International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.1, No.8, December 2012 DOI:10.15693/ijaist/2012.v1i8.26-32

Comparing secure routing protocols

Sarika S
Assistant Professor
Department of Computer Science and Engineering
Sree Narayana Gurukulam College of Engineering
Kadayiruppu, Kolenchery, Kerala

Abstract - **Security has gained a lot of importance in mobile ad-hoc network. It is difficult to design secure ad hoc network routing protocols due to the highly dynamic nature of adhoc network and due to the need to operate efficiently with limited resources. This paper presents a performance comparison of various secure routing protocols named SAODV,SEAD and SDART.These protocols provide security to different routing protocols such as AODV,DSDV and DART.The simulation results are done in order to choose the best secure routing protocol to give the highest performance when implement the secure routing protocols in the highly insecure and scalable environment**

Index terms-Mobile Adhoc networks, routing, network layer security.

I. INTRODUCTION

. A wireless mobile ad hoc network consists of an unconstrained number of networking nodes. Each node may freely roam, or remain stationary in a location for an extended period of time. In addition, each node may join the network, leave the network, or fail at any time. The nodes perform peer-to-peer communication over shared, bandwidth-constrained, error-prone, and multi hop wireless channel. In recent years, network security has received critical attention from both academia and industry. As the data network becomes more pervasive and its scale becomes larger, network intrusion and attack have become severe threats to network users. This is especially true for the emerging wireless data networks. Compared to their wired counterpart, wireless networks are prone to security attacks ranging from passive eavesdropping to active interfering. As it is even more difficult to protect network entities against the intruders in wireless environment, occasional break-ins in a large-scale mobile network are nearly inevitable over a large time period.

It is difficult to design secure ad hoc network routing protocols due to the highly dynamic nature of ad hoc network and due to the need to operate efficiently with limited resources. Existing insecure ad hoc routing protocols are often highly optimized to spread routing information and also it may not achieve scalability of the network. Expensive security mechanisms can lead to reduced routing effectiveness and may consume excessive network resources leading to many new opportunities for possible Denial-of-Service (DoS) attacks and wormhole attacks [9][10] through the routing protocol. There has been a greater focus on the subject of securing such networks in the context of increasing interest in wireless networks. Many research works and discussions has been conducted in this area. From these discussions and research works ,different security issues in the field of mobile Ad Hoc networks has arised and many papers have been written describing different proposed secure routing protocols that defend against malicious attacks that wireless network face. However, the majority of these secure routing protocols did not provide a complete solution for all the attacks .The different existing protocols were surveyed so that to choose one of the secure routing protocols according to its security-effectiveness, study it and analyze its functionality and performance

There are basically two types of security threats to a routing protocol, external and internal attackers. An external attacker can be in the form of an adversary who injects erroneous information into the network and cause the routing to stop functioning properly. The internal attacker is a node that has been compromised, which might feed other nodes with incorrect information.

Security exposures of Ad Hoc routing protocols are due to two different types of attacks: active and passive attacks. In active attacks, the misbehaving node has to bear some energy costs in order to perform some harmful operation. In passive attacks, it is mainly about lack of cooperation with the purpose of energy saving. Nodes that perform active attacks with the aim of damaging other nodes by causing network outage are considered to be malicious while nodes that perform passive attacks with the aim of saving battery life for their own communications are considered to be selfish.

Malicious nodes can disrupt the correct functioning of a routing protocol by modifying routing information, by fabricating false routing information and by impersonating other nodes. On the other side, selfish nodes can severely degrade network by simply not participating in the network operation.

In existing Ad Hoc routing protocols, nodes are trusted in that they do not maliciously tamper with the content of protocol messages transferred among nodes. Malicious nodes can easily perpetrate integrity attacks by simply altering protocol fields in order to subvert traffic, deny communication t legitimate nodes (denial of service) and compromise the integrity of routing computations in general. As a result the attacker can cause network traffic to be dropped, redirected to a different destination or to take a longer route to the destination increasing communication delays. A special case of integrity attacks is spoofing whereby a malicious node impersonates a legitimate node due to the lack of authentication in the current Ad Hoc routing protocols. The main result of spoofing attacks is the misrepresentation of the network topology that possibly causes network loops or partitioning.

The objective of this paper is to study the secure routing protocols in mobile adhoc networks.It is to make the comparison between SAODV, SEAD and SDART routing protocols, using the performance metric such as packet delivery fraction, average-end to end delay and scalability. This paper also carry out the analysis and discuss which secure routing protocol is the best between SAODV, SEAD and SDART in scalable environment.

The rest of the paper is structured as follows..Section II tells about an on demand routing protocol AODV and its operations. Section III discusses about a novel scheme for achieving security in the network by making AODV a secure routing protocol. Section IV describes about DSDV protocol and section V explains the security framework on DSDV called SEAD protocol. Section VI briefs about a scalable routing protocol DART and section VII tells about SDART,a secure routing approach upon DART.Section VIII presents the comparative study on various security parameters and section IX concludes the paper.

II. AODV PROTOCOL

Ad hoc On-Demand Distance Vector Routing [4][11] algorithm is quite suitable for a dynamic self starting network as required by users wishing to utilize networks. AODV provides loop free routes even while repairing broken links. Because the protocol does not require global periodic routing advertisements, the demand on the overall bandwidth available to the mobile nodes is substantially less than in those protocols that do necessitate such advertisements. Nevertheless it still maintains most of the advantages of basic distance vector routing mechanisms. AODV use symmetric links between neighboring nodes. It does not attempt to follow paths between nodes when one of the nodes cannot hear the other one. It uses a broadcast route discovery mechanism. However AODV relies on dynamically establishing route table entries at intermediate nodes.

In AODV, nodes that do not lie on active paths neither maintain any routing information nor participate in any periodic routing table exchanges. Further a node does not have to discover and maintain a route to another node until the two need to communicate unless the former node is offering its services as an intermediate forwarding station to maintain connectivity between two other nodes. When the local connectivity of the mobile node is of interest each mobile node can become aware of the other nodes in its neighbourhood by the use of several techniques

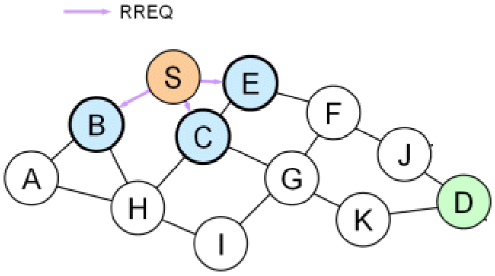

Figure 1. AODV Mechanism

When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. Fig..1 shows such a mechanism where node S would like to communicate with node D.

The node broadcasts a RREQ to find a route to the destination. S generates a Route Request with destination address, Sequence number and Broadcast ID and sent it to his neighbour nodes. Each node receiving the route request sends a route back (Forward Path) to the node. A route can be determined when the RREQ reaches a node that offers accessibility to the destination. The route is made available by unicasting a RREP back to D and is written in the routing table from S. After receiving the route reply every node has to update its routing table if the sequence number is more recent. Now node S can communicate with node D.

III . SAODV PROTOCOL

SAODV protocol[4][15] is the security integrated version of AODV.It is used to protect the routing messages of the original AODV. SAODV uses digital signatures to authenticate non-mutable fields and hash chains to authenticate the hop-count field in both RREQ and RREP messages. During the route discovery process, the source node first selects a random *seed* number and sets the Maximum Hop-count (*MHC*) value. By using a hash function *h*, the source computes the *hash* value as *h(seed*) and *Top_Hash* as *hMHC*(*seed*).When an intermediate node receives an RREQ message, it checks whether the value of *Top_Hash* is equal to *hMHCHop_Count*(*Hash*). If so, it will assume that the hop count has not been altered. Before rebroadcasting the RREQ to the neighboringnodes, the intermediate node will increment the hop-count field by one in the RREQ header and also compute the new *Hash* valueby hashing the old value (i.e., *h*(*Hash*)).Except for the hop-count field and *h*hop-count(*seed*), all otherfields of the RREQ are non-mutable and therefore can be authenticated by verifying the signature in the RREQ. When the destination node receives an RREQ, it generates an RREP in the same way. SAODV can also allow an intermediate node togenerate an RREP by using double signature extension.

IV .DESTINATION SEQUENCED DISTANCE VECTOR ROUTING (DSDV)

Destination-Sequenced Distance-Vector Routing (DSDV) [13]is a proactive routing approach for ad hoc mobile networks based on the Bellman-Ford algorithm. The main contribution of the algorithm was to solve the Routing Loop problem. Each entry in the routing table contains a sequence number, the sequence numbers are generally even if a link is present; else, an odd number is used. The number is generated by the destination, and the emitter needs to send out the next update with this number.Routing information is distributed between nodes by sending full dumps infrequently and smaller incremental updates more frequently.DSDV was one of the early algorithms available. It is quite suitable for creating ad hoc networks with small number of nodes.

DSDV requires a regular update of its routing tables,which uses up battery power and a small amount of bandwidth even when the network is idle. Whenever the topology of the network changes, a new sequence number is necessary before the network re-converges; thus, DSDV is not suitable for highly dynamic networks.

V .SEAD PROTOCOL

Secure Efficient Ad hoc Distance vector routing protocol (SEAD)[6], a secure ad hoc network routing protocol based on the design of the Destination-Sequenced Distance-Vector routing protocol (DSDV). In order to support use with nodes of limited CPU processing capability, and to guard against Denial-of-Service (DoS) attacks in which an attacker attempts to cause other nodes to consume excess network bandwidth or processing time, it uses efficient one-way hash functions and do not use asymmetric cryptographic operations in the protocol. SEAD performs well over the range of scenarios, and is robust against multiple uncoordinated attackers creating incorrect routing state in any other node, even in spite of any active attackers or compromised nodes in the network.

To reduce the number of redundant triggered updates, each node in DSDV tracks, for each destination, the average time between when the node receives the first update for some new sequence number for that destination, and when it receives the *best* update for that sequence number for it
when deciding to send a triggered update, each DSDV node delays any triggered update for a destination for this average weighted settling time, in the hope of only needing to send one triggered update, with the best metric,for that sequence number.SEAD does not use such a delay, in order to prevent attacks from nodes that might maliciously not use the delay.Since a node selects the first route it receives with highest sequence number and lowest metric, an attacker could otherwise attempt to cause more traffic to be routed through itself, by avoiding the delay in its own triggered updates.Such an attack could otherwise put the attacker in a position to eavesdrop on, modify, or discard other nodes' packets.In addition, unlike DSDV, when a node detects that its next-hop link to some destination is broken, the node does not increment the sequence number for that destination in its routing table when it sets the metric in that entry to infinity.Since higher sequence numbers take priority, this node's routing update with this new sequence number must be authenticated.For that,it makes the node flags its routing table entry for this destination to not accept any new updates for this same sequence number, effectively preventing the possible routing loop and traditional distance vector *counting to infinity* problem .

VI. DYNAMIC ADDRESS ROUTING PROTOCOL

Dynamic Address Routing Protocol (DART) [1] is a new, different, feasible way to achieve scalable ad hoc routing [2][3]. It is a proactive routing approach where each node keeps information about every other node in the network. In this approach, each node has two addresses. The routing address and the identifier of a node. The routing address of a node is dynamic and changes with node movement to reflect the node's current location in the network topology. The identifier is used to uniquely identify the node in the network. This unique number stays the same throughout the lifetime of the node. Currently IP Address is taken as the unique identifier.

A. *The Address Organization*

The addresses are organized as leaves of a binary tree. They are l bit binary numbers, ranging from $a_{l-1} \ldots a_0$. The address space is a binary address tree of $l + 1$ levels as shown in Fig 1.The leaves of the address tree represent actual node addresses; each inner node represents an address sub tree: a range of addresses with a common prefix. The links in the tree do not correspond to physical links in the network topology. The actual physical links exist between the leaf nodes.

B. *Scalability Factor*

Prefix sub graph constraint is the basic factor that provides scalability to this approach. A group of nodes form a connected sub graph by sharing a common address prefix. The distance between the nodes is less if the shared address prefix is longer. A Level-k sub tree of the address tree is defined by an address prefix of $(l-k)$ bits. For example, a Level-0 sub tree is a single address or one leaf node in the address tree. A Level-1 sub tree has a $(l-1)$-bit prefix and can contain up to two leaf nodes. A Level-2 sub tree containing addresses [000] through [011] and contain up to 8 nodes.

C .*Packet Forwarding and Routing*

When a node joins the network, it finds its dynamic address. It uses the periodic routing updates of its neighbours to identify and select an unoccupied and legitimate address. This process is known as dynamic address allocation. Usually the nodes will occupy an address in such a way to balance the network. If the first node occupies an address in one half of prefix range [3], the next incoming node will take an address in opposite prefix range. For example, in Fig 3, a joining node connecting to the node with address [00] will pick an address in the [1x] sub tree. Node A is the first node into the network and has address 00. Moreover, it takes the control of the entire 2-bit address space. When node B joins the network, it is assigned the address 10. At this time Node A can no longer assign addresses that begin with '1'.Similarly, when the third node joins the network by connecting to A, it gets assigned address 01 and when the fourth node joins via B, it gets the address 11.

In order to forward a packet, the sender node only needs to know the *identifier* of the receiver. Before sending its first packet to some destination, the sender looks up the current address of the destination node using the distributed lookup service, which maps each identifier to an address. Here, all nodes take part in the lookup table, each storing a few <*identifier*, *address*> entries. The node which stores a given <*identifier, address*> entry is called as an anchor node. A globally and priori known hash function is used here that takes an identifier as argument and returns an address where the entry can be found. If there exists a node that occupies this address, that node is the anchor node. If there is no node with that address, then the node with the least edit distance between its own address and the destination address is the anchor node .To route packets to an anchor node, a route is found to a sibling sub tree indicated by a bit in the address. If it is not found, that bit of the address is ignored, and the packet is routed to the sub tree indicated by the next (less significant) bit. When the last bit has been processed, the packet has reached its destination.

Routing is done by looking up the next hop in the routing table. To route a packet to an address, the sender node first determines the sibling sub tree to which the destination address belongs. This is done by identifying the most significant bit that differs between the current node's address and the destination's address. The process is repeated until the packet has reached its destination. Keeping the sibling sub tree information in the routing table may reduce the size of routing table and it helps to find the routing entries in a large network

VII. SDART DESIGN

SDART is a scalable and secure routing approach which detects the routing misbehaviour in the network. Some examples for routing misbehaviours are injecting malicious routing information, advertising a route with smaller distance metric, advertising routing updates with a large sequence number etc. In SDART, each node monitors the routing behaviour of its neighbours, and independently detects any malicious nodes in its own neighbourhood. This approach ensures data integrity, authentication of data and non-repudiation of important routing information in DART protocol. The approach consists of the following methods

- **Issue Token:** When a node joins the network, it is given a valid and certified token. The token is having a token number, issue time and an expiration time. Asymmetric cryptography is used to protect the tokens.
- **Local monitoring of neighbours:** The neighbouring nodes will monitor each other to find a suspicious activity. This is a distributed approach where each node is watched by the nodes in its local neighbourhood. A node is labelled as malicious only if multiple neighbours have detected a malicious activity from the same node.
- **Renew Token:** A node is allowed to be active in network activities only if it carries a valid token. Each token is having a token number, an issue time, an expiration time. If the token is about to expire, it should renew its token by sending it to its neighbours.
- **Revoke Token:** This is the process of taking back the token from a node if it is detected as malicious. Token can also be revoked if a node doesn't want to continue with the current network.

In this technique, malicious nodes are no longer allowed to sustain in a network by the distributed monitoring mechanism[5]. The nodes in the local neighbourhood may crosscheck their findings and if a node is found malicious, a notification is sent to inform all the nodes in the network.
Further, the misbehaving node is deactivated from the network by revoking its token.

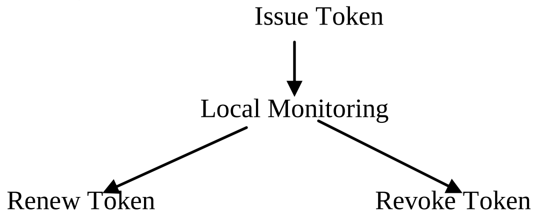

Figure2. SDART Mechanism

The deactivated tokens are stored in a list known as revoke token list which is kept with each node.
The revoked token entry is automatically deleted when the token is expired. Fig 2 shows the SDART mechanism representing issue of token, local monitoring, Renew token and Revoke token

A. Issue Token

When a node joins the network, it is issued a new token with an identifier. In DART protocol, each node is uniquely identified by a nodeID which is constant throughout the lifetime of node. In an effort to reduce the overhead associated with the addition of security measures, the node identifier can be given as token number so that the node can be identified easily. Public key cryptography is provided to ensure authentication.

B. Local Monitoring

The local monitoring mechanism [8] in SDART monitors the routing activities of each node in a distributed manner. Nodes cross check the routing updates in order to nodes discover that a node is not trustworthy. The misbehaviour of a node can be found by overhearing the channel. Fig.3 illustrates how a node can cross-check the routing updates and examine their trustworthiness.

Here node C monitors, both node A and node B. Suppose that node B previously has announced a route toward destination F with hop count as 2. Now A announces a routing update toward F with hop count as 1, claiming that its next hop is B.D can readily detect this routing misbehaviour as B has already announced a route toward F with hop count as 2. The local neighbouring nodes cross-validate their findings and collaborate with each other to reach a consensus whether a node is malicious or not. As it is a decentralized scheme[7], the scalability of the network is maintained.

C. Renew Token

Every node is provided a valid and certified token on joining the network which is having the fields token number, issue time and an expiration time. The legitimacy of the tokens is ensured by using asymmetric cryptographic primitives[14].

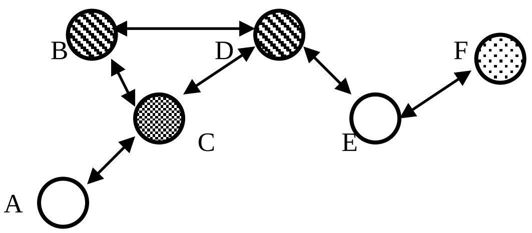

Figure 3. Routing misbehaviors

This design is following RSA based approach where each node has its own secret key and a common public key that is known to all the nodes.The RSA key pair is denoted as *(pk, sk).* The signed tokens are verified by the public key and the secret key is used to sign the tokens. RSA secret key is shared among a group of nodes and each node obtains its own part. Some of the shared pieces or all pieces are required to reproduce the original secret key. This technique is done to reduce the overhead of keeping individual keys with each node. As the secret key is shared among n nodes; no node knows the original secret key.

It is impractical to reconstruct the secret key from all the nodes. So secret key sharing is based on a technique known as threshold secret sharing [12]. A secret polynomial $p(x)$ is used to share the secret key among a set of nodes only. The number of nodes that can share the secret depends upon the degree of $p(x)$. To divide the key into n pieces, a random polynomial of degree n-1 is taken.

$$p(x)=do+dlx+\ \dots\ +dn\text{-}1x^{k-1}$$

where do=p(0) ,d1=p(1)…………..,dn=p(n).The value of d0 can also be assigned to an integer like $d0=K$. Then

$$K1=p(1),K2=p(2),\dots\dots,Kn=p(n).$$

If n number of K values are given, it is possible to find $p(x)$ using Lagrange interpolation. The coefficients of p(x) cannot be obtained if the subset of K values is less than n. This group of n nodes can sign a token together.

When the current token expires, a node should renew the same, if it wants to continue with the network operations. It sends the token to n nodes in its local neighbourhood. When a node receives a token renewal request, it checks the legitimacy of a token using the public key in network. If the token is still valid and is about to expire, it creates a new token with token number same as the old one, issue time as current time and expiration time as a time depending on the good behaviour of node in the network. Then it signs the token using its secret key part and sends the signed token to the sender node. When the requesting node receives n number of newly signed tokens, it combines these pieces into a new token.

The public key is common to all nodes in the network and it is used in an effort to verify the tokens. There is a term Revoked Token List (RTL) which is kept with each of the nodes and is having the fields token number and corresponding node identifier. The Revoked Token List (RTL) is used to verify the legitimacy of tokens. RTL contains the list of all the nodes whose token has been revoked. It maps node identifier corresponding to token number. The entries in RTL are kept until the revoked tokens are expired and are automatically deleted afterwards.

The lifetime of a node in a particular network is decided according to the opinion of nodes in its neighbourhood. A newly joined node is given a shorter lifetime. Lifetime depends on the expiration time entered by different nodes. The nodes may give expiration time during the renewal of token. If a node is behaving well in the network for a long time, it is given a longer expiration time so that the frequent token renewals can be avoided and overhead can be reduced. This overhead increase occurs due to the increased number of routing advertisements and number of control packets .In an effort to reduce the size of RTL, the new nodes are given shorter expiration. If a newly joining node is provided with a longer expiration time and is flagged as a malicious node within a short time, the node's entry is retained in RTL for a long time till it expires. This will increase the size of RTL and will result in greater overhead of maintaining longer RTL's along with each nodes.

D. Revoke Token

The token of a malicious node can be revoked only with the unanimous support of nodes in its neighbourhood. Any one or two node cannot do so. The token revocation process in SDART uses a distributed mechanism as in [5].

The nodes are monitoring each other to find the misbehaving ones .If a node suspects some routing misbehaviour on any of its neighbour, it informs the other nodes about its observation. It broadcasts a SAP (Suspicious Activity Packet) to the nodes in the network.

Parameters for comparison	**SAODV**	**SEAD**	**SDART**
Routing protocol	AODV	DSDV	DART
Routing Strategy	Reactive	Proactive	Proactive
Encryption Algorithm	Asymmetric	Symmetric	Asymmetric
Confidentiality	No	No	Yes
Authentication	Yes	Yes	Yes
Non-Repudiation	Yes	No	Yes
Scalability	No	No	Yes
Data integrity	yes	No	Yes
Cryptography scheme	Key cryptography and hashing is used	One way hash function is used	Asymmetric key cryptography is used.

Table 1.Comparison of SAODV,SEAD and SDART

If a node has received a threshold of '*n'* SAP's against the same node, the malicious nature of the suspected node is confirmed. It then creates and broadcast a token revocation notification (a TRN packet) signed by its own share of secret key. If most of the nodes reach an agreement to revoke the suspicious node, the token of the misbehaving node is revoked and its details are added to RTL of each node. If the node's details are already in the RTL, it is just neglected. Once the token is revoked, the problematic node is deprecated by the other nodes and is no longer allowed to perform in network operations.

VII.COMPARATIVE STUDY

In this section,I summarize the various secure routing protocols that have been explained above by considering several attributes .Table 1 presents the comparative study on various security parameters and try to chalk out strategy adopted by each protocol, its merits and demerits. It is clear from comparative study of these three protocols that SDART is the solution for a scalable and secure network by considering above performance parameters. SAODV is the on demand protocol which uses asymmetric RSA key pairs.It ensures authentication,data integrity and non-repudiation of information.But it fails to achieve confidentiality and scalability. SEAD acts upon DSDV guarantees only authentication but it fails in achieving other factors.

The proposed design SDART acts upon DART protocol is following RSA based approach where each node has its own secret key and a common public key that is known to all the nodes. The RSA key pair is denoted as *(pk, sk)*. The signed tokens are verified by the public key and the secret key is used to sign the tokens. RSA secret key is shared among a group of nodes and each node obtains its own part. Some of the shared pieces or all pieces are required to reproduce the original secret key. This technique is done to reduce the overhead of keeping individual keys with each node. As the secret key is shared among n nodes; no node knows the original secret key.This design is a novel approach for securing ad hoc networks ensuresconfidentiality, scalability, security, authentication, non repudiation,and data integrity. The results of comparison shows that SDART is the most feasible and promising approach for creating a secure and scalable protocol. However SDART also increases overhead due to the increased number of routing overhead packets. But still SDART is the only secure protocol that is promising for large networks.

IX. CONCLUSION

In this paper, an attempt is made to discuss various secure routing protocols and attacks that exist in adhoc networks. It is comparing SAODV, SEAD and SDART protocol.This paper suggests a new secure Ad hoc routing protocol using dynamic address routing called SDART. The goal of security in a network is to provide the routing layer with sabotage resistance. Sabotage resistance means robustness against false route advertisements, such that an attacker can only affect a limited portion of the network, over a limited time span. SDART protects the network by a distributed token based mechanism and maintains the scalability of the network. It is a feasible approach as it ensures security and reduces the overhead of maintaining large routing table. There are a number of proposed optimizations to secure protocols, which can further improve the performance of each.A comparative studyis done on the basis scalability,data integrity,authentication,non repudiation of routing information etc.From these,there are reasons to believe that SDART can be the basis for secure ad hoc routing protocols especially for massive ad hoc and mesh networks.

REFERENCES

[1] Jakob Eriksson, Michalis Faloutsos, and Srikanth Krishnamurthy, "DART: Dynamic Address RouTing for Scalable Adhoc and Mesh Networks," *IEEE/ACM transactions on Networking,* Vol 15, No. 1, February 2007.

[2] Jakob Eriksson, Michalis Faloutsos, and Srikanth Krishnamurthy, "Scalable Adhoc Routing: The case for Dynamic Addressing," in *IEEE infocom, 2004.*

[3] J. Eriksson, M. Faloutsos, and S. Krishnamurthy, "Peernet: Pushing peer-2-peer down the stack," in *IPTPS*, 2003.

[4] Yuxia Lin, A. Hamed Mohsenian Rad, Vincent W.S. Wong,, "Experimental Comparisons between SAODV and AODV Routing Protocols," *WMuNeP'05, October 13, 2005*

[5] Hao Yang, James Shu, Xiaoqiao Meng, and Songwu Lu, "SCAN: Self-Organized Network-Layer Security in Mobile Ad Hoc Networks," *IEEE journal on selected areas in communications*, vol. 24, no. 2, February 2006.

[6] Y. Hu, D. Johnson, and A. Perrig, "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks," in *Proc. IEEE WMCSA*, Jun. 2002, pp. 3–13.

[7] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for MANET," in *Proc. IEEE ICNP*, 2001, pp. 251–260.

[8] S. Capkun, L. Buttyan, and J. Hubaux, "Self-organized public-key management for mobile ad hoc networks," *IEEE Trans. Mobile Computing.*, vol 2, no. 1, pp. 52–64, Jan. 2003.

[9] Y.-C. Hu, A. Perrig, and D. B. Johnson. Packet leashes: "A defense against wormhole attacks in wireless ad hoc networks", In *Proceedings of IEEE INFOCOM*, April 2003.

[10] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Wormhole detection in wireless ad hoc networks," *Department of Computer Science, Rice University, Tech. Rep. TR01-384,* June 2002.

[11]C. Perkins, E. Royer, and S. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing," Internet Draft, draft-ietf-manet-aodv-10.txt, 2002.

[12] A. Shamir. How to Share a Secret. Communications of the ACM, 22(11):612–613, 1979

[13] Charles Perkins and Pravin Bhagwat, "Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIGCOMM'94*, 1994.

[14] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehaviour in Mobile Ad Hoc Networks," in Proc. ACM MOBICOM, 2000

[15] Manel Guerrero Zapata," Secure Ad hoc On-Demand Distance Vector Routing",in *Mobile Computing and Communications Review, Volume 6, Number 3*,2006

Author Profile

Sarika S received the **B-Tech.** degree in Information Technology from Rajiv Gandhi Institute of Technology,Kottayam, Mahatma Gandhi University,Kerala, India, in 2005 and Completed **M.E.** in Computer Science and Engineering from Karunya University, Coimbatore, India. Her research interest includes Mobile Ad hoc networks, Network security, Internet security.