ODASARA: A Novel on Demand Ant Based Security Alert Routing Algorithm for MANET in Grid Environment

R.RAMESHKUMAR,
Research Schlolar,
J.N.T.University,
Kukatpally
Hyderabad

Dr. A.DAMODARAM
Director/ U.G.C Academic Staff College,,
J.N.T.University,
Kukatpally,
Hyderabad.

Abstract
This paper proposes a novel On Demand Ant based security alert routing Algorithm (ODASARA) for mobile adhoc networks in grid environment, which combines the on-demand routing capability of Ad Hoc On-Demand Distance Vector (AODV) routing protocol with a Ant Colony Optimization mechanism using ant like mobile agents. AODV requires the actual communication to be delayed until the route is determined. This may not be suitable for real time data and multimedia communication applications. A novel On Demand Ant based Security alert Routing Algorithm provides high connectivity, reducing the amount of route discoveries before starting new connections. ODASARA enables the use of security as a negotiable metric to improve the relevance of the routes discovered by ad hoc routing protocols. This eliminates the delay before making on demand ant based security alert routing algorithm ideal for real time communication in highly dynamic networks such as MANETs. We study the performance of On Demand Ant based security alert Routing for static and dynamic network topologies.develop a two-tier classification of routing protocol security metrics, and propose a framework to measure and enforce security attributes on ad hoc routing paths. Our framework enables applications to adapt their behavior according to the level of protection available on communicating nodes in an ad hoc network. Our framework enables applications to adapt their behavior according to the level of protection available on communicating nodes in an ad hoc network in grid environment.

1. Introduction

A Mobile Ad hoc network (MANET) is a collection of wireless mobile computers forming a temporary network, which is based on radio to radio multi-hopping and has neither fixed base stations nor a wired backbone infrastructure. Routing in MANETs is a non-trivial task because hosts' movements cause frequent topology changes and require robust and flexible mechanisms to discover and maintain the routes. As mobile hosts and wireless networking equipments have become widely available, an entirely new class of applications has been created that wired network infrastructure cannot achieve. These applications include battlefield communications, disaster recovery, and rescue. These applications all rely on a quickly deployable wireless network infrastructure. One type of infrastructure is the ad hoc network, which can be rapidly deployed in a given area. Mobile ad hoc networks are collections of mobile nodes connected by wireless links. If two nodes are not within radio range, all communication between them must pass through one or more intermediate nodes that act as routers. The nodes are free to move, thus the network topology may change dynamically. Therefore, routing protocols must be able to find paths (sequences of intermediate nodes to a destination) quickly in such dynamic conditions. On-demand protocols that initiate routing activities on an on-demand basis have been widely studied because of their low routing overhead.

A substantial research effort has gone into the development of routing algorithms for MANETs. A number of routing algorithms have been proposed. Some of these are DSDV, OLSR, CGSR, AODV, DSR, TORA, ZRP, LAR and several others [1, 2, 3, and 4]. These protocols can generally be categorized as either proactive or reactive protocols. Proactive protocols build routes in the network constantly, even though there might not be packets to be transmitted between a certain set of nodes. Reactive (on-demand) protocols, on the other hand, attempt to establish multi hop between pairs of nodes only when there are packets to be exchanged between these pairs of nodes. Recently there has been great interest in so-called Swarm Intelligence [5], [6]; a set of methods to solve hard static and dynamic optimization problems using cooperative agents (usually called ants, since the method was inspired from collaborative efforts in insects). Ant-inspired routing algorithms were developed and tested by British Telecomm and NTT for both fixed and cellular networks with superior results [7, 8, 9, 10, 11, 12, 9, and 10]. Ant Net, a particular such algorithm, was tested in routing for data communication networks [7]. The algorithm performed better than OSPF, asynchronous distributed Bellman-Ford with dynamic metrics, shortest path with dynamic cost metric, Q-R algorithm and predictive Q-R algorithm [1, 7, 8, 9, 10, 11, and 12]. MANETs operate in a distributed and asynchronous

Manuscript received April 5, 2010
Manuscript revised April 20, 2010

manner. Inspired by the success of ant-agent algorithms in routing for wireless communication networks, we first proposed applications based on ideas from Ant Colony Optimization with Multi agent systems technique for Multi agent Ants based Routing in MANETs in the proposal [14]. We initiated research on these ideas since July 2004. Interest in applications of ant-based routing in MANETs has risen and several papers have appeared recently on the subject [17, 18, and 19]. For instance, Gunes has proposed an Ant-based approach to routing in MANETs in [15]. Their approach uses ants only for building routes initially and hence is a completely reactive algorithm. They have also shown some performance comparisons with other MANET routing protocols based on the pause time of mobile nodes. Marwaha [16] has explored a hybrid approach using both AODV and Ant-based exploration.

The conventional routing protocols for mobile wireless ad hoc networks suffer from certain inherent shortcomings. The proactive routing schemes continuously update the routing tables of mobile nodes consuming large portion of the scarce network capacity for exchanging huge chunks of routing table data. This reduces the available capacity of the network for actual data communication. The on-demand routing protocols on the other hand launch route discovery and requires the actual communication to be delayed until the route is determined. This may not be suitable for real time data and multimedia communication applications. Ants agents can be used for efficient routing in a network and discover the topology, to provide high connectivity at the nodes. However, the ant-based algorithms in wireless ad hoc networks have certain drawbacks. In that the nodes depend solely on the ant agents to provide them routes to various destinations in the network. This may not perform well when the network topology is very dynamic and the route lifetime is small. In ant-based routing mobile nodes have to wait to start a communication, till the ants provide them with routes. In some situations it may also happen that the nodes carrying ants suddenly get disconnected with the rest of the network. This may be due to their movement away from all other nodes in the network or they might go into sleep mode or simply turned off. In such situations, the amounts of ants left for routing are reduced in the network, which leads to ineffective routing. This paper tries to overcome these shortcomings of ant routing and AODV [20] by combining them to develop a novel routing algorithm. The On Demand Ant based Multi agents Routing Algorithm is able to reduce the end-to-end delay and route discovery latency by providing high connectivity as compared to AODV. The novel On Demand Ant based Multi agents Algorithm also does not overload the available network capacity with control messages like the proactive protocols.

2. Background Description of Ant Based Routing Algorithm

2.1Ant Agents based routing protocol

Ant-based routing algorithm for MANETs has been previously explored by [32,33,34 and 40]. Ants in network routing applications are simple agents embodying intelligence and moving around in the network from one node to the other, updating the routing tables of the nodes that they visit with what they have learned in their traversal so far as shown in figure 5.1.

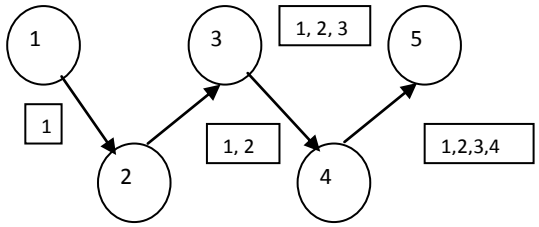

Figure2.1. Figure shows an ant traversing the network and providing routing information to nodes.

Routing ants keep a history of the nodes previously visited by them. When an ant arrives at a node, it uses the information in its history to update the routing table at that node with the best routes that it has for the other nodes in the network. The higher the history size the larger the overhead, hence a careful decision on the history size of the ants has to be made. All the nodes in the network rely on the ants for providing them the routing information, as they themselves do not run any program (protocol) for finding routes. The ant-based routing algorithm implemented in this paper does not consider any kind of communication among the ants and each ant works independently. The population size of the ants is another important parameter, which affects the routing overhead. This paper implements ants that take the "no return rule" [81] while selecting the next hop at a node. In the conventional ant algorithms the next hop is selected randomly. This is because, if the next hop selected is the same as the previous node (from where the ant came to the current node) then this route would not be optimal. Data packets sent on such routes would just be visiting a node and going back to the previous node in order to reach the destination. Every node frequent broadcasts HELLO messages to its neighbors so that every node can maintain a neighbor list, which is used for selecting the next hop by the ants.

To overcome some of the inherent drawbacks of ant routing and AODV routing protocols the proposed On Demand Ant based Multi agents technique forms a hybrid of both. The hybrid technique enhances the node connectivity and decreases the end-to-end delay and route discovery latency. In conventional ant routing techniques

route establishment is dependent on the ants visiting the node and providing it with routes. If a node wishes to send data packets to a destination for which it does not have a fresh enough route, it will have to keep the data packets in its send buffer till an ant arrives and provides it with a route to that destination. Also, in ant routing algorithms implemented so far there is no local connectivity maintenance as in AODV. Hence when a route breaks the source still keeps on sending data packets unaware of the link breakage. This leads to a large number of data packets being dropped. AODV on the other hand takes too much time for connection establishment due to the delay in the route discovery process whereas in On Demand ant based Multi agents Routing if a node has a route to a destination it just starts sending the data packets without any delay. This long delay in AODV before the actual connection is established may not be applicable in a real time communication application. On Demand Ant based Multi agents Routing Algorithm utilizes ants working independently and providing routes to the nodes as shown in figure 5.2. The nodes also have capability of launching on-demand route discovery to find routes to destinations for which they do not have a fresh enough route entry. routes to the nodes as shown in figure 5.2. The nodes also have capability of launching on-demand route discovery to find routes to destinations for which they do not have a fresh enough route entry.

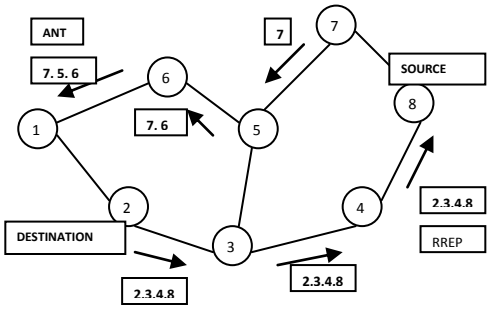

Figure 2.2. Propagation of route reply and traversal of ant in On Demand Ant agents based routing algorithm.

The use of ants with AODV increases the node connectivity (the number of destinations for which a node has un-expired routes), which in turn reduces the amount of route discoveries. Even if a node launches a RREQ (for a destination it does not have a fresh enough route), the probability of its receiving replies quickly (as compared to AODV) from nearby nodes is high due to the increased connectivity of all the nodes resulting in reduced route discovery latency. Lastly, as ant agents update the routes continuously, a source node can switch from a longer (and stale) route to a newer and shorter route provided by the ants. This leads to a considerable decrease in the average end-to-end delay as compared to both AODV and ant-based routing. Local connectivity in On Demand Ant based Multi agents Routing Algorithm is maintained in a fashion similar to AODV using route error messages (RERR). The routing table in On Demand Ant based Multi agents Routing Algorithm is common to both ants and AODV. Frequent HELLO broadcasts are used to maintain the neighbor table. This table is used to select a randomly chosen next hop (avoiding the previously visited node) from the list of neighbors by the ant.

3. Description of on Demand Ant Based Security Alert Routing Algorithm for Mobile ADHOC Networks in Grid Environment

We present a general description of our protocol and its behavior and enumerate the metrics we deploy to measure the quality of security of an ad hoc route discovered by our protocol. Originally, ad hoc routing protocols were based on modifications or augmentations to traditional routing protocols for wired networks [13]. These protocols send updates and react to topology changes, using monitoring and other infrastructure support to maintain routing tables. Current research focuses on pure on-demand[6], [5] routing protocols, and more recently, on augmentations that exploit additional information available on the ad-hoc nodes[8], [9], [14] to improve the quality of routes and reduce performance overheads. Most of the protocols that have been proposed so far focus on discovering the shortest path between two nodes as fast as possible. In other words, the length of the routes is the only metric used in these protocols. Some protocols trade performance and simplified management to obtain bounded sub-optimal paths to speed up the route discovery process[15], [16]. However, the protocol metric is still the length of the routes, measured typically as hop-count. In this paper, we contend that that there are applications that require more than just the assurance that their route has the shortest length. We argue that applications must be able to specify the quality of protection or security attributes of their ad hoc route with respect to metrics that are relevant to them. Our approach shares some similarity with the policy based routing protocols for QoS[17].

A. Protocol

For simplicity, we assume that the base protocol is an on demand protocol similar to AODV or DSR. In the original protocol, when a node wants to communicate with another node, it broadcasts a Route Request or RREQ packet to its neighbors. The RREQ is propagated to neighbors of neighbors and so on, using controlled flooding. The RREQ packets set up a reverse path to the source of the RREQ on intermediate routers that forward this packet. If any intermediate node has a path already to the RREQ destination, then this intermediate node replies with a Route Reply or RREP packet, using the reverse path to the

source. Otherwise, if there exists a route (or connectivity) in the ad hoc network, the RREQ packet will eventually reach the intended destination. The destination node generates a RREP packet, and the reverse path is used to set up a route in the forward direction.In *ODASARA* , we embbed our security metric into the RREQ packet itself, and change the forwarding behavior of the protocol with respect to RREQs. Intermediate nodes receive an RREQ packet with a particular security metric or trust level. SAR ensures that this node can only process the packet or forward it if the node itself can provide the required security or has the required au thorization or trust level. If the node cannot provide the required security, the RREQ is dropped. If an end-to-end path with the required security attributes can be found, a suitably modified RREP is sent from an intermediate node or the eventual destination. SAR can be implemented based on any on-demand ad-hoc routing protocol with suitable modification. In this paper, we use AODV[5] as our platform to implement *ODASARA* .

B. Behavior

Our modification to the traditional ad hoc routing protocol changes the nature of the routes discovered in an ad hoc network. The route discovered by *ODASARA* between two communicating entities may not be the shortest route in terms of hop-count. However *ODASARA* is able to find a route with a quantifiable guarantee of security. If one or more routes that satisfy the required security attributes exist, SAR will find the shortest such route. If all the nodes on the shortest path (in terms of hop count) between two nodes can satisfy the security requirements, *ODASARA* will find routes that are optimal. However, if the ad hoc network does not have a path with nodes that meet RREQ's security requirements, SAR may fail to find a route even if the network is connected.

C. Protocol Metrics

In this subsection, we enumerate different techniques to measure or specify the quality of security of a route discovered by our generalized SAR protocol. The first technique is the explicit representation of trust levels using a simple hierarchy that reflects organizational privileges. The next subsection enumerates the different techniques used to protect the integrity of routing messages in fixed-routing protocols.

C.1 Trust Hierarchy

ODASARA provides applications the ability to incorporate explicit trust levels into the route discovery process. Most organizations have an internal hierarchy of privileges. For example, in our battlefield scenario, the military ranks of the users of the ad hoc nodes form an explicit partial-ordering of privilege levels. A simple way of incorporating trust levels into ad hoc networks is to mirror the organizational hierarchy, and associate a number with each privilege level. These numbers represent the security/importance/capability of the mobile nodes and also of the paths. Simple comparison operators can sort these levels to reflect their position in the actual hierarchy. Another alternative is to use what we call the QoP (Quality of Protection) bit vector. For example, if mobile nodes in a network can support four different types of message protection, we can use a four bit vector to represent these message types. However, what is more important is that this trust level or protection should be immutable. A node with a lower trust level cannot arbitrarily change its trust level, or change the trust level of the RREQ request it forwards. To provide this guarantee, many techniques can be employed. If keys can be distributed a priori, or a key agreement can be reached by some form of authentication, the simplest technique is to encrypt the portion of the RREQ and RREP headers that contain the trust level. If all the nodes in a trust level share a key, then any node that does not belong to this level cannot decrypt or process the packet, and is forced to drop it. If a node is compromised, tamper-proofing can prevent attackers from learning the values of the keys. In this paper, we leverage related research in key management for ad hoc networks and assume that some mechanism to distribute keys and share secrets is already in place.

4. Protection

We develop an attack classification and itemize the protection offered by our protocol against attacks on the trust hierarchy and the information in transit in the routing protocol messages.

4.1. Trust levels

Attacks on the trust hierarchy can be broadly classified as Outsider Attacks and Insider Attacks, based on the trust value associated with the identity or the source of the attack. *ODASARA* modifies the behavior of route discovery, tying in protocol behavior with the trust level of a user. What is also needed is a binding between the identity of the user with the associated trust level. Without this binding, any user can impersonate anybody else and obtain the privileges associated with higher trust levels. To prevent this, stronger access control mechanisms are required. In order to force the nodes and users to respect the trust hierarchy, cryptographic techniques, e.g., encryption, pub- lic key certificates, shared secrets etc., can be employed. For example, all authenticated users belonging to a trust level can share a secret key.

Traditionally strong authentication schemes are used to combat outsider attacks. The identity of a user is certified by a centralized authority, and can be verified using a

simple challengeresponse protocol. Various schemes including the application of threshold cryptography [2], techniques for key sharing [19], and techniques for key agreement between multiple cooperating entities in dynamic collaborative groups [20] have been proposed to tackle the lack of a centralized authority in an ad hoc network.Our open design allows us to incorporate any of these mechanisms. For example, if one key is used per level, the trust levels are immutable and the trust hierarchy can be enforced. In our implementation, for simplicity, we use a simple shared secret to generate a symmetric encryption/decryption key per trust level. Packets are encrypted using this key and nodes and users be longing to different levels cannot even read the RREQ or RREP packets. Any user or node that is an outsider cannot obtain this key. Insider attacks are launched by compromised users within a protection domain or trust level. The users may be behaving maliciously, or their identity may be compromised (key is broken etc.). Routing protocol packets in existing ad-hoc algorithms do not carry authenticated identities or authorization credentials, and compromised nodes can potentially cause a lot of damage. Insider attacks are hard to prevent in general at the protocol level. Some techniques to prevent insider attacks include secure transient associations [21], tamper proof or tamper resistant nodes etc. For example, every time a user wants to send a RREQ, the node may require that a user re-key a password, or present her fingerprint for biometric analysis to prove her identity. If the device is lost or captured by an unauthorized user, and an attempt to send RREQs is made, this is detected by the node. The node can then destroy its keys to avoid capture (tamper proofing).

4.2. Information in Transit

In this subsection we examine specific threats to routing protocol information in transit. In addition to exploiting vulnerabilities related to the protection and enforcement of the trust levels, compromised or enemy nodes can utilize the information carried in the routing protocol packets to launch attacks. These attacks can lead to corruption of information, disclosure of sensitive information, theft of legitimate service from other protocol entities, or denial of network service to protocol entities [22]. Threats to information in transit include[23],[22],[24].

Interruption: The flow of routing protocol packets, especially route discovery messages and updates can be interrupted or blocked by malicious nodes. Attackers can selectively filter control messages and updates, and force the routing protocol to behave incorrectly. In *ODASARA* , a malicious node that interrupts the flow of packets belonging to a higher or lower trust level cannot cause an attack, because it is supposed to drop these packets in any case. If a node filters packets that belong to the same trust level as itself, the broadcast nature of the communication channel can help in detection of interruption attacks by other listeners within transmission range [3]. Interception and Subversion: Routing protocol traffic and control messages, e.g., the "keep-alive" and "are-you-up?" messages can be deflected, rerouted. In SAR, the messages are protected by the key management infrastructure. In addition, the use of flooding makes these attacks superfluous.

Modification: The integrity of the information in routing protocol packets can be compromised by modifying the packets themselves.False routes can be propagated, and legitimate nodes can be bypassed. *ODASARA* provides a suite of cryptographic techniques that can be incorporated on a need-to-use basis to prevent modification. These include digital signatures and encryption.

Fabrication: False route and metric information can be inserted into legitimate protocol packets by malicious insider nodes. In such a situation, the sender of the RREQ may receive multiple RREPs. Currently *ODASARA* picks the first RREP that arrives at the sender. The sender can be modified to verify that the RREP has credentials that guarantee the integrity of the metrics, and repudiate the ownership of attributes by challenging the intermediate nodes. We plan to incorporate this behavior in the future.

5. Implementation

In this section, we describe an implementation of *ODASARA* built as an augmentation to the AODV protocol in the Glomosim 2.02 [25] net work simulator. We retain most of AODV's original behavior.We modify the RREQ and the RREP packet formats to carry additional security information. We call our modified AODV protocol, SAODV (Security-aware AODV). In SAODV, RREQ packets have an additional field called RQ SEC REQUIREMENT that indicates the required security for the route the sender wishes to discover. This field is only set once by the sender and does not change during the route discovery phase. When an intermediate node receives a RREQ packet, the protocol first checks if the node can satisfy the security requirement indicated in the packet. If the node is secure/capable enough to participate in the routing, *ODASARA* behaves like AODV and the RREQ packet is forwarded to its neighbors. If the intermediate node cannot satisfy the security requirement, the RREQ packet is dropped and not forwarded. When an inter mediate node decides to forward the request, a new field in the RREQ packet is updated. RQ SEC GUARANTEE indicates the maximum level of security afforded by the paths discovered. This approach opens the

question of the effect of malicious nodes in networks. Since it is not uncommon to assume some mobile nodes will either be captured or compromised during the operation [2], *ODASARA* must provide a way to guarantee the cooperation of nodes. This cooperation is achieved by encrypting the RREQ headers, or by adding digital signatures and distributing keys to nodes that belong to the same level in the trust hierarchy that can decrypt these headers and re-encrypt them when necessary. The arrival of a RREQ packet at the destination indicates the presence of a path from the sender to the receiver that satisfies the security requirement specified by the sender. The destination node sends the RREP packet as in AODV, but with additional information indicating the maximum security available over the path. The value of the RQ SEC GUARANTEE field in the RREQ packet is copied to RP SEC GUARANTEE field in the RREP packet. When the RREP packet arrives at an intermediate node in the reverse path, intermediate nodes that are allowed to participate, update their routing tables as in AODV and also record the new RP SEC GUARANTEE value. This value indicates the maximum security available on the cached forward path. When a trusted intermediate node answers a RREQ query using cached information, this value is compared to the security requirement in the RREQ packet. Only when the forward path can guarantee enough security is the cached path information sent back in the RREP. In addition, *ODASARA* also has support for digital signatures. If the application requested integrity support, a new field to store the computed digital signatures was added to the RREQ.

A. Simulation Set-up

The results presented in this section are based on the simulation set up for 50 nodes moving around in 670m by 670m region. Nodes move according to the random way-point model described in [26]. The 50 nodes are classified into three levels (high, medium and low), each with 15, 15, and 20 nodes respectively. When a node sends out the RREQ, it uses its own security level as the security requirement for the route. In all measurements, the same amount of data (about 10000 packets) is sent,using the same number of flows (20), and sending at the same rate. The simulation is run until all flows complete sending.Two different traffic patterns are used to drive the simulations.Traffic pattern 1 consists of 20 CBR flows. 10% of the flows are between the high level nodes, 20% between the medium and 70% between the low level nodes. Traffic pattern 2 also has 20 CBR flows, but the distribution is 33%, 33%, 34% for the high, medium, and low level nodes. The packet size is 512 bytes, and the sending rate is 4 packets/second. The maximum number of packets in each flow is 500.

B. *ODASARA* Processing Overheads

The original AODV protocol is used as a benchmark to study the pure processing overheads of *ODASARA* . The behavior of *ODASARA* and AODV cannot be compared directly, since SAODV has larger RREQ and RREP packets compared to AODV and all the nodes participating in the route discovery must do additional processing. Initially, SAODV is configured to do trust enforcement processing, but not drop RREQ packets when required. Compared to AODV, SAODV takes 1% and 3% longer to finish with traffic patterns 1 and 2. This demonstrates that the pure overhead of adding additional processing to enable security, in the absence of dropping, is not prohibitive. We use this SAODV without RREQ dropping, *ODASARA* , as our baseline for rest of the performance measurements.

B.1 Path Discovery

Next, we ran *ODASARA* and AODV with explicit trust values, on the same traffic patterns to observe the difference in protocol behavior. The number of paths discovered by *ODASARA* and AODV, and the number of paths that violate the security requirement in *ODASARA* were recorded. Since *ODASARA* behaves like original AODV, some of the paths found violated the security requirement. This is summarized in Table II. Though *ODASARA* found more paths when the trust levels were enforced, 14 and 19 of these paths respectively were unusable. *ODASARA* discovered fewer paths, but these paths are guaranteed to obey the trust requirements of their senders.

TABLE II. NUMBER OF PATHS AND SECURITY VIOLATIONS

Traffic	1	2
Paths by *ODASARA*	94	97
Security violation by *ODASARA*	16	21
Paths by *ODASARA*	84	79

B.3 Overall Simulation Time and Transmitted Data

ODASARA security restrictions may force packets to follow longer, but more secure paths and result in taking more time to finish communication. The overhead of the protocol is illustrated in Table IV, which shows the overall time to complete transmission of all the traffic flows in both *ODASARA* with RREQ dropping and *ODASARA* , and the total amount of data transmitted. With RREQ dropping, *ODASARA* takes 2.3% and 0.2% more time to finish in traffic patterns 1 and 2 compared to *ODASARA* .Although *ODASARA* takes marginally more time to finish communication, it still finds paths in most cases and delivers almost the same amount of data from senders to the receivers.

Traffic	RREQ 1	2	RREP 1	2	1	Total 2
AODV	2245	2676	109	99	2480	2708
ODASARA	2213	1898	76	87	2413	2674

C. Secure Routing Measurements

The *ODASARA* protocol is augmented with hash digests and symmetric encryption mechanisms. The signed hash digests provide message integrity, whereas encrypting packets guarantees their confidentiality. Nodes that have the same trust level share the same encryption and decryption keys. The MD5 Hash algorithm and the Blowfish block cipher were used for these measurements. We present the measurements for Traffic Pattern 1 only, due to space constraints. The results for Pattern 2 show a similar trend. The entire RREQ packet was encrypted, with the exception of the packet-type field. The *ODASARA* protocol reflects the overhead of adding the extra field in the header. In Table V, we observe that SAODV-E (*ODASARA* with Encryption) and *ODASARA* (*ODASARA* with Signed Hash) sent fewer RREQs and RREPs than *ODASARA* . This is because nodes that were not capable of decrypting the encrypted RREQ packets, or could not verify the signatures, dropped these packets without forwarding. SAODV-E showed a 9.1% decrease and *ODASARA* showed a 17% decrease. This reinforces our claim that SAODV sends fewer control messages (RREQs and RREPs) than *ODASARA* ,though each packet needs more processing.

Traffic	Simulation time 1	2	Transmitted data 1 2
AODV	2844	2984	10041 10032
ODASARA	2989	2967	10012 10027

6. Conclusion

ODASARA enables the discovery of secure routes in a mobile ad hoc environment. Its integrated security metrics allow applications to explicitly capture and enforce explicit cooperative trust relationships. In addition, *ODASARA* also provides customizable security to the flow of routing protocol messages themselves. Routes discovered by SAR come with "quality of protection" guarantees.The techniques enabled by *ODASARA* can be easily incorporated into generic ad hoc routing protocols as illustrated by our implementation.

Reference:

[1] T. Imielinski and H. Korth, eds., "Source Routing in Ad Hoc Wireless Networks", in Mobile Computing, Kluwer Academic Publishers, Norwell, Mass., 1996, pp.153-181.

[2] J. Broch, D. Maltz, D. Johnson, Y. Hu and J. Jetcheva, "A Performance Comparison of Multi Hop Wireless AdhocNetwork Routing Protocols", Carnegie Mellon MONARCH Project, October 1998, http://www.monarch.cs.cmnu.edu/.

[3] C.E.Perkins and E.M.Royer, "Ad-Hoc On Demand Distance Vector Routing", Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications WMCSA), February 1999.

[4] C.E.Perkins and E.M.Royer, "Ad-Hoc On Demand Distance Vector Routing", Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), February 1999.

[5] E. Bonabeau, M. Dorigo and G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, 1999.

[6] M. Dorigo and G. DiCaro, "Ant Colony Optimization: a New Meta-Heuristic", Proc. 1999 Congress on Evolutionary Computation, July 6-9, 1999, pp. 1470- 1477.

[7] G. DiCaro and M. Dorigo, "AntNet: A Mobile Agents Approach to Adaptive Routing", Technical Report IRIDIA/97-12, Universite Libre de Bruxelles, Belgium.

[8] "GloMoSim Scalable Mobile Network Simulator (2000),"- UCLA Parallel Computing Laboratory, Software Package, [Online]. Available: http://pcl.cs.ucla.edu/projects/glomosim/

[9] "Parsec, Parallel Simulation Environment for Complex Systems(1998)," - UCLA Parallel Computing Laboratory,Software Package,[Online]. Available: http://pcl.cs.ucla.edu/projects/parsec/

[10] Websites:http://pcl.cs.ucla.edu/projects/parsec/samples and http://pcl.cs.ucla.edu/projects/parsec/

[11] E. M. Royer and C-K Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks,"*IEEE Personal Communications*,Apr. 1999.

[12] L. Zhou and Z. J. Haas, "Securing Ad Hoc Networks,"*IEEE Network Magazine*, Nov. 1999.

[13] S. Marti and T. Giuli and K. Lai and M. Baker, "Mitigating Routing Misbehavior in Mobile ad hoc networks," in *The Sixth Annual ACM/IEEEInternational Conference on Mobile Computing and Networking*, Boston,MA, USA, Aug. 2000.

[14] Y. Zhang and W. Lee, "Intrusion Detection in Wireless Ad-Hoc Networks," in *The Sixth Annual ACM/IEEE Conference on Mobile Computingand Networking*, Boston,MA, USA, Aug. 2000.[5]

[15] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," in*The Second IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, USA, Feb. 1999.

[16] J. Broch and D. B. Johnson, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," IETF Internet Draft, October 1999.

[17] F Azzedin, M. Maheswaran, "Towards trust-aware resource management in Grid computing systems", Cluster Computing and the Grid 2nd IEEE/A CM International Symposium CCGRID2002,2002, Page(s): 452 -457.
[18] Jeffrey Dwoskin , Sujoy Basu[+], Vanish Talwar[+], Raj Kumar[+], Fred Kitson* and Ruby Lee "Scoping Security Issues for Interactive Grids" proceedings of the IEEE conference, Nov'2003.

Prof. R.Rameshkumar is pursuing his PhD at JNT University, Hyderabad under the guidance of Prof.Dr.A.Damodaram, Director of UGC Academic Staff College of JNT University Hyderabad. He has obtained his Bachelor Degree in Computer Science and Engineering from Mookamibigai College of Engineering (Bharathidasan University) and Master Degree in Computer Science and Engineering from Arulmigu Kalasalingam College of Engineering(M.K.University).

Dr. A. Damodaram joined as Faculty of Computer Science and Engineering in 1989 at JNTU, Hyderabad. He worked in the JNTU in various capacities since 1989. Presently he is a professor in Computer Science and Engineering Department. In his 19 years of service Dr. A. Damodaram assumed office as Head of the Department, Vice-Principal and presently he is the Director of UGC Academic Staff College of JNT University Hyderabad. He was board of studies chairman for JNTU Computer Science and Engineering Branch (JNTUCEH) for a period of 2 years. He is a life member in various professional bodies. He is a member in various academic councils in various Universities. He is also a UGC Nominated member in various expert/advisory committees of Universities in India. He was a member of NBA (AICTE) sectoral committee and also a member in various committees in State and Central Government.