A Survey of Secure Mobile Ad Hoc Routing Protocols

Loay Abusalah, Ashfaq Khokhar, and Mohsen Guizani

Abstract — **Several routing protocols have been proposed in recent years for possible deployment of Mobile Ad hoc Networks (MANETs) in military, government and commercial applications. In this paper, we review these protocols with a particular focus on security aspects. The protocols differ in terms of routing methodologies and the information used to make routing decisions. Four representative routing protocols are chosen for analysis and evaluation including: Ad Hoc on demand Distance Vector routing (AODV), Dynamic Source Routing (DSR), Optimized Link State Routing (OLSR) and Temporally Ordered Routing Algorithm (TORA). Secure ad hoc networks have to meet five security requirements: confidentiality, integrity, authentication, non-repudiation and availability. The analyses of the secure versions of the proposed protocols are discussed with respect to the above security requirements.**

Index Terms — **Ad hoc networks, routing protocols, security, wireless systems, mobile routing.**

1. Introduction

In the next generation of wireless communication systems, there is a tremendous need for the rapid deployment of independent mobile users. Significant examples include emergency search/rescue missions, disaster relief efforts, mine site operations, battlefield military operations, electronic classrooms, conferences, convention centers, etc. [40]. A network of such users is referred to as Mobile Ad hoc Network (MANET). Such a network does not have any fixed infrastructure (i.e., no base stations/ routers); nodes arbitrarily change their positions resulting in a highly dynamic topology [1, 11, 49] causing wireless links to be broken and re-established on-the-fly.

Routing in ad hoc networks has been an active research area and in recent years numerous routing protocols have been introduced for MANETs [32, 4, 22, 10]. The deployment of such networks still faces challenges, such as limited physical security, node mobility, and limited resources (i.e., processor, power, bandwidth, storage). The major issues that affect the design, deployment, and performance of a MANET include: medium access scheme, routing, multicasting, transport layer protocol, pricing scheme, quality of service provisioning, self-organization, security, energy management, addressing and service discovery, scalability and deployment consideration [40]. The protocol design issues are inherently related to the underlying ad hoc applications. Routing protocols are designed for purposes such as quality of service provisioning, energy management and security. In this paper, we focus on security aspects of theMANET routing protocols.

The security of communication in ad hoc wireless networks is important, especially in military applications. The absence of any central coordination mechanism and shared wireless medium makes MANETs more vulnerable to digital/cyber attacks than wired networks. These attacks are generally classified into two types: passive and active attacks. Passive attacks do not influence the functionality of a connection. An adversary aims to interfere in a network and read the transmitted information without changing it. If it is also possible for the adversary to interpret the captured data, the requirement of confidentiality is violated. It's difficult to recognize passive attacks because under such attacks the network operates normally. In general, encryption is used to combat such attacks. Active attacks aim to change or destroy the data of a transmission or attempt to influence the normal functioning of the network. Active attacks when performed from foreign networks are referred to as external attacks. If nodes from within the adhoc network are involved, the attacks are referred to as internal attacks.

In order to combat passive and active attacks, a secure ad hoc network is expected to meet the following different security requirements [40, 56]:

Confidentiality: Only the intended receivers should be able to interpret the transmitted data.

Integrity: Data should not change during the transmission process, i.e., data integrity must be ensured.

Availability: Network services should be available all the time and it should be possible to correct failures to keep the connection stable.

Authentication: Every transmitting or receiving node has its own signature. Nodes must be able to authenticate that the data has been sent by the legitimate node.

Non-repudiation: Sender of a message shall not be able to later deny sending the message and that the recipients shall not be able to deny the receipt after receiving the message.

Table I outlines different active attacks that have been used in the literature to study the performance of routing protocols corresponding to above described security requirments. We use these attacks along with the security requirements as a guide to review the salient passive, active, and hybrid routing protocols for MANETs.

Most of the security related research work in MANETs has been focused on addressing issues related to confidentiality

Manuscript received June 13, 2006; revised December 25, 2007.

Loay Abusalah and Ashfaq Khokhar are with the University of Illinois at Chicago (e-mail: labusa@ece.uic.edu).

Mohsen Guizani is with Western Michigan University.

Digital Object Identifier 10.1109/SURV.2008.080407.

TABLE I
ACTIVE AD HOC NETWORK ATTACKS.

Black-Hole ***(Network Layer Attack)*****:** All packets are dropped by sending forged routing packets, the attacker could route all packets for some destination to itself and then discard them, or the attacker could cause the route at all nodes in an area of the network to point "into" that area when in fact the destination is outside the area.
Wormhole ***(Network Layer Attack)*****:** Using a pair of attacker nodes A and B linked via a private network connection. Every packet that A receives from ad hoc network, A forwards through the wormhole to B, to then be rebroadcast by B, similarly, B may send all ad hoc network packets to A.
Malign (Network Layer Attack): Watchdog and path-rater are used in ad hoc routing protocols to keep track of perceived malicious nodes in a blacklist. An attacker may blackmail a good node, causing other good nodes to add that node to their blacklists, thus avoiding that node in routes.
Partition ***(Network Layer Attack)*****:** An attacker may try to partition the network by injecting forged routing packets to prevent one set of nodes from reaching another.
Detour ***(Network Layer Attack)*****:** An attacker may attempt to cause a node to use detours through suboptimal routes. Also compromised nodes my try to work together to create a routing loop.
Routing table poisoning ***(Network Layer Attack)*****:** The publication and advertisement of fictitious routes.
Packet replication ***(Network Layer Attack)*****:** The replication of stale packets, to consume additional resources such as bandwidth, etc.
Session Hijacking ***(Transport Layer Attack)*****:** One weak point is that most authentications processes are only carried out once when a session starts. An adversary could try to appear as an authentic node and hijack the session. (Transport Layer Attack)
DoS: An adversary tries to disturb the communication in a network, for example by flooding the network with a huge amount of packages. Services offered by the network are not working as usual, slow down or even stop. Ad hoc wireless Networks are more affected than wired networks, because there are more possibilities to perform such an attack. Depending on the layer an adversary starts an attack he could disturb transmissions on physical layer, manipulate the routing process on network layer or bring down important service on application level. *Jamming* ***(MAC Layer Attack)****:* An adversary sends signals with the same frequency in that a sender and receiver communicates what cause a lot of errors in the transmission.
Impersonation: An adversary fakes the identity of an authorized node, to gain access to network resources, snoops the traffic or disturb the normal functioning of the network. With a man-in-the-middle attack an adversary reads or even alter the information transmitted between two nodes, without let them know they are not talking directly with each other.

and integrity [4, 5, 12, 21, 24, 24, 28, 28, 56]. A few solutions have been proposed to address availability and trusted routing [6, 59]. In this paper we study well-known routing protocols in terms of security and identify their limitations. We also study different security augmented solutions for these protocols.

The rest of this paper is organized as follows. In Section 2, we present classification of existing routing protocols and outline the needed security characteristics to make them secure. In Section 3, we review salient reactive routing protocols, evaluate their security characteristics, and expose existing security limitations. Section 4 presents OLSR as an example case for proactive routing protocols and discusses its security aspects. In Section 5, we discuss security in hybrid routing protocols. We conclude our discussion in Section 6.

2. Secure Ad Hoc Routing Protocols

This section will discuss the revolution of the wireless ad hoc network routing protocols and the issues that affect the design, deployment, and performance of an ad hoc wireless system network. Four routing protocols were chosen for analysis and evaluation: AODV [47], DSR [32], OLSR [9]and TORA [46]. The analyses of the secure versions of these protocols are presented. The above four routing protocols have been chosen for two main reasons. First, they are considered the most popular ad hoc routing protocols. Second, many secure versions have been derived from their basic implementation.

Routing protocols for ad hoc wireless networks can be classified into three types based on the underlying routing information update mechanism employed. An ad hoc routing protocol could be reactive (on demand), proactive (table driven) or hybrid. Figure 1, shows the three types of ad hoc routing protocols and list the available routing protocols for that category as well as some of their secure versions.

Reactive routing protocols obtain the necessary path when it is required, by using a connection establishment process. They do not maintain the network topology information and they do not exchange routing information periodically.

Section 3 explores some of the existing secure reactive routing protocols. Reactive routing protocols often outperform proactive ones due to their ability to adjust the amount of network overhead created to track the mobility in the network.

In proactive routing protocols, such as DSDV [48], every node maintains the network topology information in the form of routing tables by periodically exchanging routing information. Routing information is generally flooded in the whole network. Whenever a node requires a path to a destination, it runs an appropriate path finding algorithm on the topology information it maintains. Section 4 explores some of the existing secure proactive routing protocols.

Hybrid routing protocols such as ZRP [20] and SLSP [44] combine the best features of both reactive and proactive routing protocols. For example, a node communicates with its neighbors using a proactive routing protocol, and uses a reactive protocol to communicate with nodes farther away.. In other words, for each node, nodes within certain geographical are reached using proactive routing protocols. Outside the geographical area, reactive routing protocols will be used. Section 5 explores some of the existing secure hybrid routing protocols.

Major challenges that a routing protocol designed for Ad Hoc wireless networks faces include: mobility of nodes, resource constraints, error-prone channel state, and hidden and exposed terminal problems [40].

- *Mobility*: the network topology in an ad hoc wireless network is highly dynamic due to the movement of nodes and the addition of new nodes to the network. Disruption in service may occur either due to the movement of the intermediate nodes in the path or due to the movement of the end nodes.
- *Bandwidth constraints*: in wireless networks, the capacity of

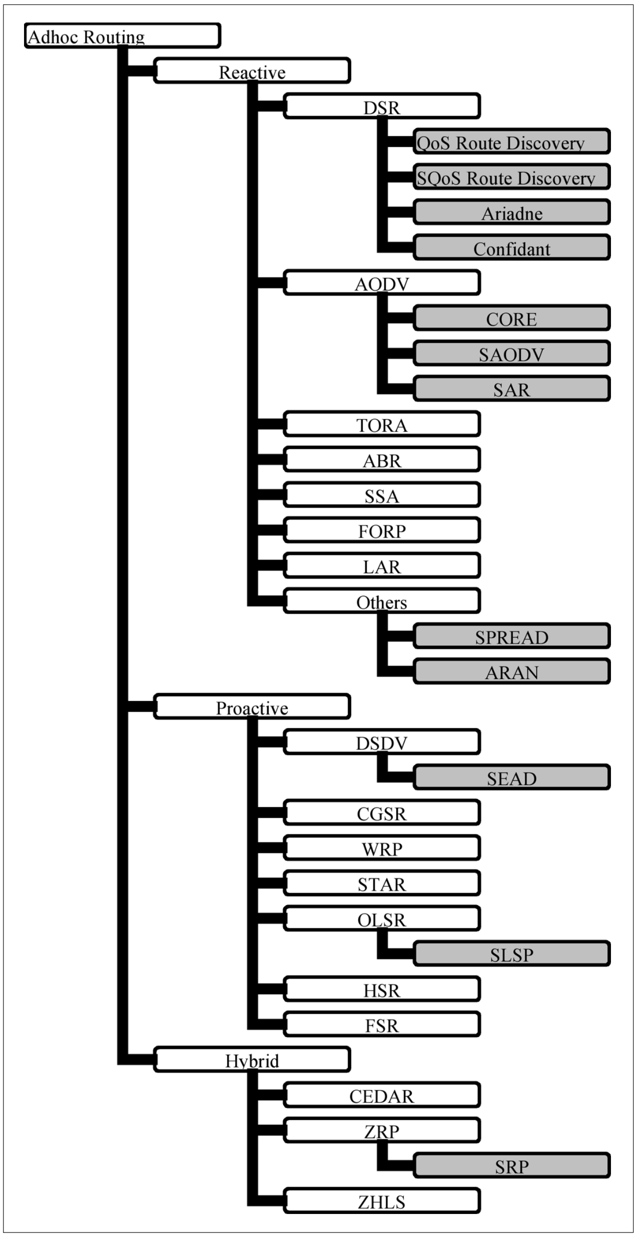


Fig. 1. Ad hoc routing protocols.

the radio band is limited and hence the data rates it can offer are much less than what a wired network can offer. That is why the routing protocol should use the bandwidth optimally to keep the overhead as low as possible.

- *Error-Prone Channel state*: the wireless links have time-varying characteristics in terms of link capacity and link-error probability. This requires that the ad hoc wireless network routing protocol should interact with the MAC layer to find alternate routes through better quality links.
- *Hidden terminal problem*: refers to the collision of packets at a receiving node due to the simultaneous transmission of those nodes that are not within the direct transmission range of the sender, but are within the transmission range of the receiver. Collations occur when both nodes transmit packets at the same time without knowing about the transmission of each other. Solution to this problem includes the use of Medium Access Collision Avoidance for Wireless MACAW [17]. This protocol requires that the receiver acknowledges each successful reception of data packet. Successful transmission is a four-way exchange mechanism, namely RTS-CTS-Data-ACK.
- *Exposed terminal problem*: refers to the inability of a node to transmit to another node when the wireless channel is not free due to transmission by the nearby transmitting node.
- *Resource constraints*: battery life and processing power are two essential and limited resources that form the major constraint for the nodes in an ad hoc network. Thus ad hoc wireless network routing protocols must optimally manage these resources.

3. Reactive Routing Protocols

Reactive routing protocols obtain the necessary path, when required, by using a connection establishment process. Such protocols do not maintain the network topology information and they do not exchange routing information periodically. In this section, we will focus on three routing protocols and some of their secure versions. First, we discuss DSR [32]. The secure versions, such as, QoS Guided Route Discovery [37], Securing Quality of Service Route Discovery [22], Ariadne [26] and CONFIDANT [4] are presented as well. Second, AODV [47] is discussed with its secure versions, CORE [39], SAODV [19]and SAR [58]. Finally, TORA [46] is discussed followed by the discussion of two ad hoc security techniques, SPREAD [36] and ARAN [51]. We focus more on reactive routing protocols because they often outperform proactive ones due to their ability to adjust the amount of network overhead created to track the mobility in the network affecting current communication.

3.1 DSR

DSR is an on-demand protocol designed to restrict the bandwidth consumed by control packets in ad hoc wireless networks by eliminating the periodic table update messages required in the proactive routing protocols. There is no *beacon* (does not require a periodic update *hello* packet, which are used by a node to inform its neighbors of its presence). Johnson [32] divided the problem of routing in to two areas, route discovery and route maintenance. In order for a node to communicate with another node in the network, it must initially discover a suitable route to use in sending packets to the destination node. As long as conditions remain unchanged, this route should then continue to work for as long as it is needed.

In the discovery procedure, the initiator transmits a *RouteRequest* packet, identifying the target to which the route is needed. Each node upon receiving the Route Request, in general, retransmits the request if it has not already forwarded a copy of the *RouteRequest*; when the target node receives the request, it returns a Route Reply to the initiator, listing the route taken by the Request, rather than forwarding the request. The target node returns a route reply for each copy of the route request that it receives. So, the source will then select a route with the lowest latency. Each *RouteRequest* packet carries a sequence number generated by the source node and the path it has traversed. A node upon receiving the *RouteRequest* packet, checks the sequence number on the packet before forwarding it. The sequence number on the packet is used to prevent loop formations and to avoid multiple transmissions of the

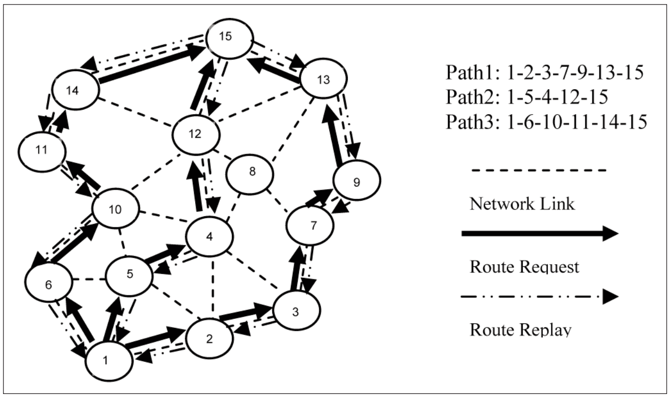


Fig. 2. Route establishment in DSR.

same *RouteRequest* packet by an intermediate node that receives it through multiple paths. For the benefit of the intermediate nodes, this protocol uses a route cache that stores all possible information extracted from the source route contained in a data packet. If an intermediate node receiving a *RouteRequest* has a route to the destination node in its route cache, it replies to the source node by sending a *RouteReply* message with the entire route information from the source node to the destination node. An exponential back-off algorithm is used to avoid frequent *RouteRequest* packets flooding in the network when the destination is in another disjoint set. DSR also allows piggy-backing of a data packet on the *RouteRequest* message so that a data packet can be sent along with the *RouteRequest* message.

In the Maintenance procedure when an intermediate node in the path moves away causing wireless link to break, a *RouteError* message is generated from the node adjacent to the broken link to inform the source node. The source node reinitiates the route establishment procedure. The cached entries at the intermediate nodes and the source node are removed when a *RouteError* packet is received. Figure 2 illustrates the working of DSR.

When node 1 wishes to send data to node 15, it first sends a *RouteRequest* message to all its neighbors. Each neighbor will check its route cache to see if the desired destination is listed. If it is there, it will send a *RouteReply* back to the sender. If it does not find a match, it will forward the *RouteRequest* to all of its neighbors. In order to avoid loops, each neighbor will check if it has already forwarded the *RouteRequest* message using the associated sequence number. The destination, node 15, will reply to all the *RouteRequest* messages it receives with a *RouteReply* message. Once the source will receive all the *RouteReply* messages within certain specific time, it will send the data through the path that has lowest hop count; in this example case, path 1-5-4-12-15. If the link between node 12 and 15 breaks, then, the upstream node (node 12) will send a *RouteError* message to the source. The source will reinitiate the *RouteRequest* message. All intermediate nodes as well as the sender will remove the tale entry from their route cache.

No security issues have been introduced in the basic DSR configuration. Also the resource management is not utilized well. For example if an intermediate node does not know the destination address, it forwards the *RouteRequest* message to all its neighbors..

3.1.1 QoS-Guided Route Discovery — Maltz, [37] introduced QoS-Guided route discovery protocol which allows a node to specify QoS metrics that must be satisfied by a discovered path. So, when a node needs to initiate a Route Request it will look first in its cache route. If the route to the destination exists, the node may choose to use it. If the flow establishment is successful, it is not necessary to perform a QoS-Guided route discovery, although it may be performed in an attempt to find a better route. The decision about whether or not to perform such a discovery may be based on resources available along a preexisting route or it may be based on the nodes' estimate of the probability of successful flow along that route. A node may choose to always perform a second search requesting a slightly higher level of resources that is available along the preexisting route.

To use this QoS-Guided route discovery mechanism, a node sending a *RouteRequest* also inserts in the Request an optional QoS Request Header for each type of resource required. Each QoS Request Header indicates the type of resource, the minimum acceptable resource level and the resource level of the current path. The resource level of the current path is initialized to the desired resource level, but may be reduced as the *RouteRequest* message traverses the network. A node receiving a *RouteRequest* message containing one or more QoS Request Headers processes each QoS Request Header to determine whether or not the node can support a new flow with resources at a level at least equal to the minimum requested. If it is unable to support the minimum requested resource level for any requested resource, the node silently discards the *RouteRequest* message. If it is unable to support the current level specified in any QoS Request Header in the packet, the node setting the current level equal to the maximum resource level it can support, and then forwards the Route request normally.

A node that is able to support the current level specified in all the QoS Request Headers contained in the packet forwards the *RouteRequest* packet normally without modifying the QoS Request Header. Maltz [38] is using the three traditional QoS metrics, bandwidth, latency and jitter. With the bandwidth metric, a node forwarding a packet updates the current resource level filed with the value that is lesser of the resource level that it received and its own resource level. For example, when a node with 240 kb/s of available bandwidth receives a request with a current resource level of 640kb/s, it reduces the bandwidth level in the *RouteRequest* packet before forwarding it. For the metrics of latency and jitter, each node actually increases the latency and jitter specified in the Request, and therefore adds the local latency or jitter to the received value.

A routing protocol using QoSGuided route discovery can find suitable routes through the network. Once such a route is found the routing protocol either must reserve those resources for a flow, or it will have to use that route on a best effort basis. In order to accomplish resource reservation a path establishment and resource reservation protocol need to be employed, in which a source establishes a flow along the path computed during route discovery by sending an Establish Flow packet along the path. Each node on the path receiving the Establish Flow packet reserves the resources needed by the flow and forwards the Establish Flow packet to the next node on the path. When a node that has been forwarding traffic for a flow is no longer able to meet the QoS requirements of the flow, it sends a Flow Error packet to the source of the flow. Although flow establishment requires two additional packet types, they are specific to the routing protocol in use. In general, Establish Flow packets can be authenticated either through broadcast authentication, or through the use of pair-wise authentication using shared keys between the source and each forwarding node. One of these two types of authentication is generally required to secure other routing messages. When each node can authenticate the source, it can use policy to determine whether or not that source is authorized to reserve these resources. In addition Flow Error packets can be authenticated in the same way as Route Error packets used by on-demand routing protocols.

One important problem in QoS-Guided route discovery protocol is determining the resources available at any particular node. Another issue in QoS-Guided Route discovery is that a node should only ignore a REQUEST if it has forwarded a better REQUEST. This causes three problems: first, an intermediate node may not know which trade-offs of different QoS metrics are preferred by the source. Second, an attacker can force a node to forward a large number of RouteRequests by broadcasting a single Request multiple times, using progressively better metrics. Third, if a node forwards each better REQUEST, a large number of forwarded packets can result from a single route discovery request.

3.1.2 Securing Quality of Service Route Discovery — SQoS [22] is a secure form of QoS-Guided route discovery for on demand ad hoc network routing. SQoS relies entirely on symmetric cryptography. Symmetric cryptographic primitives are three to four orders of magnitude faster (in computation time) than asymmetric cryptography.

SQoS builds on hash chains and MW chains. Hash function is simply a one way hash function. If X is any random number, then Y = H(X); where H is the hash function and there is no way to know X if you get Y. For example, instead of storing the user's password X, the system stores only the value Y = H(X). The user identifies himself by sending X to the system; the system authenticates his identity by computing H(X) and checking that it is equal to the stored value Y. Hash chain is accomplished by applying the hash function repeatedly . That is, if we have X, then we can get H(X), H(H(X)), H(H(H(X)))… HN(X), by applying hash function N times, where N is user specified parameter. The last computed value should be known to the receiver. For authentication purposes, the sender must send the (N–1)th value of the hash chain to the destination. When the receiver receives it, it applies the hash function one time. If the result matches the stored value it authenticates the sender. Next time the sender has to send (N–2)th value of the hash function for next authentication. Because now the receiver already has the (N-1)th value, the receiver will again apply the hash function to the received value to compare it with (N–1)th value In other words, each user password is the value needed by the system to authenticate the next password.

MW chain provides instant authentication and low storage overhead. MW chain is based on one time signature. One time signature works as follows. Each node selects a private key K that is used to generate verification key V and signature S. If the node has a message to send, it will sign it using its signature S. Only nodes that have been communicated key V can read the message (note that node that has V, can not generate S). In this way we can sign each message with different S (derived from K), and verify it using either different V or in some cases the same V. MW chain has the same properties of a hash chain and has the additional property that a signature S using key K_{i+1} can be used to generate key K_i using the equation K_i = f(s, m), but can not be used to drive K_{i+1}. In other words, if node A has a private key K_{i+1}, then S = G (K_{i+1}), and V = G(S). G must be a secure one-way hash function. For example, node A is sending a message m to node B. Node A will send the message m signed with S in the format of f(s, m). B can derive Ki from knowing f(s, m). B now has the new private Key Ki, which can be used later in its communication with A.

At SQoS, the hash chain has been replaced with an MW-chain to prevent the modification of the immutable fields of the request. A node uses one MW-chain step for each route discovery, and uses the signature S from that MW-chain step to authenticate the immutable fields of *RouteRequest* message.

SQoS solves all the three problems that have been earlier identified for the simple QoS scheme by providing the source with control over which *RouteRequest* message are re-forwarded. In SQoS the initiator specifies a list of metrics of interests, such as latency and bandwidth. For each metric, the initiator indicates the maximum necessary level and minimum desirable level, the length of the hash chain and whether steps are to be divided linearly or logarithmically.

SQoS has focused on secure quality of service guarantees such as bandwidth and latency through the use of MW chain. However, SQoS did not discuss if the intermediate nodes would in fact be able to support what ever it agrees to support. Important parameters such as the node power, CPU, RAM, encryption capability, exposure to other nodes and the organizational hierarchy have not been taken in the route computation process.

3.1.3 Ariadne — Ariadne [26] is a secure on-demand routing protocol that can authenticate messages using one of three ways: shared secrets between each pair of nodes, shared secrets between communicating nodes combined with broadcast authentication, or digital signature. Ariadne is based on TESLA, which is an efficient broadcast authentication scheme that requires loose time synchronization.

Ariadne consists of two steps: the first is to verify the authenticity of the route check and the second is to verify that there is no node missing on the *RouteRequest* (or *RouteError*)

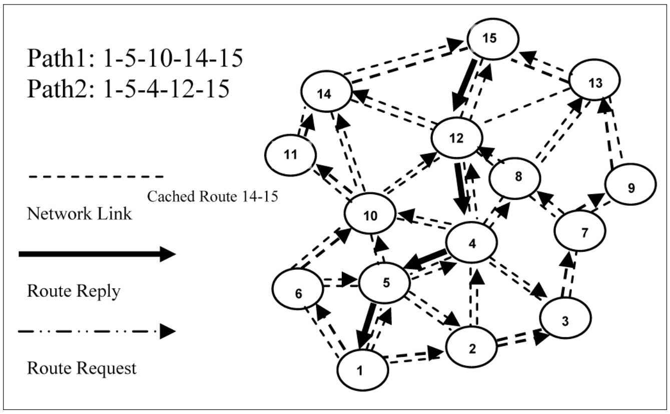


Fig. 3. Route establishments in AODV.

message. For the authentication step, the initiator simply indicates a Message Authentication Come (MAC) computed with the Ksd (shared key) over a unique data, for example, a time stamp. Then the target can easily verify the route requests authenticity and freshness using the shared key Ksd. Ariadne then uses a per hope hashing to verify that no hop was omitted.

Ariadne is vulnerable to an attacker that happens to be along the discovered route. The node can not determine whether intermediate nodes are in fact forwarding packets that they have been requested to forward. So, there is no feedback (past history) of how the intermediate nodes are behaving (even that DSR itself is based on past history through including the full route through the route request, hop-by-hop) [27].

3.1.4 CONFIDANT — CONFIDANT [4] is a secure on demand routing protocol for making misbehavior nodes unattractive for other nodes to communicate with. It is based on selective altruism and utilitarianism. It aims at detecting and isolating misbehaving nodes, thus making it unattractive to deny cooperation. Trust relationships and routing decisions are based on experienced, observed or reported routing and forwarding behavior of other nodes. The design of CONFIDANT assumes that the network layer is based on DSR. CONFIDANT consists of the following components: the monitor, the reputation system, the path manager, and the trust manager. Each component takes its function from its name. The monitor is for the neighborhood nodes to record (by listening to other communication) communication between other nodes. The trust manager deals with the incoming and outgoing ALARM messages. ALARM messages are sent by the trust manager of a node to warn others of malicious nodes. The reputation system is mainly used to avoid a centralized rating, local rating lists and/or black lists maintained at each node and potentially exchanged with friends. Similar reputation systems are used in some online auctioning systems. They provide a means of obtaining a quality rating of participants of transactions by having both the buyer and the seller give each other feedback on how their activities were perceived and evaluated. Path manager performs the following functions: i) path re-ranking according to security metric, e.g. reputation of the nodes in the path, ii) deletion of paths containing malicious nodes, iii) action on receiving a request for a route from a malicious node, e.g. ignore, do not send any reply, and iv) action on receiving request for a route containing a malicious node in the source route, e.g. ignore, alert the source.

When the monitor detects an anomaly, it informs the reputation system to take an action, which maintains a local ratings list. These lists are potentially exchanged with other nodes; the trust monitor handles input from other nodes. If a list is received from a highly trusted node, the receiver can directly place information from the list into its local ratings list. On the other hand if a list is received from an un-trusted source, the receiver can completely ignore it or give it substantially less weight than a list received from a more trusted node. Finally, the path manager chooses paths from the node's route cache based on a blacklist and the local ratings list. The path manager also specifies the reaction to a REQUEST from a node on the blacklist or to a REQUEST that has traversed a node on the blacklist.

CONFIDANT maintains global reputation values. Each node maintains a single reputation value for every other node with which it interacts, where this value combines all the various functional reputation values. Using global reputations may lead to several other issues [34]. In particular, a global reputation value may enable a node to hide bad behavior with respect to one function by correctly supporting another function. Global reputation values, therefore, do not reveal the importance placed on different services by different nodes. The distributed nature of the mechanism can lead to several inconsistencies in the reputation value. It can also lead to possible attacks on the reputation value such as advertising false high rating or false low rating about another node and negative discrimination (a node refuses services to only some nodes). In general, a simple local reputation mechanism will be more efficient than a complex reputation mechanism.

3.2 AODV

AODV [47] is very similar to DSR. AODV works by sending a *RouteRequest* message to the destination. The source node and the intermediate nodes store the next hop information corresponding to each flow for data packet transmission. The major difference between AODV and other on demand routing protocols is that it uses a destination sequence number (DesSeqNum) to determine an up to date path to the destination. A node updates its path destination only if the DesSeqNum of the current packet received is greater than the last DesSeqNum stored at the node. The *RouteRequest* message carries six items: the source identifier, destination identifier, source sequence number, destination sequence number, broadcast identifier and time to live.

In Figure 3, when node 1 sends a *RouteRequest* message, the intermediate nodes either forward the request or reply with *RouteReply* message, if it has a valid route to the destination. The broadcast identifier and the source ID are used together to detect if the node has received an earlier copy of the *RouteRequest* message.. The source node might get more than one reply, in which case it will determine later which message to select based on the hop count. Every node, before forwarding the packet, will store the broadcast identifier and the pre-

TABLE II
AODV SCENARIOS.

Step	Node	Action
1	Node 1	*Source node, destination sequence number= 3, source sequence number =1*
2	Node 15	*Destination node*
3	Node 1 Neighbors	*2,5, 6 (no idea about the destination), thus forward the RouteRequest to 3, 4 and 10.*
4	Node 4	*No idea about the destination*
5	Node 10	*It has a route to 15 (14-15), the destination sequence number =4*
6	Node 3	*It has a route to 15 (7-9-13-15), the destination sequence number=1*
7	Node 10 Node 3	*Reply, because (4>3),* *Does not reply (1<3)* *That means that node 3 has an older route to 15.*
8	Node 4	*Forward to 12, forward to 15, reply from 15*
9	Node 1	*Will get two routes:* *1-5-10-14-15* *1-5-4-12-15 will be selected (# of hops)*
10	Node 4, 5	*Path breaks between 4 and 5.*
11	Node 4	*RouteError to 15.*
12	Node 5	*RouteError to 1.*
13	Node 15	*Delete the route entry from its table.*
14	Node 1	*Delete the route entry from its table.*
15	Node 1	*Reinitiate path finding with new broadcast identifier and the previous destination sequence number.*

vious node number from which the request came. A timer is used then by this intermediate node to delete this entry in case no reply is received for the request. If there is a reply, the intermediate node stores again the broadcast identifier and the previous node from which the reply came.

AODV does not repair a broken path locally. When a link breaks, which is determined by observing the periodical beacons or though ACK messages, the source and the destination nodes are notified (end nodes). The source node then reestablishes the route with the destination using higher layers. Table II shows the sequence of steps for the case of node 1 intending to send a message to node 15 in the network shown in Figure 5.

It is important to recognize the main differences among the DSR and AODV. DSR is a pure on-demand Ad hoc routing protocol. AODV is essentially a combination of both DSR and DSDV [48]. It borrows the basic on-demand mechanism of Route Discovery and Route Maintenance from DSR, plus the use of hop-by-hop routing, sequence numbers and periodic beacons from DSDV.

AODV does not provide any type of security. *Also the resource management is not utilized well. For example if the intermediate node does not know the destination address, it will forward the* RouteRequest *to all the nodes.*

3.2.1 CORE — Selfishness that causes lack of node activity can not be solved by classical security means that aim at verifying the correctness and integrity of an operation. CORE [39] suggests a generic mechanism based on reputation to enforce cooperation among the nodes of an ad hoc network to prevent selfish behavior. Each network entity keeps track of other entities collaboration using a technique called reputation. The reputation is calculated based on various types of information on each entity's rate of collaboration. Since there is no incentive for a node to maliciously spread negative information about other nodes, simple denial of service attacks using the collaboration technique itself are prevented.

Three reputation systems are used in CORE: subjective reputation, indirect reputation and functional reputation. The subjective reputation is calculated directly from the subject observation. A subjective reputation (direct observation) at time t from the point of view of subject s is calculated using a weighted mean of the observation's rating factors, giving more relevance to the past observations. Indirect reputation reflects the value given to the final reputation by the characteristics of the complex societies. Functional reputation is used to apply a function f (which could be a forwarding function, packet function, or any other function) to the subjective reputation value or/ and the indirect value. The function reputation may apply more than one function to the same input and use a third function to get a final functional value.

CORE consists of three components: network entity, reputation table and the watchdog mechanism. The network entity comprises of the mobile nodes in the network. Each node is enriched with a set of Reputation Tables (RT) and a Watchdog Mechanism (WD). The RT and the WD together constitute the basis of the collaborative reputation mechanism. These two components allow each entity to observe and classify entities that get involved in a request/ reply process, reflecting the cooperative behavior of the involved parts. The RT is defined as a data structure stored in each network entity. The watchdog mechanism detects misbehaving nodes.

The ambiguous collision problem due to exposed terminal may prevent node A from overhearing transmissions from node B. As Figure 4 illustrates, a packet collision occurs at node A while it is listening for node B to forward the packet. In such a case, Node A will never know if node B ever forwarded the packet. Because of this uncertainty, node A should instead continue to watch node B over a period of time.

In the receiver collision problem. Figure 5, node A can only tell whether node B has sent the packet to node C, but it cannot tell if node C has received it. If a collision occurs at node C, node A only sees that node B has forwarded the packet and assumes that C has successfully received it. Thus, node B could skip retransmitting the packet and evade detection.

False misbehavior can occur when nodes falsely report other nodes as misbehaving. A malicious node could attempt

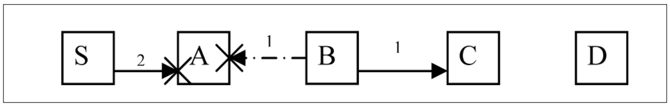


Fig. 4. Ambiguous collision.

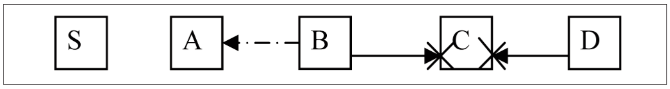


Fig. 5. Receiver collision.

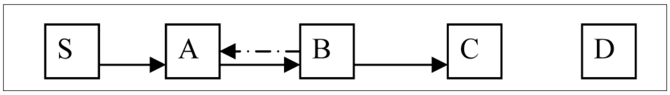


Fig. 6. Watchdog mechanism.

to partition the network by claiming that some nodes in the forwarding path are misbehaving. For instance, node A could report that node B is not forwarding packets when in fact it is. This will cause node S to mark node B as misbehaving, whereas the culprit is node A. This behavior, however, is easy to address. Since node A is passing messages onto node B (as verified by node S), then any Acknowledgments from D to S will go through node A to node S, and node S will wonder why it received replies from node D when supposedly node B dropped packets in the forward direction. In addition, if node A drops Acknowledgments to hide them from node S, the node B can detect this misbehavior and report it to D.

Another problem is that a misbehaving node that can control its transmission power can circumvent the watchdog [38]. A node could limit its transmission power such that the signal is strong enough to be overheard by the previous node but too weak to be received by the true recipient.

Multiple nodes in collusion can mount a more sophisticated attack. For example, nodes B and C from Figure 11, could collude to cause mischief. In this case, node B forwards a packet to node C but does not report to node A when node C drops the packet. Because of its limitation, it may be necessary to disallow two consecutive un-trusted nodes in a routing path.

Also, a node can circumvent the watchdog by dropping packets at a lower rate than the watchdog's configured minimum misbehavior threshold (partial dropping). Although the watchdog will not detect this node as misbehaving, this node is forced to forward at the threshold bandwidth. In this way the watchdog serves to enforce this minimum bandwidth. For the watchdog to work properly it must know where a packet should be in two hops.

3.2.1.1 Watchdog Mechanism — Many Protocols use watchdog mechanism. Watchdog mechanism has been introduced by [65]. Figure 6 illustrates the working of the watchdog mechanism. Node A can not transmits all the way to node C, but it can listen the node B's traffic. Thus when node A transmits a packet for node B to be forwarded to node C, node A can often tell if node B has transmitted the packet. If encryption is not performed separately for each link, which can be expensive, then node A can also tell if node B has tampered with the payload or the header.

The watchdog is implemented by maintaining a buffer to see if there is a match in the packets received and packets forwarded. If so, the packet in the buffer is removed and forgotten by the watchdog, since it has been forwarded on towards its final destination. If the packet has remained in the buffer for longer than a certain timeout, the watchdog increments a failure tally (RT) for the node responsible for forwarding the packet. If a tally exceeds a certain threshold bandwidth, it determines that the node is misbehaving and sends a message to the source notifying it of the misbehaving node.

The advantage of the watchdog mechanism is that it can detect misbehaving nodes at forwarding level and not just the link level. The disadvantage is that it might not detect misbehaving nodes in presence of ambiguous collusions, receiver collusions, limited transmission power, false misbehavior, collision and partial dropping [57].

3.2.2 SAODV — The black-hole attack is a killer attack for AODV. In a black hole attack a malicious node acts as an intermediate node, and advertises itself on the shortest path to the destination, which will make the sender node send all the packets through it. The malicious node will then simply drop the packets.

SAODV [19] was introduced to combat the black-hole attack. One solution is to prevent the intermediate nodes from sending a *RouteReply* message. This is still not good enough because the destination node might select a route that has the malicious node, which will then again drop all the packets. Also, by not making the intermediate node send a *RouteReply* message, the delay in the network will increase. To solve this problem the *FurtherRouteRequest* message has been introduced in SAODV. When the intermediate node sends a *RouteReply* message to the source, the source will send a quick *FurtherRouteRequest* message to the neighbors of that intermediate node (the *RouteReply* message will contain information about the next hop on the route). The neighbor node will reply with *FurtherRouteReply* message which must contain the intermediate node listed in its route (that has sent the *RouteReply* message). If it does not, then that neighbor node is a malicious node.

The approach adopted in SAODV is adequate for solving the black-hole problem but it fails to detect the wormhole attacks (when two malicious nodes works together to attack the network).

3.2.3 SAR — Security Aware Ad-Hoc Routing (SAR) protocol [58] makes use of trust levels (security attributes assigned to nodes) to make informed, secure routing decisions. Current routing protocols discover the shortest path between two nodes. But SAR can discover a path with desired security attributes (e.g., a path through nodes with a particular shared key).

A node initiating route discovery sets the desired security level for the route, i.e., the required minimal trust level for nodes participating in the query/ reply propagation. Nodes at

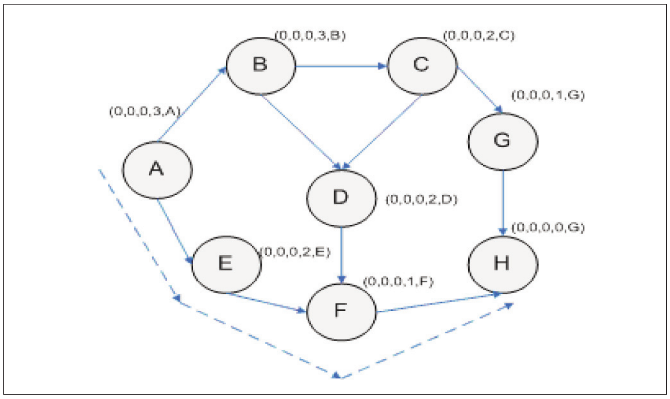


Fig. 7. Establishment of DAG for TORA.

that have the trust level share symmetric encryption keys. Intermediate nodes of different turst levels cannot decrypt in-transit routing packets or determine whether the required security attributes can be satisfied, therefore drop all such packets. . Only the nodes with the correct key can read the header and forward the packet. So if a packet has reached the destination, it must have been propagated by nodes having the same trust level.

SAR approach can be extended to any routing protocol. However, it has been presented as an extension of AODV. Most of AODV's original behavior such as on-demand discovery using flooding, reverse path maintenance and forward path setup via *RouteRequest* and *RouteReply* (RREP) messages is retained. The RREQ (Route REQuest) and the RREP (Route REPly) packets formats are modified to carry additional security information. The RREQ packet has an additional field called RQ_SEC_REQIREMENT that indicates the required security level for the route the sender wishes to discover. This could be a bit vector. An intermediate node at the required trust level, updates the RREQ packet by updating another new field, RQ_SEC_GUARANTEE field. The RQ_SEC_GUARANTEE field contains the minimum security offered in the route. This can be achieved if each intermediate node at the required trust level performs an 'AND' operation with RQ_SEC_GUARANTEE field it receives and puts the updated value back into the RQ_SEC_GUARANTEE field before forwarding the packet. Finally the packet reaches the destination if a route exists. In the RREP packet one additional field is also added. When an RREQ successfully traverses the network to the sender, the RQ_SEC_GUARANTEE represents the minimum security level in the entire path from source to destination. So the destination copies this from the RREQ to the RREP, into a new field called RP_SEC_GUARANTEE field. The sender can use this value to determine the security level on the whole path, since the sender can find routes which offer more security than asked for, with which he can make informed decisions.

A major drawback in SAR is that it involves significant encryption overhead, since each intermediate node has to perform encryption/decryption operation.. Also, the nodes are classified based on the level of trust. This creates a hierarchical trust based network. SAR evaluates the rust level of routes based only on hierarchy. This hierarchy is pre-determined and therefore implies that the trust level of the nodes is static. Furthermore, nodes can spoof each others' trust level. This can be avoided only by using tamper-proof hardware. The protocol could also create a bottleneck if most of the nodes request routes with high trust level. The protocol in general does not scale well.

3.3 TORA

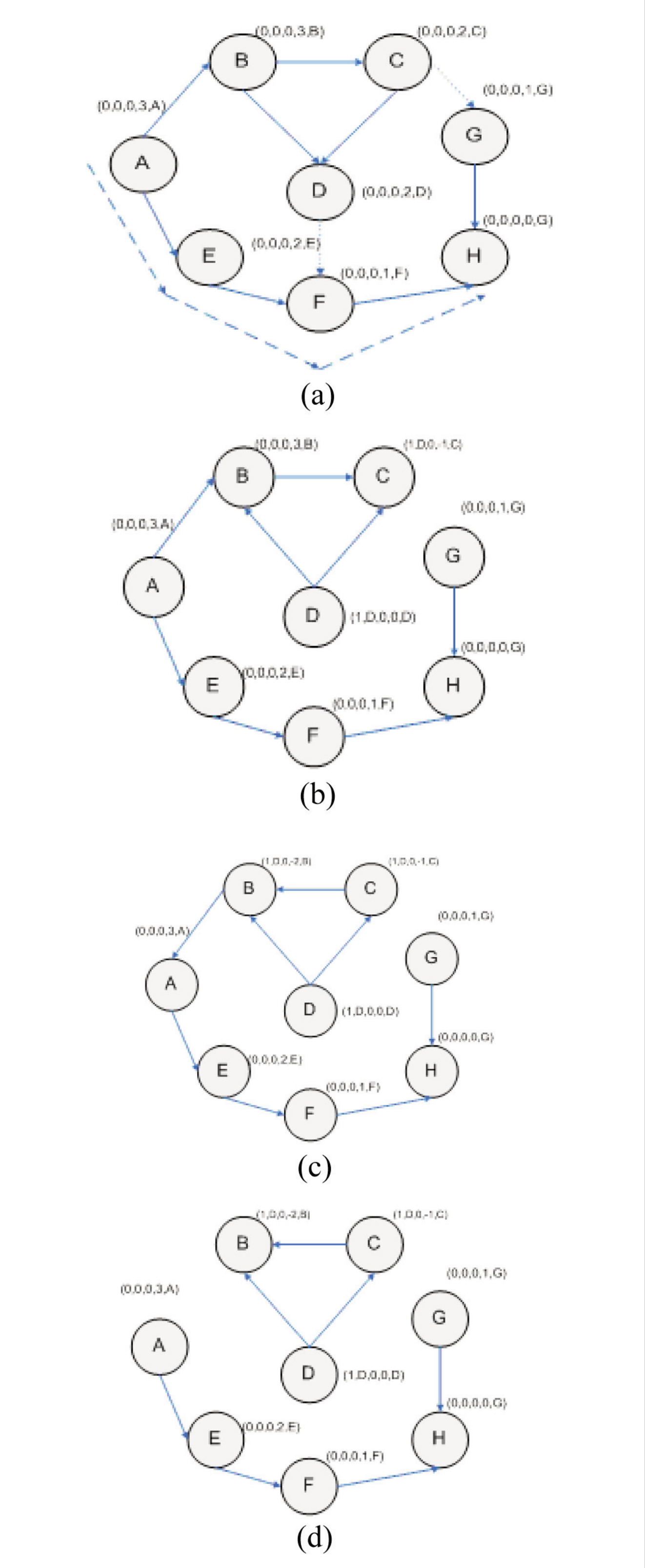
In TORA [46], routs are defined by a Directional Acyclic Graph (DAG), rooted at the destination node. To create the DAG, nodes use a height metric, consisting of five parameters: logical time of link failure, unique ID of the node defining the new reference level, reflection indicator bit, a propagation ordering parameter with respect to common reference level and unique ID of node. These five parameters are identified in parenthesis in Figure 12. Three types of control packets are used: query (QRT), update (UPD), and clear (CLR). QRT messages are flooded to all intermediate nodes until the destination node is reached and upon which a UPD message is used to update nodes along the reverse path from destination to source. UPD messages are also used to indicate link failure. A CLR broadcast is sent throughout the network to clear invalid routes. Figure 7 shows an example DAG of the connecting nodes and their heights after QRT and UPD messages have flooded the network and a path is found between nodes A and H.

In 3-dimension, it is possible to imagine the "height" of source being taller than that of the destination and the flow of data/route will be analogous to that of water flowing down from a higher to lower ground. The process of establishing the DAG is similar to the query and reply process as proposed in a Lightweight Mobile Routing (LMR) [34]. Upon link failures, route maintenance is necessary to re-establish the DAG rooted at the same destination. As shown in Figure 8 (b), the link failure at node D generates a new reference level, resulting in a propagation of reaction of link reversal, which effectively reflects the changes in adoption to the new reference. The effective new DAG is shown in Figure 8 (d) with the isolated and disconnected network consisting of nodes B, C, and D.

As timing is an important factor within the height metric, synchronization of timing is important for effectively executing TORA routing. This is sometimes achieved through an external clock source such as GPS. However, not all mobile devices are GPS enabled, and therefore, this routing protocol will pose a considerable challenge for wide spread deployment and inter-operability for heterogeneous mobile devices.

3.3.1 SPREAD — The basic idea of SPREAD [36] is to transform a secret message into multiple shares by secret sharing schemes and then deliver the shares via multiple independent paths to the destination so that even if a small number of nodes that are used to relay the message shares are compromised, the secret message as a whole in not compromised. Figure 9, shows SPREAD mechanism.

The node could make the final decision whether a message is delivered at certain time instant according to the security level and the availability of multiple paths. Also, the chosen set of multiple paths maybe changed from time to time to avoid



(a)

(b)

(c)

(d)

Fig. 8. Route Maintenance for TORA/link reversal process.

any potential capture of those multiple paths. SPREAD is a mechanism to distribute the secrecy, first by secret sharing algorithm at the source node and then by multi-path routing while shares are delivered across the network, so that in the event that a small number of shares are compromised, the secret as a whole will not be compromised.

SPREAD considers the security when massages are transmitted across the network, assuming the source and destination are trusted. SPREAD, scheme can not address the confidentiality alone. It only statistically enhances such service. For example, it is still possible for adversaries to compromise all the shares, e.g. by collusion.

3.3.2 ARAN — ARAN [51] or Authenticated Routing for Ad hoc Networks detects and protects against malicious actions by third parties and peers in an ad hoc environment. ARAN introduces authentication, message integrity, and non-repudiation. It is composed of two distinct stages. The first stage is simple and requires little extra work from peers beyond traditional ad hoc protocols. Nodes that perform the optional second stage increase the security of their route, but incur additional cost for their ad hoc peers who may not comply (e.g., if they are low on battery resources).

ARAN makes use of cryptographic certificates for the purposes of authentication and non-repudiation. Stage 1 contains a preliminary certification stage and a mandatory end to end authentication stage. It is a lightweight stage and does not demand too many resources. ARAN requires the use of a trusted certificate server T. Before entering the ad hoc network, each node requests a certificate from T. For a node A,

T -> A: CertA = [IPA, KA+, t, e] KT

The certificate contains the IP address IPA of node A, the public key KA of node A, a timestamp t of when the certificate was created, and a time e at which the certificate expires. These variables are concatenated and signed by T. All nodes must maintain fresh certificates with the trusted server and must know T's public key. The goal of Stage 1 is for the source to verify that the intended destination was reached. In this stage, the source trusts the destination to choose the return path.

Stage 2 is performed only after Stage 1 has been successfully executed. This is because the destination certificate is required in Stage 2. This stage is primarily used for discovery of shortest path in a secure fashion. Since a path is already discovered in Stage 1, data transfer can be pipelined with Stage 2)'s shortest path discovery operation.

ARAN is an on-demand protocol. Nodes keep track of whether routes are active. When no traffic has occurred on an existing route for that route's lifetime, the route is simply deactivated in the route table. Data received on an inactive route causes nodes to generate an Error (ERR) message that travels the reverse path towards the source. Nodes also use ERR messages to report links in active routes that are broken due to node movement. All ERR message must be signed. For a route between source A and destination X, a node B generates the ERR message for its neighbor C as follows:

B -> C: [ERR, IPA, IPX, CertC, NB, t] KB-

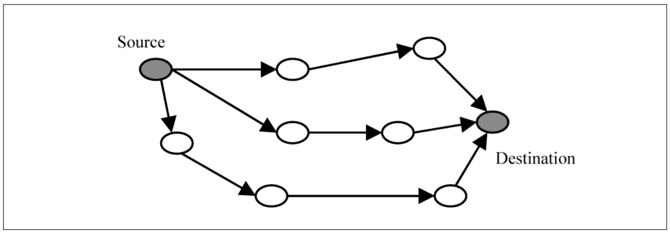


Fig. 9. SPREAD Mechanism.

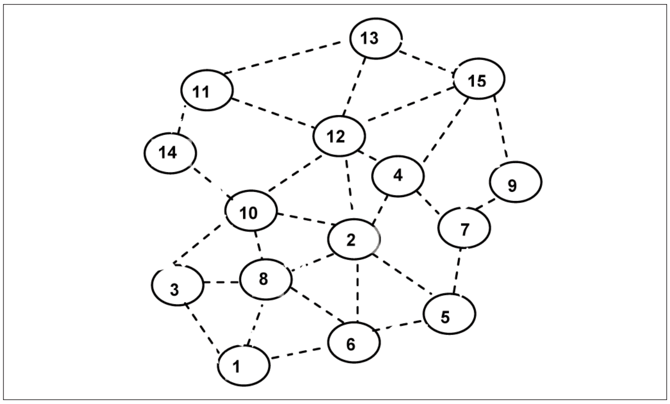


Fig. 10. Route establishment in DSDV.

This message is forwarded along the path towards the source without modification. A nonce and timestamp ensures the ERR message is fresh. Because messages are signed, malicious nodes cannot generate ERR messages for other nodes. The non-repudiation provided by the signed ERR message allows a node to be verified as the source of each ERR message that it sends. A node which transmits a large number of ERR messages, whether the ERR messages are valid or fabricated, should be avoided.

ARAN attempts a best effort key revocation that is backed up with limited time certificates. In the event that a certificate needs to be revoked, the trusted certificate server, T, sends a broadcast message to the ad hoc group that announces the revocation. Calling the revoked certificate cert r, the transmission appears as:

T -> broadcast: [revoke, CertR] KT-

Any node receiving this message re-broadcasts it to its neighbors. Revocation notices need to be stored until the revoked certificate would have expired normally. Any neighbor of the node with the revoked certificate needs to reform routing as necessary to avoid transmission through the now-un-trusted node.

This method is not failsafe. If an un-trusted node, whose certificate is being revoked, is the only link between two parts of an ad hoc network, it may not propagate the revocation message to the other part - leading to a partitioned network. To detect this situation and to hasten the propagation of revocation notices, when a node meets a new neighbor, it can exchange a summary of its revocation notices with that neighbor. If these summaries do not match, the actual signed notices can be forwarded and re-broadcasted to restart propagation of the notice.

3.4 Other Ad Hoc Reactive Routing Protocols

Location-Aided Routing protocol (LAR) [33] utilizes the location information for improving the efficiency of routing by reducing the control overhead. LAR assumes the availability of the Global Positioning System (GPS) for obtaining the geographical position information necessary for routing.

Associatively-Based Routing (ABR) [55] protocol is a distributed routing protocol that selects routes based on the stability of the wireless links. It is a beacon-based, on demand routing protocol. A link is classified as stable or unstable based on its temporal stability. The temporal stability is determined by counting the periodic beacons that a node receives from its neighbors.

Signal Stability Based Adaptive routing protocol (SSA) [15] is an on demand routing protocol that uses signal stability as the prime factor for finding stable routes. This protocol is a beacon-based protocol, in which the signal strength of the beacon is measured for determining link stability. The signal strength is used to classify a link as stable or unstable. This protocol consists of two parts: forwarding protocol (FP) and dynamic routing protocol (DRP). These protocols use an extended radio interface that measures the signal strength from beacons. DRP maintains the routing table by interacting with the DRP processes on the other hosts. FP performs the actual routing to forward a packet on its way to the destination.

Flow Oriented routing protocol (FORP) [54] is an on demand routing protocol that employs a prediction based multi hop handoff mechanism for supporting time sensitive traffic in ad hoc wireless networks. This protocol has been proposed for IPv6 based ad hoc wireless networks where quality of service (QoS) needs to be provided. The multi hop handoff is aimed at alleviating the effects of path breaks on the real time packet flows.

None of the above routing protocols has discussed security issues. LAR is aimed at reducing routing overhead, ABR focuses on establishing a stable route, SSA strives to establish stable next hop links,, and FORP supporsthe routing of time sensitive traffic.

4. Proactive Routing Protocols

In proactive or table driven routing protocols, such as DSDV [48] or OLSR [9], every node maintains the network topology information in the form of routing tables by periodically exchanging routing information. Routing information is generally flooded in the whole network. Whenever a node requires a path to a destination, it runs an appropriate path finding algorithm on the topology information it maintains.

4.1 DSDV

Destination Sequenced Distance Vector protocols (DSDV) [48] is based on Bellman-Ford shortest path algorithm. Each node has a table, which contains the shortest path to every other node in the network. These tables are constantly updated and forwarded to other nodes in the network whenever a change is

TABLE III
ROUTING TABLE FOR NODE 1.

Dest	Next Node	Dist	Seq No
2	8	2	12
3	3	1	23
4	8	3	37
5	6	2	49
6	6	1	67
7	6	3	111
8	8	1	128
9	6	4	134
10	8	2	155
11	8	4	167
12	8	3	170
13	8	4	173
14	8	3	182
15	8	4	185

detected. When a node receives an update it can either update the tables or hold it for a while in order to select shortest rout. Figure 10 shows an example where node 1 is the source and node 15 is the destination. The routing table of node 1, Table III, shows that the shortest route to the receiving node is through node 8 while the distance to it is 4 hops. When broken link is detected, the end node initiates a table update. The update message has "infinity" assigned to it and sequence number for that destination. When a node receives a message with infinity weight it quickly forwards it to neighboring nodes.

For example if node 11 moves to different location and communication is lost between 14 and 10, node 10 detects broken link and sets the broken link between them to infinity. The "infinity message" starts the table update process. Each update might increase or decrease the number of hops between any two nodes. In this example, the distance between 1 and 14 has increased from 3 to 5 hops.

Incremental update let wireless network be easily incorporated. The update creates lots of traffic and slows the network down. Nodes need to wait for a table update message by the same destination node creating delays. The sequence number tags are used to prevent the formation of loops, to counter the count-to infinity problem and for faster convergence.

4.1.1 SEAD — Secure Efficient Ad hoc Distance vector SEAD [24] is based on DSDV [48] and designed mainly to overcome security attacks such as DoS and resource consumption attacks. The protocol uses a one-way hash function and does not involve any asymmetric cryptographic operation.

SEAD uses authentication to differentiate between updates that are received from non-malicious nodes and malicious nodes. This minimizes resource consumption attacks caused by malicious nodes. SEAD used a one way hash function for authenticating the updates, discussed earlier in section 2.

SEAD avoids routing loops unless the loop contains more than one attacker. SEAD would not be able to overcome attacks where the attacker uses the same metric and sequence number that were used by the recent update message and sends a new routing table.

4.2 OLSR

The Optimized Link State Routing Protocol (OLSR) [9] is a proactive link state routing protocol. The details of the OLSR can be found in IETF's RFC 3626 [42]. There are two types of control messages used in OLSR: Hello message and Topology Control (TC) message .

Hello messages are used to build the neighborhood of a node and to discover the nodes that are within the vicinity of the node. These messages are also used to compute the multipoint relays of a node. OLSR uses the periodic broadcast of Hello messages to sense the neighborhood of a node and to verify the symmetry of radio links. The Hello messages are received by all one-hop neighbors, but are not forwarded. For every fixed interval, known as Hello Interval, the nodes broadcast hello messages. Hello messages also allow the nodes to discover their two-hop neighbors since a node can passively listen to the transmission of its one-hop neighbor. The status of these links with the other nodes in its neighborhood can be asymmetric, symmetric or MultiPoint Relay (MPR). A symmetric link means that connectivity is bi-directional, whereas asymmetric links are unidirectional. Given the set of one-hop and two-hops neighbors, a node can then proceed to selects its multipoint relays, which will enable the node to reach out to all the neighbors within a two-hop range. Every node k will keep a MPR selector set, which contains all the nodes that has selected node k as a MPR. Hence, node k can re-broadcast messages received only from the nodes found in the MPR selector set [50].

Topology Control (TC) messages contains the MPR selector set information of a particular node k. These TC messages are broadcast periodically, within the TC interval, to other MPRs, which can further relay the information to their MPRs. Thus, any node within a network can be accessed either directly or through the MPRs. With the neighborhood and topological information, nodes can construct the entire network routing table. Routing to other nodes is calculated using the shortest path algorithm such as Dijkstra'a algorithm. Sequence numbers are used to ensure that the routing update information is not stale. Whenever there are changes to the topology or within the neighborhood, the MPR set is re-calculated. Updates are sent to the entire network so that the routing has to be re-calculated to update the route information to each known destination in the network.

As specified above, hello messages are exchanged only between one-hop neighbors. Since the size of the MANET can be considerable, there is a need for a more efficient way of dis-

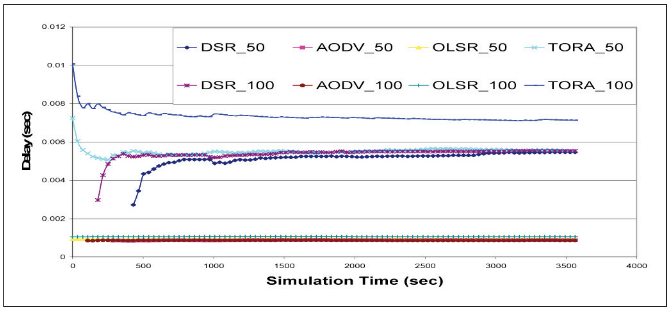


Fig. 11. Wireless delay.

seminating topological information. The traditional method would be full flooding of the network. While simple in implementation, it is not efficient since a great many control packets overheads are generated and not all are useful. MPR concept has been designed to reduce these control overheads by allowing selective flooding to occur. Only selected MPR nodes are allowed to re-broadcast topological information.

In general, the routing traffic sent and received in DSR and TORA is much better than AODV and OLSR. However, the network delay, Figure 11, in AODV and OLSR is much better than DSR and TORA.

These results are very important when designing a secure ad hoc routing protocol where the network resources are the most important factors in determining the successful of the protocol. For example, if our main goal when design a secure protocol is to have a low routing traffic, then we should start our design by some type of mechanism similar to DSR and TORA. On the other side, if our main goal (other than security) is the delay in the network, then we should think about a mechanism similar to what have been implemented in AODV and OLSR. OLSR does not provide any type of security.

4.2.1 SLSP — SLSP [44] is a secure link state protocol, which uses digital signature and one-way hash chains to ensure the security of link-state updates. SLSP is a periodic protocol that receives link-state information through a periodic Neighbor location protocol. As part of the Neighbor Lookup Protocol (NLP), each node broadcasts a signed pairing between its IP address and its MAC address. A node's NLP can notify SLSP when one MAC address uses two IP addresses, when two MAC address claim the same IP address, and when another node uses the same MAC address as the detecting node.

These protocols ensure some level of integrity of MAC and IP addresses within a two hop radius. SLSP link state updates are signed and propagated limited number of hops. In SLSP link state updates would have a maximum hop count equal to a zone radius. To ensure that an SLSP update does not travel too many hops, each update includes a hop count representing the number of hops traveled by the SLSP updates. As in SEAD and SAODV a hash chain is used to authenticate the hop count, and the hash chain values are authenticated using the hash chain's anchor, which is included in the signed portion of the SLSP link state update.

A disadvantage in SLSP is that nodes that relay or generate fewer link state updates are given priority over any node that sends more link state updates. As in SRP, an attacker can masquerade as a victim node and flood the victim's neighbors with link-state updates that appear to originate at the victim. Although the victim might be able to detect the attack, due to NLP's duplicate MAC address detection functionality, the victim will have no way to protest.

4.3 Other Ad Hoc Proactive Routing Protocols

The Cluster head Gateway Switch Routing protocol (CGSR) [8] uses a hierarchical network topology, whereas most of the other table driven routing approaches that employ flat topologies. CGSR organizes nodes into clusters, with coordination among the members of each cluster entrusted to a special node named cluster-head. This cluster head is elected dynamically by employing a least cluster change algorithm. According to this algorithm, a node ceases to be a cluster-head only if it comes under the range of another cluster head, where the tie is broken either using the lowest ID or highest connectivity algorithm.

Source-Tree Adaptive Routing protocol (STAR) [18] is a variation of table driven routing protocols with the Least Overhead Routing Approach (LORA) as the key concept rather than the Optimum Routing Approach (ORA) that was employed by earlier table driven routing protocols. The ORA protocols attempt to update the routing information quickly enough to provide optimum paths with respect to defined metric (which may be the lowest number of hops(, but with LORA, the routing protocol attempts to provide feasible paths that are not guaranteed to be optimal, but involve much less control overhead.

The Wireless Routing Protocol (WRP) [41], similar to DSDV, inherits the properties of the distributed Bellman FORD algorithm. It differs from DSDV in table maintenance and in the update procedure. While DSDV maintains only one topology table, WRP uses a set of tables to maintain more accurate information.

The Hierarchical State Routing (HSR) protocol [29] is a distributed multi-level hierarchical routing protocol that employs clustering at different levels with efficient membership management at every level of clustering. The use of clustering enhances resource a location and management.

The Fisheye State Routing (FSR) protocol [29] is a generalization of the GSR [7] protocol. FSR uses the fisheye technique to reduce information required to represent graphical data to reduce routing overhead. The basic principle behind this technique is the property of the fish's eye that can capture pixel information with greater accuracy near its eyes focal point. This property is translated to routing in ad hoc networks by a node, keeping accurate information about nodes in its local topology, and not so accurate information about far away nodes, the accuracy of the network information decreased with increasing distance.

5. Hybrid Routing Protocols

Hybrid routing protocols such as ZRP [20] and SRP [23] are protocols that combine the best features for both reactive and proactive routing protocols. For example, nodes communicate

TABLE IV
DEFENSE AGAINST ATTACKS.

Attack	Targeted Layer	Proposed Solution
Jamming	Physical and MAC	FHSS, DSSS
Wormhole	Network	Packets Leashes [25]
Blackhole	Network	[13, 59]
Byzantine	Network	[1]
Resource consumption	Network	SEAD [24]
Information Disclosure	Network	SMT [45]
Location disclosure	Network	SRP[58], NDM [16]
Routing attacks	Network	[27], SEAD [24], ARAN [51], ARIADNE [26]
Repudiation	Application	ARAN [51]
DoS	Multi-layer	SEAD [24], ARIADNE [26]
Impersonation	Multi-layer	ARAN [51]

with their neighbors using proactive routing protocols and communicate with far distance nodes using reactive routing protocols.

5.1 ZRP

Zone Routing Protocol (ZRP) [20] is a hybrid of both proactive and reactive ad hoc routing protocols. Every node has an intra-zone mechanism and extra-zone mechanism. When the node wants to work in the intra-zone, it will communicate using any proactive ad hoc routing protocol, such as DSDV. When the node wants to communicate outside the intra-zone (which is the extra-zone), it will use one of the reactive ad hoc routing protocol, such as DSR or AODV..

5.1.1 SRP — SRP [23] does not attempt to secure Route Error packets but instead delegates the route maintenance function to the secure route maintenance portion of the Secure Manager Transmission protocol. SRP uses a sequence number in the Request to ensure freshness, but this sequence number can only be checked at the target. SRP requires a security association only between communicating nodes and uses this security association just to authenticate Route Requests and Route Replies through the use of message authentication codes. At the target, SRP can detect modification of the Route Request, and at the source, SRP can detect modification of the Route Reply.

SRP does not attempt to prevent unauthorized modification of fields that are ordinarily modified in the course of forwarding these packets. For example, a node can freely remove or corrupt the node list of a Route Request packet that it forwards.

Because SRP requires a security association only between communicating nodes, it uses extremely light-weight mechanisms to prevent other attacks. For example, to limit flooding, nodes record the rate at which each neighbor forwards RouteRequest packets and gives priority to Request packets sent through neighbors that less frequently forward the Request packets. Such mechanisms can secure a protocol when few attackers are present, however, such techniques provide secondary attacks such as sending forged RouteRequest *packets to reduce the effectiveness of a node's authentic Route Requests.*

SRP does not attempt to address the route maintenance question. In SRP, as in Ariadne, multiple Replys are returned for each Request; nodes use secure message transmission (SMT) to ensure successful delivery of data packets, In SMS, data messages are split into packets using secret sharing techniques so that if M out of N such packets are received, the message can be reconstructed.

5.2 OTHER HYBRID AD HOC ROUTING PROTOCOLS CORE EXTRACTION

Distributed Ad hoc Routing (CEDAR) [52] integrates routing and support for QoS. It is based on extracting core nodes (also called as a dominator nodes) in the network, which together approximate the minimum dominating set. A dominating set (DS) of a graph is defined as a set of nodes in the graph such that every node in the graph either present in the DS or is a neighbor of some node present in the DS. There exists at least one core node within three hops. The DS of the least cardinality in a graph is called the minimum dominating set. Nodes that choose a core node as their domination node are called core member nodes of that core node.. The path between two core nodes is termed as a vertical link. CEDAR employs a distributed algorithm to select core nodes. The selection of core nodes represents the core extraction phase.

Zone based Hierarchical Link State (ZHLS) routing protocol [30] is a hybrid hierarchical routing protocol that uses the geographical location information of the nodes to form none overlapping zones. A hierarchical addressing that consists of a zone ID and a node ID is employed. A main disadvantage of this protocol is the additional overhead incurred in the creation of the zone level topology.

The definition of each attack is listed in Table IV. Each secure version of the above proposed protocols aims to solve one or more network attack through increasing the confidentiality of the network using encryption techniques. Trust as a measure of data certainty without using encryption techniques has not discussed well in the proposed protocols.

6. CONCLUDING REMARKS

In this paper we have presented different reactive, proactive and hybrid ad hoc routing protocols. The secure versions of each of the proposed protocols have also been reviewed. Traditionally, a secure ad hoc network has to meet different security requirements, Confidentiality, Integrity, Availability, Authentication and non-repudiation. Different digital attacks have been developed to undermine the security of mobile Adhoc networks. These attacks are listed in Table IV. Table IV summarizes the routing protocols in terms of proposed solutions to withstand different network attacks.

Most of the existing work has focused on confidentiality and integrity. Few works have been done on availability. In more recent works trust based routing in MANETs has gained some interest.

Trust is playing a growing security role in an open environment where unknown devices can join or leave the system at anytime. Also, due to limited processing and battery power, existing encryption based security mechanism appear too burdensome to be considered viable solutions. As defined in [14], trust is an assessment based on experience that is shared through networks of people." These shared experiences lead to trust development that augments and decays with time and frequency of interactions. Since communication is becoming pervasive, and pervasive security is called for [14], it is only natural to use the notion of pervasive trust [35] where trust relationships are ubiquitous throughout the system. Trust can be used as a measure of certainty for a given operation such as routing in a network. In a more recent work, Pirzada et al [52] has presented a comparison of trust based reactive protocols.

Trust-Aware Routing Protocol (TARP) [62] has been proposed as a secure-trusted Ad-hoc routing protocol. In TARP security is inherently built into the routing protocol where each node evaluates the trust level of its neighbors based on a set of attributes [63]. TARP trust routing mechanism is based on the basic idea of neighborhood trust where the trust-level of a node is based on its reputation among its neighbors [64].

REFERENCES

[1] B. Awerbuch *et al.*, "An On-Demand Secure Routing Resilient to Byzantine Failures," *Proc. ACM Wksp. Wireless Security 2002*, Sept. 2002, pp. 21–30.

[2] B. Bhargava *et al.*, "Trust, Privacy, and security. Summary of Workshop Breakout session," NSF Information and DATA Management (IDM) Wksp. held in Sept. 2003 in Seattle, Technical Report, CERIAS, Purdue University, West Lafayetite, IN, Dec. 2003.

[3] B. Bhargava and Y. Zhong, "Authorization Based on Evidence and Trust," *Proc. Data Warehouse and Knowledge Management Conf. (DaWaK)*, Aix-en-rovence, France, Sept. 2002.

[4] S. Buchegger and J. L. Boudec, "Performance Analysis of the CONFIDANT Protocol Cooperation Of Nodes-Fairness In Dynamic Ad-hoc Networks," *Proc. IEEE/ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHOC)*, 2002.

[5] S. Buchegger and J. Y. Boudec, "Nodes Bearing Grudges: Toward Routing Security, Fairness and Robustness in Mobile Ad Hoc Networks," *10th Euromicro Wksp. Parallel, Distributed and Network-based Proc.*, 2002.

[6] S. Buchegger and J. Boudec, "A Robust reputation System for P2P and Mobile Ad-hoc Networks," Technical paper, CH-1015 Lausanne, Switzerland, 2004.

[7] T. W. Chen and M. Gerla, "Global State Routing: A new Routing Scheme for Ad hoc Wireless Networks," Proc. IEEE ICC 1998, June 1998, pp. 171–75.

[8] C. C. Chiange *et al.*, "Routing Multi Hop Mobile Wireless Networks with Fading Channel," *Proc. IEEE SiCon 1997*, Apr. 1997, pp. 197–211.

[9] T. H. Clausen *et al.*, "The Optimized Link-State Routing Protocol, Evaluation through Experiments and Simulation," *Proc. IEEE Symp. Wireless Personal Mobile Communications 2001*, Sept. 2001.

[10] D. Coppersmith and M. Jakobsson, "Almost Hash Sequence Traversal," *Proc. 4th Conf. Financial Cryptography (FC '02)*, Lecture Notes in computer Science, 2002.

[11] S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," RFC2501, Jan. 1999.

[12] B. Dahill *et al.*, "ARAN: A secure Routing Protocol for Ad HOC Networks," UMASS Tech Report,, 2002 pp. 2–32.

[13] H. Deng, W. Li, and D. P. Agrawal, "Routing Security in Wireless Ad Hoc Networks," *IEEE Commun. Mag.*, vol. 40, no. 10, Oct. 2002, pp. 70–75.

[14] D. Denning, "A New Paradigm for Trusted Systems," *Proc. ACM New Security Paradigms Wksp.*, 1993, pp. 36–41.

[15] S. Devadas *et al.*, "Pervasive Security," Theme Panel # 3, NSF Inaugural Cyber Trust Princible Investigators Meeting and Research Directions Workshop, Baltimore, Aug. 2003.

[16] R. Dube *et al.*, "Signal Stability- Based Adaptive Routing for Ad hoc Mobile Networks," *IEEE Pers. Commun. Mag.*, Feb. 1997, pp. 36-45.

[17] A. Fasbender, D. Kesdogan, and O. Kubitz, "Variable and Scalable Security: Protection of Location Information in Mobile IP," *Proc. IEEE VTC 1996*, vol. 2, May 1996, pp. 963–67.

[18] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks," *Proc. ACM SIGCOMM '97*, 1997.

[19] J. J.Gracia-Luna-Aceves M. Spohn, "Source-Tree Routing in Wireless Networks," *Proc. IEEE ICNP 1999*, Oct. 1999, pp. 273–82.

[20] M. Guerrero Zapata, "Secure Ad hoc On-Demand Distance Vector Routing," Mobile Computing and Commun. Review, vol. 6, no. 3.

[21] Z. J. Haas, "The Routing Algorithm for Reconfigurable Wireless Networks," *Proc. ICUPC 1997*, vol. 2, Oct. 1997, pp. 562–66.

[22] R. Hauser, T., Przygienda, and G. Tsudik, "Lowering Security Overhead in Link State Routing," Computer Networks, vol. 31, no. 8, Apr. 1999, pp. :885–94.

[23] Y. Hu and D. B. Johonson, "Securing Quality-of-Service Route Discovery in On-Demand Routing for Ad Hoc Networks," *Proc. ACM SASN '04*, Oct. 20, 2004.

[24] Y. Hu and A. Perrig, "A Survey of Secure Wireless Ad Hoc Routing," *IEEE Computer Society*, 2004.

[25] Y. Hu, D. B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," *Proc. IEEE Mobile Computing Systems and Applications*, 2002, pp. 3–13.

[26] Y. Hu, A. Perrig, and D. B. Johnson, "Packet Leashes: A defense against Wormhole Attacks in Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM 2003*, vol. 3, Apr. 2003, pp. 1976–86.

[27] Y. Hu, A. Perrig, and D. B. Johonson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," *Proc. ACM MobiCom '02*, Sept. 23–26, 2002.

[28] Y. Hu, A. Perrig, and D. B. Johonson, "Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols," *Proc. ACM Wksp. Wireless Security 2003*, Sept. 2003, pp. 30–40.

[29] Y. Hu, A. Perrig, and D. B. Johonson "Efficient Security Mechanisms for Routing Protocols," *10th Annual Network and Distributed System Security Symp.*, Oct. 2003.

[30] A. Iwata *et al.*, "Scalable Routing Strategies for Ad hoc Wireless Networks," *IEEE JSAC*, vol. 17, no. 8, Aug. 1999, pp. 1369–79.

[31] M. Joa-Ng and I. T. Lu, "A Peer-to Peer Zone-Based Two Level Link State Routing for Mobile Ad hoc Networks," *IEEE JSAC*, vol. 17, no. 8, Aug. 1999, pp. 1415–25.

[32] D. B. Johnson, "Routing in Ad Hoc Networks of Mobile Hosts," *Proc. IEEE Wksp. Mobile Computing Systems and Application*s, Dec. 1994.

[33] D. B. Johnson and D. A. Maltz, "Dynamic Sources Routing in Ad Hoc Wireless Networks," *Mobile Computing*, 1996.

[34] Y. KO and N. H.Vaidya, "Location-Aided Routing (LAR) in Mobile A Hoc Networks," *Proc. ACM MOBICOM 1998*, Oct. 1998, pp. 66–75.

[35] L. Lamport, "Password Authentication with Insecure Communication," *Commun. ACM*, vol. 24, no. 11, Nov. 1981.

[36] L. Lilien, "Developing Pervasive Trust Paradiam for Authentication and Authorization," *Cracow Grid Wksp. (CGW'03)*, Cracow, Poland, Oct. 2003.

[37] W. Liu and Y. Fang, "SPREAD: Enhancing Data Confidentiality in Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM 2004*, 2004.

[38] D. A. Maltz, "Resource Management in Multi-hop Ad Hoc Networks," CMU School of Computer Science Technical Report CMU-CS-00-150, Nov. 21, 1999.

[39] S. Marti *et al.*, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *Proc. 6th Annual Int'l. Conf. Mobile Computing and Networking Table of Contents*, Boston, Massachusetts, 2000, pp. 255–65.

[40] P. Michiardi and R. Molva,, "CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks," *IFIP-Communication and Multimedia Security Conf. 2002.*

[41] S. Murthy, C. Siva Ram, and B. S. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*, Prentice Hall, Chapter 7, 2004.

[42] S. Murthy and J. J. Gracia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM Mobile Networks and Applications Journal,* Special Issue on *Routing in Mobile Communication Networks*, vol. 1, no. 2, Oct. 1996, pp. 183–97.

[43] OLSR RFC: http://www.ietf.org/rfc/rfc3626.txt?number=3626.

[44] OPNET University Program: http://www.opnet.com/services/university/

[45] P. Papadimitratos and Z. J. Haas, "Secure Link State Routing for Mobile Ad Hoc Networks," *Proc. IEEE Wksp. Security and Assurance in Ad Hoc Networks,* in Conjunction with the *2003 Int'l. Symp. Applications and the Internet*, Jan. 28, 2003.

[46] P. Papadimitratos and Z. J. Haas, "Secure Routing: Secure Data Transmission in Mobile Ad Hoc Networks," *Proc. ACM Wksp. Wireless Security 2003*, Sept. 2003, pp. 41–50.

[47] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proc. IEEE INFOCOM '97*, 1997, pp. 1405–13.

[48] C. E. Perkins and E. M. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Proc. 2nd IEEE Wksp. Mobile Computer Systems and Applications*, 1999, pp. 90–100.

[49] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *ACM SIGCOMM '94*, 1994.

[50] C. E. Perkins, *Ad Hoc Networking*, Addison Wesley Professional, Dec. 2000.

[51] P. Jacquet, P. Muhlethaler, and A. Qayyum, "Optimized Link State Routing Protocol," RFC 3626, Oct. 2003.

[52] A. A. Pirzada, C. Mcdonald, and A. Datta, "Performance Comparison of trust-based Reactive Routing Protocols," *IEEE Trans. Mobile Computing*, vol. 5, issue 6, June 2006, pp. 695–710.

[53] K. Sanzgiri *et al.*, "A Secure Routing Protocol for Ad Hoc Networks," *Proc. IEEE Network Protocols*, 2002, Nov. 12–515, 2002, pp. 78–87.

[54] P. Sinha, R. Sivakumar, V. and Bharghavan, "CEDAR: A Core Extraction Distributed Ad Hoc Routing Algorithm," *IEEE JSAC*, vol. 17, no. 8, Aug. 1999, pp. 1454–66.

[55] W. R.Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmission, and Fast Recovery Algorithm," IETF RFC 2001, Jan. 1997.

[56] Su, W., and Gerla, M., "IPv6 Flow Handoff in Ad Hoc Wireless Networks Using Mobility prediction," *Proc. IEEE Globecom 1999* Dec. 1999, pp. 271–75,.

[57] C. K. Toh, "Associatively-Based Routing for Ad Hoc Mobile Networks," *Wireless Pers. Commun.*, vol. 4, no. 2, Mar. 1997, pp. 1–36.

[58] Y. Wang, G. Attebury, and B. Ramamurthy, "A Survey of Security Issues in Wireless Sensor Networks," *IEEE Commun. Surveys & Tutorials*, vol. 8, no. 2, Apr. 2006.

[59] P. W. Yau and C. J. Mitchell, "Reputation Methods for Routing Security for Mobile Ad Hoc Networks," *IST Wksp. Mobile Future and Symp. Trends in Communications*, Bratislava, Slovakia, Oct. 2003.

[60] S. Yi, P. Naldurg, and R. Kravets, "A Security-Aware Routing Protocol for Wireless Ad Hoc Networks," *Proc. ACM MobiHoc '01*, 2001.

[61] Z. Yu *et al.*, "Risk-based Probabilistic Routing for Ad Hoc Networks," *Peer Reviewed Poster Session at Wireless Security Wksp. Conjunction with Mobicom*, Sept. 2002.

[62] L. Abusalah, A. Khokhar, and M. Guizani, "TARP: Trust-Aware Routing Protocol," *Proc. ACM Int'l. Wireless Communication and Mobile Computing Proceeding-IWCMC.*, Vancouver, Canada. August, 2006.

[63] L. Abusalah, A. Khokhar, and M. Guizani, "Trust Aware Routing in Mobile Ad Hoc Networks," *Proc. 49th Annual IEEE Globecom Int'l. Conf.*, San Francisco, California, Nov. 2006.

[64] L. Abusalah and A. Khokhar, "TARP Performance in a Mobile World," *Proc. 50th Annual IEEE Globecom Int'l. Conf.*, Washington, DC. Nov., 2007.

[65] S. Marti *et al.*, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *Proc. 6th Annual ACM/ IEEE Int'l. Conf. Mobile Computing and Networking*, 2000, pp. 255–65.

Loay Abusalah received his Bachelors of Science in Electrical Engineering from Philadelphia University in 1997 and his Master of Science in Telecommunication Engineering from the University of Jordan in 1999, and his Ph.D. in Electrical and Computer Engineering from the University of Illinois at Chicago in 2006. After receiving his Masters in 1999, he served as a Lecturer and Assessment Chair in the Computer Studies Department at Robert Morris College in Chicago. After receiving his Ph.D. in 2006, he joined the Department of Electrical and Computer Engineering at the University of Illinois at Chicago, where he currently serves at the rank of visiting Assistant Professor. He has published many technical papers in refereed conferences and journals in the areas of wireless networks, security and computer networks. His research interests include the design, analysis and implementation of network security systems, wireless routing protocols and next generation network architectures and protocols. He is serving as a reviewer for many journals and conferences such as IEEE Communication and Information Security, IEEE Survey and Tutorials, Wiley-Wireless Communications and Mobile Computing, Network Security Systems- Globecom 2006, Vehicular Technology Conference–VTC 2006, Computer and Communications Network Security Symposium, ICC 2007. He also served as the Technical Program Committee Chair for Computer and Communications Network Security Symposium, ICC and in the IEEE Communication and Information Security.

Ashfaq A. Khokhar received his M.S. in computer engineering from Syracuse University, in 1989 and Ph.D. in computer engineering from University of Southern California, in 1993. After his Ph.D., he spent two years as a Visiting Assistant Professor in the Department of Computer Sciences and School of Electrical and Computer Engineering at Purdue University. In 1995, he joined the Department of Electrical and Computer Engineering at the University of Delaware, where he first served as Assistant Professor and then as Associate Professor. In Fall 2000, He joined UIC in the Department of Computer Science and Department of Electrical and Computer Engineering, where he currently serves on the rank of Professor. He has published over 170 technical papers and book chapters in refereed conferences and journals in the areas of wireless networks, multimedia systems, data mining, and high performance computing. He is a recipient of the NSF CAREER award in 1998. His paper entitled "Scalable S-to-P Broadcasting in Message Passing MPPs" has won the Outstanding Paper award in the International Conference on Parallel Processing in 1996. He has served as the Program Chair of the 17th Parallel and Distributed Computing Conference (PDCS), 2004, Vice Program Chair for the 33rd International Conference on Parallel Processing (ICPP), 2004, and General Chair of the Workshop on Frontiers of Information Technology, 2004.

Mohsen Guizani is currently a Full Professor and the Chair of the Computer Science Department at Western Michigan University. He served as the Chair of the Computer Science Department at the University of West Florida from 1999 to 2003. He was an Associate Professor of Electrical and Computer Engineering and the director of graduate studies at the University of Missouri-Columbia from 1997 to 1999. Prior to joining the University of Missouri, he was a Research Fellow at the University of Colorado-Boulder. From 1989 to 1996, he held academic positions at the Computer Engineering Department at the University of Petroleum and Minerals, Dhahran, Saudi Arabia. He was also a Visiting Professor in the Electrical and Computer Engineering Department at Syracuse University, Syracuse, New York during academic year 1988-1989. He received his B.S. (with distinction) and M.S. degrees in Electrical Engineering; M.S. and Ph.D. degrees in Computer Engineering in 1984, 1986, 1987, and 1990, respectively, all from Syracuse University, Syracuse, New York. His research interests include Wireless Communications and Computing, Computer Networks, Design and Analysis of Computer Systems, and Optical Networking. He served/serving on the editorial boards of more than 20 journals, such as the IEEE Transaction on Wireless Communications (TWireless), IEEE Transaction on Vehicular Technology (TVT), IEEE Communications Magazine, and the Journal of Parallel and Distributed Systems and Networks. He is also the Founder and General Chair of the two International Conferences: International Wireless Conference on Wireless Communications and Mobile Computing (ACM IWCMC 2006) and Wireless Networks, Communications, and Mobile Computing (IEEE WirelessCom 2005). He is the author/co-author of six books and about 180 articles in refereed journals and conferences in the areas of wireless networking and communications, mobile computing, optical networking and network security.