereby a compromised node impersonates a legitimate one due to the lack of authentication in the current ad hoc routing protocols [31, 32]. The main result of a spoofing attack is the misrepresentation of the network topology that may cause network loops or partitioning. Lack of integrity and authentication in routing protocols creates fabrication attacks [33–35] that result in erroneous and bogus routing messages.

Denial of service (DoS) is another type of attack, in which the attacker injects a large amount of junk packets into the network. These packets overspend a significant portion of network resources, and introduce wireless channel contention and network contention in ad hoc networks [36, 37]. In addition, the routing table overflow attack, where an attacker attempts to create routes to nonexistent nodes, and the sleep deprivation attack, where an attacker tries to consume the batteries of a node, are two other types of DoS attacks [38].

5.2. Intrusion detection in MANETs

When a set of actions that attempt to compromise the integrity, confidentiality, or availability of a mobile node takes place, intrusion prevention techniques, such as encryption and authentication, are usually the first line of defense. However, intrusion prevention alone is not sufficient when systems become more complex. There are two main approaches in current securing ad hoc environments. The first is intrusion prevention measures, such as authentication and encryption. The second is intrusion detection and response approach.

5.2.1. Intrusion prevention in MANET

The secure efficient ad hoc distance (SEAD) vector routing protocol proposed in [7] employs the use of hash chains to authenticate hop counts and sequence numbers. SEAD is based on the design of the proactive ad hoc routing protocol DSDV. The SEAD protocol has the utilization of a clock synchronization mechanism or the establishment of a shared secret between each pair of nodes as the minimum requirement. It provides loop freedom and protects the nodes from impersonation and several other attacks.

Unlike SEAD, Aridane is based on a reactive protocol, namely, DSR, and it follows an end-to-end approach for building a security mechanism [28]. Aridane assumes the existence of a shared secret key between two nodes and uses

a message authentication code (MAC) in order to authenticate point-to-point messages between nodes.

Zapata [39] proposed an additional routing protocol that uses hash chains to provide security features; it is the secure ad hoc on-demand distance vector (SAODV). SAODV proposes a set of extensions that secures the AODV routing packets. For authenticating the nonmutable fields, it uses cryptographic signatures, while one-way hash chains are used for securing every different route discovery process. In order to carry out the asymmetric cryptography, it requires the existence of a key management mechanism. Because cryptography-based prevention techniques consume much energy and they are invalid to internal attacks, other efficient security schemes should be researched in ad hoc network.

5.2.2. Intrusion detection and response in MANET

Zhang and Lee [40] build on the completely distributed structure of wireless ad hoc networks. Every node in the network participates in the process of intrusion detection. Each node is responsible for detecting intrusion locally and independently based on the data collected by it. They use data on the node's physical movements and the corresponding change in its routing table as the trace data to build the anomaly detection model. When the local detector finds a malicious node, it is broadcast to the entire network. Each node also makes a final decision based on the detection reports from other nodes. Because all the nodes run on the local detection engine that analyzes local data for anomalies, it is too expensive to detect some special attacks.

Multiple sensors intrusion detection system for ad hoc wireless networks based on mobile agent technology has been proposed in [41]. Audit data is efficiently merged from multiple network sensors, and the entire ad hoc wireless network is analyzed for intrusions, and it is tried to inhibit intrusion attempts. A multisensor intrusion detection system employing cooperative detection algorithm has been introduced. A mobile agent implementation is chosen, to support such features of the IDS system as mobility of sensors, intelligent routing of intrusion data throughout the network, and lightweight implementation. It employs several sensor types that perform specific certain functions, such as network monitoring, host monitoring, decision making, and action. There are three major agent classes: monitoring, decision-making, and action agents. Some are present on all mobile hosts, while others are distributed to only a select group of nodes. The monitoring agent class consists of packet, user, and system monitoring agents. Because of scarce computational and power resources in mobile nodes, multiple sensors and agent communication have heavy pressure on ad hoc network.

6. WHY SPA-ARA

The application of SPA-ARA to MANETs has many advantages. SPA-ARA is reliable, survivable, and secure routing algorithm. SPA-ARA has many of the on-demand and table routing advantages and at the same time avoids many of

their drawbacks. In addition to that, SPA-ARA is self-built and self-configured optimization algorithm that matches the characteristics of MANETs. It depends on routing tables (*routing pheromone table and trust pheromone table*), thus providing a high number of redundant and already graded paths to the destination, which increases the survivability of the algorithm. When the best path fails (due to mobility, node battery depletion, decrease in trust value due to misbehavior, etc.), the algorithm immediately uses the next available path. These paths increase the survivability of the algorithm. The updating of the tables is done on demand, and it is done mainly in the nodes, which leads to the best paths. This lowers the overhead compared to both table-driven algorithms and on-demand algorithms. Both table-driven and on-demand algorithms update needed and unneeded paths to the destination.

To prove the validity of our proposed algorithm, SPA-ARA, we have presented a comparison with the representative reactive (on-demand) protocols, that is, ad hoc on-demand distance vector (AODV) [6] and dynamic source routing (DSR) [42]. Comparison has also been carried out with a representative ant routing protocol, ant-colony-based routing algorithm (ARA) [11], and the comparison of security features of the proposed protocol has been done with an on-demand secure protocol, secure routing protocol (SRP) [43]. Salient features of all the protocols are given in the next section.

6.1. AODV

AODV is an improvement on the destination-sequenced distance-vector (DSDV) protocol. It minimizes the number of route broadcasts by creating routes on an on-demand basis. Route discovery is initiated on an on-demand basis; the route request is then forwarded to the neighbors, and so on until either the destination or an intermediate node with a fresh route to the destination is located.

When a node wants to find a route to a destination node, it broadcasts an RREQ message with a unique RREQ ID (RID) to all its neighbors. When a node receives an RREQ message, it updates the sequence number of source node and sets up reverse routes to the source node in the routing tables. The reverse route will be used to send the corresponding RREP message to the originating node. It also updates the sequence number of the destination node in its routing table to the maximum of the one in its routing table and the one in the RREP message. If the node is the destination or the node has a route to the destination that meets the freshness requirements, it unicasts an RREP back to the source node. The source node or the intermediate node that receives RREP will update its forward route to destination in the routing tables. Otherwise, it continues broadcasting the RREQ. If a node receives an RREQ message that has already been processed, it discards the RREQ and does not forward it. AODV maintains the connectivity of neighbor nodes by sending the hello message periodically. From this description, it is clear that the protocol is an efficient on-demand protocol. On-demand routing protocols have been demonstrated to perform better with significantly lower overheads than proactive

routing protocols in many scenarios since they are able to react quickly to topology changes; yet they are able to reduce routing overhead in areas of the network in which changes are less frequent. The other prominent on-demand protocol is DSR. A performance comparison for the two representative on-demand protocols has been carried out by Das et al. [44], which shows that AODV outperforms DSR in more stressful situations, that is, with more load and larger number of nodes. Also, AODV has been found to be more scalable and the packet delivery ratio is better in case of AODV when compared to DSR [45]. Also, due to lack of any mechanism to expire stale routes or to determine the freshness of routes when multiple choices are available, DSR shows large delay and throughput performances. AODV also has commercial applications in sensor networks and military networks. AODV routing algorithm has been implemented on a Bluetooth-based wireless network [46]. The experimental results show that it is possible to create a commercial application, which can be used in many scenarios to solve practical problems. But AODV does not take up power-aware routing and does not employ any security mechanism.

6.2. DSR

The key feature of DSR [42] is the use of source routing. That is, the sender knows the complete hop-by-hop route to the destination. These routes are stored in a route cache. The data packets carry the source route in the packet header. When a node in the ad hoc network attempts to send a data packet to a destination to which it does not already know the route, it uses a route discovery process to dynamically determine such a route. Route discovery works with flooding the network with route request (RREQ) packets. Each node receiving an RREQ rebroadcasts it unless it is the destination or it has a route to the destination in its route cache. Such a node replies to the RREQ with a route reply (RREP) packet that is routed back to the original source. RREQ and RREP packets are also source-routed. The RREQ builds up the path traversed so far. The RREP routes itself back to the source by traversing this path backwards. The route carried back by the RREP packet is cached at the source for future use.

If any link on the source route is broken, the source node is notified using a route error (RERR) packet. The source removes any route using this link from its cache. A new route discovery process must be initiated by the source if this route is still needed.

DSR makes very aggressive use of source routing and route caching. No special mechanism to detect routing loops is needed. Also, any forwarding node caches the source route in a packet it forwards for possible future use. Other additional features of DSR are *gratuitous route repair* and *promiscuous listening*. *Gratuitous route repair* takes place in the following manner. A source node receiving an RERR packet piggybacks the RERR in the following RREQ. This helps in cleaning up the caches of other nodes in the network that may have the failed link in one of the cached source routes. By employing *promiscuous hearing*, when a node overhears a packet not addressed to itself, it checks whether the packet could be routed via itself to gain a shorter route. If so, the

node sends a gratuitous RREP to the source of the route with this new better route. Aside from this, promiscuous listening helps a node to learn different routes without directly participating in the routing process. DSR always shows a lower routing overhead as compared to AODV [44].

6.3. SRP

We have carried out a comparative analysis of the proposed algorithm with SRP [43]. Selection of SRP as the control algorithm for comparing the security features of the proposed protocol is due to the following reasons. Current efforts toward the design of secure routing protocols are mainly oriented to on-demand protocols because of the advantages of reactive protocols over proactive protocols already mentioned in the previous section. Performance comparison performed on routing protocols by Bo et al. [47] shows that for both on-demand protocols AODV and DSR, throughput declines, average packet delay is high, and the overall performance degrades under security attacks. Therefore, SRP was selected as a candidate protocol for comparing performance under security attacks. SRP is conceived of as an extension that can be applied to a multitude of existing on-demand protocols. SRP combats attacks that can disrupt the route discovery process, and guarantees the acquisition of correct topological information. SRP mitigates the detrimental effects of maliciously behaving nodes that disrupt the route discovery in order to obstruct or disable the network operation. SRP provides correct routing information, that is, factual, up-to-date, and authentic connectivity information regarding a pair of nodes that wish to communicate in a secure manner. In this algorithm, there exists a concept of security association (SA) between the source node and the destination node without the need for the intermediate nodes to cryptographically validate the control traffic. Security association is achieved prior to the route initiation phase through a shared key K_{ST} between the source S and target T . During the route discovery phase, the SRP uses an additional header called SRP header to its RREQ packet. SRP header contains the following fields: the query sequence number Q_{SEC} , query identifier number Q_{ID} , and a 96-bit message authentication code (MAC) field.

When intermediate nodes receive RREQ message, they check the SRP header. If the SRP header is missing, they discard the message; otherwise, they forward the message towards destination after extracting Q_{ID} , source, and destination addresses. After receiving the request packet, target T verifies if the packet has originated from the node with which it has SA. If Q_{SEC} is greater or equal to Q_{MAX} , the request is dropped as it is considered to be replayed. Otherwise, it calculates the keyed hash of the request fields and if the output matches SRP MAC, then the authenticity of the sender and the integrity of the request are verified. Upon receiving the RREP from the destination, source checks the source address, destination address, Q_{ID} , and Q_{SEC} . It discards the RREP if it does not match the currently pending query. In case of match, it compares reply IP source route with the exact reverse of the route carried in reply packet. If the two routes match, then the source calculates the MAC by using

the replied route, the SRP header fields, and the secure key between source and destination. If the two MACs match, then the validation is successful and it confirms that the reply did come from the destination T .

SRP has been proven to be essentially immune to IP spoofing [43]. This feature is also present in SPA-ARA. But a malicious node can harm the route that it belongs to, that is, where the proposed protocol outperforms SRP as shown by the simulation analysis of both protocols later.

7. POWER-AWARE DETECTION FRAMEWORK

SPA-ARA is a multipath protocol and it does not maintain paths to all destinations at all times (like the ACO algorithms for wired networks), but it sets up paths when they are needed at the start of a session. This is done in a *reactive path setup phase*, where ant agents (called *reactive forward ants*) are launched by the source in order to find multiple paths to the destination, and backward ants return to set up the paths. The paths are represented in *pheromone tables* indicating their respective quality. After path setup, data packets are routed stochastically as datagrams over the different paths using these pheromone tables. The algorithm reacts to link failures by warning preceding nodes on the paths.

Every node periodically broadcasts a hello message. With this mechanism, a node can detect its new neighbors. When a new neighbor is found, the node checks the trust level of the neighbor and puts the neighbor along with its trust level in its *trust pheromone table*. Every node sets up a secret key with each trustworthy neighbor by using a two-party key establishment protocol in a manner similar to [48].

The routing information of a node N_i is represented in its *routing pheromone table* $PT(N_i)$ as shown in Table 6. The entry $T_{N_i, N_{i+1}, d}$ of a table is the pheromone value indicating the estimated goodness of going from N_i over neighbor N_{i+1} to reach destination d . It takes into account both end-to-end delay and number of hops. The pheromone values of each entry in the table can be initialized to zero value, thus providing nonbiased search for the best path. We introduce a new routing metric that takes into account the quality of links and nodes available in a connection. We call this metric “next-hop availability,” which is a combination of two metrics. It maximizes path availability and minimizes travel time of packets, and therefore it offers a good balance between selection of fast paths and a better use of network resources. *Next-hop availability* is defined as the probability to find next hops, that is, nodes and links available for routing on a path. If pheromone information is available, and there is more than one path available for routing, then the ant chooses that node as its next hop for which the next-hop availability is maximum.

Another pheromone table known as the *trust pheromone table* $TPT(N_i)$ contains the trust level for all the neighbors as shown in Table 7. The entry in the table is known as trust pheromone (T_p). T_p is a representation of the trust value of a node depending on its reliability. At the beginning, all nodes can be trusted and they are assigned a trust value, which is indicated as trust pheromone in the trust pheromone table of the neighbor node. If the value of the trust pheromone is

FIGURE 2: Flowchart showing initiation of reactive forward ant.

lower than a defined threshold, the node may be considered as a malicious node.

7.1. Reactive path setup phase

Every node in the network can work as a source node, destination node, and/or intermediate node. When a source node S starts a communication session with a destination node D , and it does not have available routing information for S , it temporarily buffers the data and broadcasts a reactive forward ant F_{SD} as shown by Figure 2. The source node before broadcasting the reactive forward ant, F_{SD} , attaches to it a message authentication code (MAC). The MAC covers the whole reactive forward ant and is generated by using a keyed hash algorithm [49] and a shared group key K (to prevent modification from external attackers). Therefore, the F_{SD} now would be

$$\text{Forward reactive ant: } F_{SD} \parallel K(\text{MAC}), \quad (1)$$

where “ $\parallel$ ” denotes concatenation and $K(\text{MAC})$ denotes a MAC generated by using key K . When a node receives an F_{SD} and it is not the destination node, it verifies the MAC of F_{SD} using the shared group key K . If the MAC is correct, the node authenticates the sender by obtaining the *trust pheromone* (T_p) for the sender node from its own *trust pheromone table*. If the *trust pheromone* is above the defined threshold value, the receiver further considers the F_{SD} and establishes a secret key with the sender using a two-party key establishment protocol [48]. If the *trust pheromone* is below the defined threshold value, the receiver kills the ant.

Due to this initial broadcasting, each neighbor of s receives a replica of F_{SD} . We refer to the set of replicas, which originated from the same original ant, as an ant generation. The task of each ant of the generation is to find a path

connecting S and D . At each node, an ant is either unicast or broadcast, according to whether or not the node has routing information for S .

Assume that node N_{i+1} is available for routing with a probability $P_n(N_{i+1})$ and the link (N_i, N_{i+1}) is available with a probability $P_l(N_i, N_{i+1})$. Next-hop availability $P_{nh}(N_i, N_{i+1})$ for the node N_{i+1} from the node N_i is expressed as

$$P_{nh}(N_i, N_{i+1}) = P_n(N_{i+1}) \times P_l(N_i, N_{i+1}). \quad (2)$$

In real systems, many factors contribute to the values of P_n and P_l . In this algorithm, we have expressed P_n in terms of the remaining battery life and P_l with respect to end-to-end delay and number of hops. The methodology for calculating both P_n and P_l is discussed later in the next sections.

Due to broadcasting, ants can proliferate quickly over the network, following different paths to the destination. However, ants which have reached a maximum number of hops, related to the network diameter, are killed or discarded. When a node receives several ants of the same generation, it compares the path traveled by each ant to that of the previously received ants of this generation. Only if its number of hops and travel time are both less than those of the best ant of the generation, it will forward the ant. Using this policy, overhead is limited by removing ants, which follow bad paths. However, it does have as an effect that the ant, which arrives first in a node, is let through, while subsequent ants meet selection criteria set by the best of the ants preceding them. So, they have higher chances of being killed. Duplicate ants, which result from a broadcast of the best ant just before it reaches the destination, are close in performance to the best ant and have higher chances of being accepted. After the above procedure, all the nodes involved are authenticated and a secret key is set up between each pair of neighbor nodes. The untrustworthy or suspicious nodes are excluded in the process.

Each forward ant keeps a list P of the nodes $[1, \dots, n]$ it has visited. Upon arrival at the destination d , it is converted into a backward ant B_{DS} , which travels back to the source retracing P (if this is not possible because the next hop is not there, e.g., due to node movements, the backward ant is discarded). The destination node attaches to the B_{DS} a MAC calculated by using the secret key that destination D shares with the next hop N for moving towards the source. Therefore, the updated B_{DS} would now be

$$\text{Backward reactive ant: } B_{DS} \parallel K_{DN}(\text{MAC}). \quad (3)$$

When node N receives the reactive backward ant, it verifies the MAC. If it is correct, the *routing pheromone table* entry gets updated. The *trust pheromone table* entry also gets updated. The process of updating the *routing pheromone table* and *trust pheromone table* is explained a little later. Also, N replaces the MAC in the B_{DS} by a MAC calculated using a secret key that it shares with the next hop obtained from the *routing pheromone table* of N . Let the next hop now be M . In this case, the updated reactive forward ant would be

$$\text{Reactive forward ant: } B_{DS} \parallel K_{NM}(\text{MAC}). \quad (4)$$

At each node N_i , the backward ant sets up a path towards the destination D , creating or updating the value of pheromone table entry $\zeta_{N_i, N_{i+1}, d}$ in the table $PT(N_i)$. This process is repeated and finally the source node receives B_{DS} . Therefore, a number of good secure paths are set up between the source and the destination. The entire process is depicted in Figure 3.

If, on the other hand, no backward ant has come back to the source after a certain amount of time, data are temporarily buffered and the whole process is restarted. This is repeated for a maximum number of times, after which the buffered data are discarded.

7.2. Energy sensitivity

Batteries are the major source of energy in mobile nodes. To provide greater portability, batteries need to be small and lightweight, which unfortunately restricts the total energy that they can carry. Once batteries exhaust their energy, they need to be replaced or recharged, which typically reduces the independence of a mobile node for few hours of operation.

Energy consumption, in communication-related tasks, depends on the communication mode of a node. A node may be transmitting, receiving, or in idle mode. Naturally, transmission consumes more energy than the other two modes. From the routing perspective, our interest is in selecting routes in such a way that the transmission and reception of packets are intelligently distributed on the network so as to maximize the overall average battery lifetime of the nodes. Therefore, we are interested in getting forward ant agents to select, with greater frequency, those nodes which have the longest remaining battery lifetime. If B_r represents the remaining battery lifetime of node N_i , $P_n(N_i)$ can be expressed as

$$P_n(N_i) = B_r \div B_m, \quad (5)$$

where B_m is the lifetime of a fully charged battery.

Therefore, from (2) and (5), next-hop availability $P_{nh}(N_i, N_{i+1})$ for the node N_{i+1} from the node N_i becomes

$$P_{nh}(N_i, N_{i+1}) = B_{r_{i+1}} \div B_m \times P_l(N_i, N_{i+1}). \quad (6)$$

If the nodes batteries' remaining energy is not considered in the optimization, the best path's node energy will be used more unfairly than the other nodes in the network. These nodes may fail after a short time because of their battery depletion, whereas other nodes in the network may still have high energy in their batteries.

7.3. Routing pheromone table update

The backward ant incrementally computes an estimate T_P of the time it would take a data packet to travel over the path $P[1, \dots, n]$ towards the destination, which is used to update the pheromone routing tables. T_P is the sum of local estimates $T(N_i, N_{i+1})$ in each node $N_i \in P$ of the time to reach the next hop N_{i+1} , and it is given as

$$T_P = \sum_{i=1}^{n-1} T(N_i, N_{i+1}). \quad (7)$$

FIGURE 3: Flowchart describing node's function.

$T(N_i, N_{i+1})$ is expressed as

$$T(N_i, N_{i+1}) = (Q(N_i)_{\text{mac}} + 1) \times T(N_i)_{\text{mac}}, \quad (8)$$

where $Q(N_i)_{\text{mac}}$ is the number of packets in the queue at the MAC layer and $T(N_i)_{\text{mac}}$ is calculated as a running average of the time elapsed between the arrival of a packet at the MAC layer and the end of a successful transmission. Since $T(N_i)_{\text{mac}}$ is calculated at the MAC layer, it includes channel access activities; so it accounts for local congestion of the shared medium. Forward ants also calculate a similar time estimate T_P , which is used for filtering the ants, as mentioned before.

The pheromone table entry $\zeta_{N_i, N_{i+1}, d}$ in the table $PT(N_i)$ of node N_i represents a running average of the inverse of the cost, in terms of both estimated time and number of hops, to travel to destination d through node N_{i+1} . If $T(N_i, d)$ is the traveling time estimated by the ant and h is the number of hops, the value of the updating factor $\alpha(N_i, d)$, which is used to update the running average, is expressed as

$$\alpha(N_i, d) = ((T(N_i, d) + h \times T_{\text{hop}}) \div 2)^{-1}, \quad (9)$$

where T_{hop} is a parameter representing the time to take one hop under unloaded conditions. So the pheromone table up-

dating factor $\alpha(N_i, d)$ takes into account both end-to-end delay and number of hops. The value of $\zeta_{N_i, N_{i+1}, d}$ is updated as

$$\zeta_{N_i, N_{i+1}, d} = \gamma \times \zeta_{N_i, N_{i+1}, d} + (1 - \gamma) \times \alpha(N_i, d), \quad \gamma \in [0, 1]. \quad (10)$$

7.4. Trust pheromone table update

If a node in a network finds another node dropping packets, it may consider that node malicious and decreases its trust value in its *trust pheromone table* so that the node is not able to participate in reactive path setup phase. If a node receives B_{SD} from a trustworthy node, it further reinforces its trust level in its table.

7.5. Link availability

As explained earlier, the value of probability for availability of a link P_l takes into account the number of hops and end-to-end delay. It is expressed as

$$P_l(N_i, N_{i+1}) = \zeta_{N_i, N_{i+1}, d} \div \sum_{j \in N_{jd}} \zeta_{N_i, N_j, d}, \quad (11)$$

TABLE 4: Simulation parameters.

Network size	3000 m × 3000 m
Number of nodes	20, 49
Mobility model	Random
Minimum node speed	5–20 m/s
Pause time	0, 4, 8, 16, and 24 s
Data rate	1 message per minute
Time	600 s

where N'_{jd} is the set of neighbors of N_i over which a path to d is known. Therefore, from (6) and (11), we have

$$P_{nh}(N_i, N_{i+1}) = (B_{r_{i+1}} \div B_m) \times \left(\zeta_{N_i, N_{i+1}, d} \div \sum_{j \in N'_{jd}} \zeta_{N_i, N_j, d} \right). \quad (12)$$

7.6. Protection of data packet and routing

It is also essential to provide protection for the data packet in an ad hoc network. For prevention of data packet modification, each node calculates a MAC by using the secret key it shares with the next hop and attaches the MAC to the data packet. When the next hop receives the packet, it can verify the MAC using the secret key to check the integrity of the data packet.

Data can be sent using many techniques. It can be sent along the maximum pheromone path, sent following path of the best forward ant agent, or sent along the path using maximum value of next-hop availability or a combination of them.

8. IMPLEMENTATION AND PERFORMANCE EVALUATION

We have implemented the SPA-ARA algorithm using scalable wireless ad hoc network simulator (SWANS) [50]. SWANS is built atop the JIST [51] platform, a general-purpose discrete event simulation engine. The size of the field was 3000 × 3000 m² area. Different number of nodes were tested. Range of each node was assumed to be 625 m. Different speeds (from 5 to 20 m/s) of movement were tested. Random mobility was assumed with the pause time of 4 seconds. The algorithm was also tested for different pause times, which were 0, 4, 8, 16, and 24 seconds. Pause time of 0 second corresponds to continuous motion of the node. Simulation was also performed for varying number of malicious nodes. The algorithm was tested for different scenes for each set of parameters. Table 4 shows the simulation parameters. Each of the runs was repeated five times with different seeds, which resulted in a slightly changed order of events. These five runs were merged together to build an average case, which was then used for comparison. Therefore, each data point shown in Figures 4, 5, 6, and 7 represents an average of five runs.

All the nodes use batteries as their source of energy. The energy in a battery will decrease when a packet is sent or received. In this paper, a battery model for the “Lucent wave-

LAN PC card 2.4 GH direct sequence spread spectrum” [52] has been simplified and used in the node models. From this battery model, only the power consumed by routing (forward reactive ants, backward ants, hello messages) and data packets is considered in this battery model. The power consumed by lower layers is not included.

The equations governing the battery model are as follows (simplified version of [52]).

When a node sends a packet

$$\text{Node_Energy} = \text{Node_Energy} - E_Send \cdot \text{PacketSize(Bytes)}. \quad (13)$$

When a node receives a packet

$$\text{Node_Energy} = \text{Node_Energy} - E_Rec \cdot \text{PacketSize(Bytes)}, \quad (14)$$

where

$$\begin{aligned} E_Send &= 2 \mu \text{ WattSecond/Byte}, \\ E_Rec &= 0.5 \mu \text{ WattSecond/Byte}. \end{aligned} \quad (15)$$

8.1. Fair nodes' local energy usage distribution

To achieve energy-efficient communication in MANETs, different techniques have been employed which may be *power saving routing*, *power control*, and *maximum lifetime routing*. A *power saving protocol* puts a node's network interface into the sleep state in order to save energy. *Power control* techniques allow nodes to modify their transmit power to increase network capacity and reduce energy consumption. *Maximum lifetime* routing increases the network lifetime by balancing the energy consumption across the network. In MANETs, a large number of nodes depend on limited energy sources such as batteries as their source of energy. If the nodes batteries' remaining energy is not considered in the optimization, the best path's nodes energy will be used more unfairly than the other nodes in the network. These nodes may fail after a short time because of their battery depletion. However, other nodes in the network may still have high energy in their batteries.

SPA-ARA employs maximum lifetime routing approach to achieve energy-efficient communication between the nodes of the network. Therefore, the energy distribution across the network has been visualized, which is better in SPA-ARA as compared to AODV, DSR, and ARA. So in this approach, we are not trying to control power or save power, but we are trying to select energy-aware routes to increase network lifetime. Therefore, SPA-ARA has been applied to optimize the number of hops and the nodes batteries' remaining energy. In this section, we present the application of SPA-ARA, AODV, DSR, and ARA to 20- and 49-node networks. We will use the following definition. Energy standard deviation is the remaining nodes' energy standard deviation from the average remaining energy for all the network nodes.

Figures 4(a) and 4(b) show the energy standard deviation for 20 and 49 nodes, respectively. From these figures, we notice that the maximum energy standard deviation is 48 Watt·s in case of AODV, 43 Watt·s in case of DSR, 47 Watt·s

FIGURE 4: Node's energy standard deviation.

TABLE 5: Maximum energy standard deviation.

Protocol	Energy standard deviation (Watt·s)	
	(20 nodes)	(49 nodes)
SPA-ARA	37	63
AODV	48	88
DSR	43	80
ARA	47	94
(AODV-SPA-ARA/AODV)%	23	28
(DSR-SPA-ARA/DSR)%	14	21
(ARA-SPA-ARA/ARA)%	21	33

TABLE 6: Routing pheromone table at node N_i .

Destination	Neighbor node			
	N_{i+1}	N_{i+2}	...	N_n
d_1	T_{N_i, N_{i+1}, d_1}	T_{N_i, N_{i+2}, d_1}	—	T_{N_i, N_n, d_1}
d_2	T_{N_i, N_{i+1}, d_2}	T_{N_i, N_{i+2}, d_2}	—	T_{N_i, N_n, d_2}
—	—	—	—	—
d_n	T_{N_i, N_{i+1}, d_n}	T_{N_i, N_{i+2}, d_n}	—	T_{N_i, N_{i+2}, d_n}

in case of ARA, and 37 Watt·s in case of SPA-ARA for 20 nodes. In Figure 3(b), it is 88 Watt·s for AODV, 80 Watt·s for DSR, 94 Watt·s for ARA, and 63 Watt·s for SPA-ARA. This is because the energy usage is distributed across the network nodes in SPA-ARA. Table 5 shows that the difference in maximum energy standard deviation is lower by 23% for SPA-ARA than for AODV in one case and by 28% in another case. When compared to DSR the difference in energy standard deviation is lower by 14% and 21%, and when compared to ARA the difference is lower by 21% and 33%.

8.2. Survivability in larger networks

The algorithm has been tested for small as well as larger networks. We have noticed that the algorithm has the ability to

FIGURE 5: Successful routes found.

respond to dynamic changes in the network due to mobility. Figure 5 shows the number of routes found during the simulation for a 49-node network. From this figure, we can see that the number of successful routes discovered for SPA-ARA algorithm is more than for AODV, DSR, and ARA, and it goes on increasing with the simulation time.

8.3. Destination location capability

In order to compare the performance of our proposed protocol with another secure routing protocol SRP [43], we evaluated them with respect to the following metrics: *destination location time* and *number of packets dropped by malicious nodes*.

Figure 6 presents the average time it takes an F_{SD} to reach its destination for the first time. The route request of the SRP propagates the request towards the destination faster than SPA-ARA since it rejects any variant of a specific request. The F_{SD} of the SPA-ARA has slightly longer living times than

FIGURE 6: Destination location time.

FIGURE 7: Packets dropped by malicious nodes.

TABLE 7: Trust pheromone table.

Neighbor node	Trust pheromone
N_1	T_{p1}
N_2	T_{p2}
—	—
N_n	T_{pn}

SRP. This is reasonable as it attempts to authenticate messages by using trust levels provided in the trust pheromone table along with the MAC attached with the messages calculated using the shared keys.

8.4. Malicious node identification

As shown in Figure 7, the number of packets dropped by malicious nodes using SPA-ARA routing security scheme is less than SRP. As the number of malicious nodes increases, the difference between SPA-ARA and SRP becomes more significant. The malicious nodes can be identified by SPA-ARA. After this identification, their trust pheromone in the trust pheromone tables of their neighboring nodes is decreased below the threshold level. Therefore, there are less packets going through the malicious nodes, and adversaries drop fewer packets maliciously. Power consumption is a critical limitation of mobile hosts in mobile ad hoc networks. If there is a malicious node with sufficient power supply, it can send lots of packets to attack other nodes. Once mobile nodes receive these packets, they may have to relay these packets or record route entries. Thus, these attacking packets may consume power of mobile nodes. Therefore, by identification and isolation of these malicious nodes, power consumption of mobile nodes is also reduced.

8.5. Power awareness and security analysis

Now, we present an analysis of how the proposed protocol is able to select the best optimum path in terms of number of

FIGURE 8: An example MANET with Q as compromised node.

hops and power of batteries, and at the same time provide defense against various types of attacks.

8.5.1. Power awareness

The battery's remaining energy is being used for optimization, in addition to the primary optimization parameter. If the remaining local energy in the nodes' batteries is considered as a parameter in the optimization, the network shown in Figure 8 may respond as follows. The path S-X-V-D has the minimum number of hops and it will be used. When the node battery's remaining energy in this path decreases to a certain threshold, the S-Q-R-T-D path information will be better and the data packets will use this path. When the remaining energy of node batteries of this path decreases to the threshold, the data packets will switch to the S-P-Y-Z-T-D path and so on.

8.5.2. External attacks

Fabrication threats from external attackers are prevented, because the external attackers are excluded by authentication scheme. An external attacker may try to modify a routing message. Since external attackers do not possess the group key K or the secret key shared between each pair of nodes, they cannot generate correct MAC corresponding to the message. So, modifications can be prevented.

Threats using impersonation, that is, IP spoofing, can be prevented too. There are two possibilities. First, an attacker may impersonate a nonexistent node, but the authentication scheme will keep it out. Second, the attacker may impersonate an existent node, for example, node P in Figure 8, and replay an eavesdropped reactive forward ant. When node Q receives this F_{SD} , it will accept the message if it has never received it. But when Q receives the corresponding B_{DS} later, Q will forward the B_{DS} to P rather than the attacker. Thus, node P will detect the problem if node P is in the transmission range of node Q . If P is out of Q 's transmission range, the B_{DS} is lost.

8.5.3. Internal attacks

If a node is compromised, then the attackers may control everything at the node: the shared group key K , the shared secret keys with other nodes, and the *trust pheromone table*. The attacker can pass authentication and generate correct MAC. For example, suppose that node Q in Figure 8 becomes a compromised node. Other nodes in Figure 8 are good nodes. Possible attacks are analyzed as follows.

(a) Attacks on reactive forward ant (F_{SD}): when attacker Q receives an F_{SD} from P , the following attacks are possible.

(i) IP spoofing attack:

When forwarding F_{SD} to R , Q may put an incorrect IP address in the message. Q has two choices: (1) using nonexistent IP address, which will be detected by R , since it cannot be authenticated; (2) using existing IP address, for example, IP address of node Y . Node R will accept it if Y is in R 's *trust pheromone table* and can be trusted. Otherwise, R will try to authenticate Y and set up a key with Y . This requires Y to be in R 's transmission range. If not, R will drop the F_{SD} and the attack fails. But even if R accepts the F_{SD} , when R receives the corresponding B_{SD} from T or V , R will forward it to Y rather than Q . There are three possibilities. (1) Y is not participating in the current reactive path setup. Y would detect the problem since it has never forwarded the F_{SD} . (2) Y is a participator within the transmission range of R . The attack helps in reactive path setup other than doing something harmful, since Y should receive a B_{SD} from R . (3) Y is a participator and not in the transmission range of R ; the B_{SD} will be lost as Y cannot receive it. Therefore, the attack fails.

(ii) Modification:

Attacker Q can modify the source address (IP_s) of F_{SD} and generate a correct MAC. When R receives the modified F_{SD} , R will not detect it. The destination node D will not detect it either. When Q receives the B_{SD} from R , Q modifies it back to the correct one and then forwards it to P . The final result is still a correct path from S to D , although it is modified during the process of reactive path setup. So the attacker is not able to disrupt the route discovery. An attacker can modify the destination address (IP_d) of F_{SD} . If IP_d is modified to a nonexistent IP address, eventually the F_{SD} will disappear. If Q modifies IP_d from the IP address of D to

IP addresses of some other nodes, for example, IP address of R , node R will not find the problem. When Q receives the corresponding B_{SD} from R , Q modifies IP_d back to the IP address of node D and forwards it to P ; P will accept it. So the attack has been unsuccessful. But after Q modifies the F_{SD} , it needs to broadcast the modified F_{SD} . When other nodes receive it, they will rebroadcast it. Due to the flooding of F_{SD} , the original source of F_{SD} , that is, node S , will receive the modified F_{SD} and detect the problem.

(b) Attacks on reactive backward ant (B_{SD}): when attacker Q receives B_{SD} from R , the following attacks are possible.

(i) IP spoofing:

When Q forwards the RREP to P , Q uses some other nodes' IP addresses instead of using its IP address to corrupt P 's *routing pheromone table*. If Q uses a nonexistent IP address, P will detect it because there is no key shared between P and the nonexistent IP address. If Q wants to use the IP address of an existing node, for example, IP address of node Y , it requires that Y is in P 's *trust pheromone table*, and Q needs to break the secret key shared between P and Y . This makes the attack difficult to succeed.

(ii) Modification:

Attacker Q can modify the source and destination addresses IP_s and IP_d of B_{SD} . But nodes P and S will find the problem and drop the modified B_{SD} , because there is no corresponding F_{SD} .

(c) Fabrication of route reply: when attacker Q receives a reactive path setup, Q can forge a B_{SD} instead of forwarding the route request to other nodes. Q can copy the source and destination addresses from the B_{SD} . Since Q knows the secret keys that it shares with P and S , Q can generate the correct MAC for the B_{SD} . So Q is able to fabricate a correct route reply and send it to P and S . Nodes P and S will not detect the problem. They will think that the B_{SD} is forwarded by Q and accept it. So an invalid path is established. But our protocol discovers multiple paths; so there are still other paths to be used. If there is a secret key shared between the source and destination nodes, this problem can be prevented by utilizing a MAC calculated using the secret key.

8.5.4. Message authentication

Our protocol makes use of pairwise authentication. Each pair of neighboring intermediate nodes authenticates the routing messages. For F_{SD} , two neighboring nodes authenticate each other through the authentication process. For B_{SD} , a MAC generated by using the secret key shared between two neighboring nodes provides message authenticity. If the destination node is in the *trust pheromone table* of the source node or vice versa, the two ends already establish a shared secret key. End-to-end authentication can also be utilized besides pairwise authentication. Each reactive forward ant carries two MACs: one for pairwise authentication and one for end-to-end authentication, which is generated by using the secret key shared between the source and destination nodes.

9. CONCLUSION

SPA-ARA is an efficient routing algorithm for managing the energy usage and security in MANETs. It is a dynamic routing algorithm with controlled routing overheads. The routing packets are concentrated in the best paths' regions. This allows for better optimization with lower number of packets. In addition, old explored, unexplored, and bad regions are visited with lower rate. SPA-ARA is self-built and self-configured optimization algorithm that matches the characteristics of MANETs. The algorithm can use different parameters in the optimization process.

We have introduced a new routing metric to cope with the limited power resources of mobile networks. *Next-hop availability*, which models the probability to find the best available next hop to be taken by the reactive ant and data packets to reach the destination, was employed to take into account a number of parameters. These optimization parameters were the number of hops, travel time, and the batteries' remaining energy. The algorithm has the ability to continuously check for better paths in the network with controlled overheads, which make the algorithm more suitable for network resources management. SPA-ARA combines many advantages of the on-demand-based and table-based routing algorithms. Simulation results show the ability of the algorithm to find the optimum solution and to achieve fair energy usage distribution as an example of network resources management.

At each node, there exists a *routing pheromone table* as well as *trust pheromone table*. The *trust pheromone table* gets updated according to the trustworthiness possessed by a neighbor. This trust level helps in isolating malicious and compromised nodes present in the network. Integrity is ensured by MAC, which is calculated using a shared key in forward path setup and pairwise shared secret keys in the backward path. The protocol can detect most of the attacks, which are common to ad hoc network routing protocols. Moreover, the protocol is capable of discovering secure paths as well as the most power-optimized path for dynamically changing topology.

REFERENCES

- [1] T. B. Reddy, I. Karthigeyan, B. S. Manoj, and C. S. R. Murthy, "Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions," *Ad Hoc Networks*, vol. 4, no. 1, pp. 83–124, 2006.
- [2] S. Guo and O. Yang, "QoS-aware minimum energy multicast tree construction in wireless ad hoc networks," *Ad Hoc Networks*, vol. 2, no. 3, pp. 217–229, 2004.
- [3] M. Dorigo and G. Di Caro, "The ant colony optimization meta-heuristic," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds., pp. 11–32, McGraw-Hill, New York, NY, USA, 1999.
- [4] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, no. 2, pp. 46–55, 1999.
- [5] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *Computer Communications Review*, vol. 24, no. 4, pp. 234–244, 1994.
- [6] C. E. Perkins and E. Royer, "Ad hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop Mobile Computing Systems Applications*, pp. 90–100, New Orleans, La, USA, February 1999.
- [7] Y.-C. Hu, D. B. Johnson, and A. Perrig, "SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 1, pp. 175–192, 2003.
- [8] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shield, and E. M. B. Royer, "A secure routing protocol for ad hoc networks," in *Proceedings of the 10th International Conference on Network Protocols (ICNP '02)*, pp. 78–87, IEEE Press, Paris, France, November 2002.
- [9] W. J. Gutjahr, "ACO algorithms with guaranteed convergence to the optimal solution," Tech. Rep. ISDS 2001-02, University of Vienna, Vienna, Austria.
- [10] T. Stutzle and M. Dorigo, "A short convergence proof for a class of ACO algorithms," Tech. Rep. 35, IRIDIA, Brussels, Belgium, 2000.
- [11] M. Günes and O. Spaniol, "Routing algorithms for mobile multihop ad hoc networks," in *Proceedings of International Workshop on Next Generation Network Technologies*, pp. 20–24, October 2002.
- [12] J. S. Baras and H. Mehta, "A probabilistic emergent routing algorithm for mobile ad hoc networks," in *Proceedings of the Conference on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '03)*, Sophia-Antipolis, France, March 2003.
- [13] M. Heissenbüttel and T. Braun, "Ants-based routing in large scale mobile ad hoc networks," in *Proceedings of the Kommunikation in Verteilten Systemen (KiVS '03)*, March 2003.
- [14] M. Roth and S. Wicker, "Termite: emergent ad-hoc networking," in *Proceedings of the 2nd Mediterranean Workshop on Ad-Hoc Networks (MedHoc '03)*, Mahdia, Tunisia, June 2003.
- [15] C.-C. Shen, C. Jaikao, C. Srisathapornphat, Z. Huang, and S. Rajagopalan, "Ad hoc networking with swarm intelligence," in *Proceedings of the 4th International Workshop on Ant Colony, Optimization and Swarm Intelligence (ANTS '04)*, vol. 3172 of *Lecture Notes in Computer Science*, pp. 262–269, Springer, Brussels, Belgium, September 2004.
- [16] M. N. Doja and S. Mehruz, "A comprehensive review of power conservation strategies for mobile ad hoc networks," in *Proceedings of International Conference on Wireless Networks*, pp. 12–18, Hyderabad, India, December 2005.
- [17] S. Singh, M. Woo, and C. S. Raghavendra, "Power aware routing in mobile ad hoc networks," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM '98)*, pp. 181–190, Dallas, Tex, USA, October 1998.
- [18] C.-K. Toh, "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks," *IEEE Communications Magazine*, vol. 39, no. 6, pp. 138–147, 2001.
- [19] Q. Li, J. Aslam, and D. Rus, "Online power-aware routing in wireless ad-hoc networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM '01)*, pp. 97–107, Rome, Italy, July 2001.
- [20] M. Maleki, K. Dantu, and M. Pedram, "Power-aware source routing protocol for mobile ad hoc networks," in *Proceedings of International Symposium on Low Power Electronics and Design, Digest of Technical Papers*, pp. 72–75, Monterey, Calif, USA, August 2002.
- [21] R. Oppliger, *Internet and Intranet Security*, Artech House, Norwood, Mass, USA, 1998.

- [22] A. Cardenas, N. Benammar, G. Papageorgiou, and J. Baras, "Cross-layered security analysis of wireless ad hoc networks," in *Proceedings of the 24th Army Science Conference*, p. 3, Orlando, Fla, USA, November 2004.
- [23] J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla, and S. Lu, "Adaptive security for multi-layer ad-hoc networks," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 533–547, 2002.
- [24] L. Blažević, L. Buttyán, S. Čapkun, S. Giordano, J.-P. Hubaux, and J.-Y. Le Boudec, "Self-organization in mobile ad hoc networks: the approach of terminodes," *IEEE Communications Magazine*, vol. 39, no. 6, pp. 166–173, 2001.
- [25] P. Kyasanur and N. H. Vaidya, "Detection and handling of MAC layer misbehavior in wireless networks," in *Proceedings of International Conference on Dependable Systems and Networks*, pp. 173–182, San Francisco, Calif, USA, June 2003.
- [26] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 275–283, Boston, Mass, USA, August 2000.
- [27] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Aridane: a secure on-demand routing protocol for ad hoc networks," in *Proceedings of the 8th International Conference on Mobile Computing and Networking (MOBICOM '02)*, pp. 12–23, ACM Press, Atlanta, Ga, USA, September 2002.
- [28] Y. Hu, A. Perrig, and D. Johnson, "Packet leases: a defense against wormhole attacks in wireless networks," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, vol. 3, pp. 1976–1986, New York, NY, USA, June 2002.
- [29] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," in *Proceedings of International Conference on Network Protocols (ICNP '01)*, pp. 251–260, Riverside, Calif, USA, November 2001.
- [30] L. Buttyán and J. Hubaux, "Enforcing service availability in mobile ad-hoc WANs," in *Proceedings of the 1st Annual International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 87–96, Boston, Mass, USA, August 2000.
- [31] J.-P. Hubaux, L. Buttyán, and S. Čapkun, "The quest for security in mobile ad hoc networks," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '01)*, pp. 146–155, Long Beach, Calif, USA, October 2001.
- [32] P. Papadimitratos, Z. J. Haas, and E. G. Sirer, "Path set selection in mobile ad hoc networks," in *Proceedings of the 3rd International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '02)*, pp. 1–11, Lausanne, Switzerland, June 2002.
- [33] B. DeCleene, L. Dondeti, S. Griffin, et al., "Secure group communications for wireless networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM '01)*, vol. 1, pp. 113–117, Washington, DC, USA, November 2001.
- [34] C. Perkins, "Ad hoc on-demand distance-vector routing (AODV)," IETF draft, 2001.
- [35] B. Sun, K. Wu, and U. W. Pooch, "Towards adaptive intrusion detection in mobile ad hoc networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '04)*, vol. 6, pp. 3551–3555, Dallas, Tex, USA, November 2004.
- [36] C. Douligieris and A. Mitrokotsa, "DDOS attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks*, vol. 44, no. 5, pp. 643–666, 2004.
- [37] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13–64, 2003.
- [38] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in mobile ad hoc networks: challenges and solutions," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 38–47, 2004.
- [39] M. G. Zapata, "Secure Ad Hoc On-Demand Distance Vector (SAODV) Routing," IETF Internet-Draft, draft-guerrero-manet-saodv-00.txt, October 2001.
- [40] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 275–283, Boston, Mass, USA, August 2000.
- [41] O. Kachirski and R. Guha, "Effective intrusion detection using multiple sensors in wireless ad hoc networks," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS '03)*, vol. 2, pp. 57–65, Big Island, Hawaii, USA, February 2003.
- [42] J. Uroch, D. Johnson, and A. D. Maltz, "The dynamic source routing protocol for mobile ad hoc networks," IETF Internet Draft, December 1998, <http://www.ietf.org/internetdrafts/draft-ietf-manet-dsr-01.txt>.
- [43] P. Papadimitratos and Z. Hass, "Secure routing for mobile ad hoc networks," in *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS '02)*, pp. 27–31, San Antonio, Tex, USA, January 2002.
- [44] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 1, pp. 3–12, Tel Aviv, Israel, March 2000.
- [45] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks*, vol. 2, no. 1, pp. 1–22, 2004.
- [46] O. Guerra and M. Sánchez-Adame, "Network ad hoc routing algorithm: an application with bluetooth," in *Proceedings of the 17th International Conference on Electronics, Communications and Computers (CONIELECOMP '07)*, p. 6, Cholula, Mexico, February 2007.
- [47] S. M. Bo, H. Xiao, A. Adereti, and J. A. Malcolm, "A performance comparison of wireless ad hoc network routing protocols under security attack," in *Proceedings of the 3rd International Symposium on Information Assurance and Security (IAS '07)*, pp. 50–55, Manchester, UK, August 2007.
- [48] N. Asokan and P. Ginzboorg, "Key agreement in ad hoc networks," *Computer Communications*, vol. 23, no. 17, pp. 1627–1637, 2000.
- [49] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," Request for Comments RFC 2104, February 1997.
- [50] R. Barr, SWANS- Scalable wireless ad hoc network simulator: user guide and tutorial, March 2004.
- [51] R. Barr, JiST-Java in simulation time: user guide and tutorial, September 2003.
- [52] L. M. Feeney, "An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 6, no. 3, pp. 239–249, 2001.