Power Aware and Secure Dynamic Source Routing Protocol in Mobile Ad Hoc Networks

Mohit Miglani[1], Deepika Kukreja[2],
Sanjay Kumar Dhurandher[3], and B.V.R. Reddy[4]

[1] Associate Application Developer, CSC, India
[2] Assistant Professor, MAIT, GGSIP University, Delhi, India
[3] Associate Professor, NSIT, Delhi University, Delhi, India
[4] Professor and Dean, USICT and USET, GGSIP University, Delhi, India

Abstract. Mobile Ad Hoc Networks (MANETs) show better and valuable performance in the circumstances where the generally used wireless networks fail to work. In order to make routing in MANETs secure, number of security based routing protocols have been proposed in the literature but none of them is compliant with the MANETs environment. We propose a protocol, termed as Power aware Secure Dynamic Source Routing (PS-DSR) that makes the standard Dynamic Source Routing protocol secure by using power aware trust based approach. The monitoring operation is distributed among a few set of nodes called monitor nodes. The set of monitor nodes is selected sporadically which makes the proposed method adaptable to the two focal concerns of MANETs: dynamic network topology and energy constraint devices. The method detects malicious packet dropping and packet modification attacks. It ensures the trustworthy and authentic selection of routes by the PS-DSR protocol and improves the overall performance of the protocol in presence of malicious nodes.

Keywords: Mobile Ad Hoc Networks, DSR,security, trust, routing, attacks, power.

1 Introduction

Mobile Ad Hoc Networks (MANETs) are infrastructure less networks, in which there is no central authority and each node functions as a host as well as a router. There are two focal concerns of MANETs: dynamic network topology and energy constraint devices. MANETs have dynamic network topology which means that mobile nodes are not fixed, they are free to move, and they may leave or join the network at any time. MANETs consist of nodes that are mainly battery operated hand held devices. Battery power is a limited resource which adds energy constraint problem to MANETs. Power aware Secure Dynamic Source Routing (PS-DSR) focuses on power saving and dynamic network topology which makes it complaint with MANETs environment. There are varied routing protocols in the literature, some of them are "proactive" and the others are "reactive" protocols. Dynamic Source Routing (DSR) protocol is a reactive protocol, which

J. Lloret Mauri et al. (Eds.): SSCC 2014, CCIS 467, pp. 45–56, 2014.

means it is source initiated and it will search for routes on demand and store them in its cache. PS-DSR is a power saving trust based protocol that ensures the trustworthy and authentic selection of routes thereby enhances the security and improves the performance of DSR in MANETs. The working of Power Aware Secure Dynamic Source Routing (PS-DSR) is divided into four phases: Maintenance of trust table, Selection of monitor nodes, Detection of nodes behaviors with updation of trust values and Route selection.

The organization of the paper is as follows. Section 2 gives the literature review on security based routing protocols. In sec. 3, the proposed security scheme is covered in detail. Section 4 presents our simulation results and their analysis. Finally, the conclusions are drawn in section 5.

2 Literature Review

Dynamic Source Routing was developed and proposed for Mobile Ad Hoc networks by Broch, Johnson and Maltz [1]. There are a number of routing protocols in the literature that were proposed and implemented to secure MANETs [2]. Marti et al. designed Watchdog and Pathrater method [3] to optimize and improve the technique of packet forwarding in the Dynamic Source Routing (DSR) protocol. It has two major components: Watchdog and Pathrater. Watchdog component is used to detect selfish nodes and Pathrater then uses this information and helps routing protocols to avoid the detected nodes. Watchdog fails to detect a misbehaving node in the presence of: Ambiguous collisions, false misbehavior, Receiver collisions, Partial dropping and Limited transmission power. CONFIDANT (Cooperation Of Nodes, Fairness In Dynamic Ad hoc NeTworks) [4] enhances [3] and adds two other components to it: trust manager and reputation system. In [5], Pirzada et al. proposed a method for establishing trust based routing in MANETs without requiring a trust infrastructure. Node's trust in [5] is calculated taking in view the packet forwarding behavior. C. Wang et al. [6] proposed a routing algorithm tr-DSR, which is an extension of DSR and is based on nodes' trust and path's trust. The method used in the paper selects the highest trust path used for data transmission. Pirzada et al. [7] modified the DSR protocol such that intermediary nodes act as Trust Gateways that keeps track of trust levels of the nodes in order to detect and avoid malicious nodes. In Pirzada et al. [8] a trust-based model based on direct experience rather than trusted third party is proposed. In this, trust agents that reside on each node perform three functions: Trust Derivation, Quantification, and Computation. Huang Chuanhe et al. [9] proposed a trusted routing protocol called Dynamic Mutual Trust based Routing protocol (DMTR), based on DSR protocol that secures the network using the Trust Network Connect (TNC), and improves the path security which is selected by barrel theory. P. Narula [10] introduces a method of message security using trust-based multi-path routing. It uses soft encryption techniques and avoids introducing large overheads. The whole message is divided into parts and the parts are self-encrypted. S. K. Dhurandher and V. Mehra [11] proposed a trust based routing for Ad Hoc networks which protects the message against modification. In this, trust is calculated in a dynamic way and a path is used to transmit data based on the security requirement of the message. S.K. Dhurandher et al. [12] proposed FACES, in this trust of the nodes is determined by sending challenges and sharing friends' lists. In PS-DSR only few selected nodes works in the promiscuous mode, this approach for detection of malicious nodes differs PS-DSR from the existing models.

3 Power Aware Secure Dynamic Source Routing Protocol

In standard DSR protocol, the routes are selected on a first come first serve basis, however route shortening is done whenever there is a shorter route to the destination, but there is no process to detect malicious nodes present in the path. In order to surmount this shortcoming, we employ a power aware trust based scheme to secure DSR. The proposed method detects malicious packet dropping and packet modification attacks. It also ensures the trustworthy and authentic selection of routes. Malicious nodes present in the network are observed and detected by a set of nodes called monitor nodes. Monitor nodes monitor the behavior of their neighboring nodes and based on behavior, the trust of the neighboring nodes is updated. The set of monitor nodes is chosen from time to time to make PS-DSR adaptable to the two central problems of MANETs: dynamic network topology and energy constraint devices. PS-DSR follows the standard DSR strategy to obtain the network topology information, it constructs forwarding routes when the source node broadcasts the RREQ packets to its neighbors and then destination node or an intermediate replies with RREP packet containing the path to the destination node, and those routes are stored in the cache. However our proposed scheme PS-DSR differs from DSR such that when the PS-DSR searches the routes from the cache it selects whose trust values are greater than Mal_threshold. Power Aware Secure Dynamic Source Routing (PS-DSR) is divided into four phases as follows:

1. Maintenance of trust table
2. Selection of monitor nodes
3. Detection of nodes' behaviors and updation of trust values
4. Route selection

Route selection is based on the trust values of the hops in the route. In order to accommodate the trust values for all the nodes in the network, a trust table is maintained, which stores the trust values of all the network nodes. The source node checks the trust table, accesses the trust values and according to the route selection strategy explained in section D, it selects the most suitable route. The route selected by the proposed protocol is shortest trustworthy route.

3.1 Maintenance of Trust Table

The trust model maintains a trust table regularly updated by a set of nodes in the network, called the Monitor Nodes (MNs); these nodes constantly work in promiscuous mode and overhear packets in its neighborhood. These nodes are selected by using the algorithm given in [13]. The trust table is accessed by only few nodes in the network: Monitor Nodes and the source. These monitor nodes detect two types of attacks, first is the packet dropping attack, and second is the packet modification attack. Whenever a node in the network drops or modifies a packet, its neighboring monitor node decreases the trust value of that node, similarly when a node forwards a packet; its neighbor monitor node increases the trust value of that node. Since a node may have more than one neighboring monitor node that are working in promiscuous mode and overhearing the packets, there must be only one trust update for node's activity. In order to avoid multiple updates, the algorithm given in sect. 3.3, detects the packet forwarding behavior and ensures that only one monitor node will update the trust table according to its behavior.

3.2 Selection of Monitor Nodes

There are two main reasons for executing the algorithm for selection of monitor nodes set periodically and on demand. The first reason is the dynamic network topology. As MANETs consist of mobile nodes which are free to move. The second is the energy as mobile nodes mainly are battery powered devices. The periodic selection of monitor nodes ensures complete network coverage and these nodes observe the behavior of each network node and hence do not leave any malicious node undetected. The time interval for periodic selection of monitor nodes depends on the network stability. The on demand selection of monitor nodes ensures distributed loss of energy and prevents any node from becoming energy deficient and thus prevents the induction of selfish behavior, in which an energy deficient node does not forward the packets which it was supposed to forward, in order to save its own energy. All the monitor nodes are continuously checked for energy and if any monitor node is found to be deficient in energy, program for selecting a new set of monitor nodes is called [13]. In this state, algorithm [13] does not consider energy deficient nodes and hence new set of monitor nodes that have enough remaining energy are selected. The algorithm first computes the node degree of all the wireless nodes in a given network and then checks for the circular links of the nodes one by one. By the checking of circular links we mean that the node arranges its neighbors in increasing order of their node ID's and check if that particular set of nodes are connected by 1 hop or 2 hops, if all the circular links are not present then the node is marked as the monitor node, otherwise it checks for the log links and for that we take the floor value of the logarithm of node degree for that node, supposedly it comes "n" then we check the connectivity of nodes (either connected by 1 hop or 2 hops) at a distance of n hops away from that node in the circular fashion that was previously taken. If all the log links are present then the node is marked as the regular node, otherwise it is marked as the monitor node. If monitor node lies in the vicinity of the route selected for data forwarding, then that monitor node operates in promiscuous mode.

3.3 Algorithm to Detect Packet Forwarding Behavior and Prevent Multiple Updates

Terminology Used in Algorithm

1. tap(Packet P): Packets overheard in promiscuous mode enter tap() function, P carries the reference of the packet overheard.
2. node_id: It is the ID of the node currently executing the tap function.
3. monitors[i to j]: It is the array of node IDs of monitor nodes executing the tap function, index values lie in the range from i to j.
4. nexthop.access(P) : It is a function which returns the address (node ID) of next hop from the source route field of the packet P.
5. prevhop.access(P): It is a function which returns the address (node ID) of previous hop from the source route field of the packet P.
6. next_hop: This variable contains the address (node ID) of the next hop returned by function nexthop.access(P).
7. previous_hop: This variable contains the address (node ID) of the previous hop returned by function prevhop.access(P).
8. isNeighbor(monitors[k], next_hop): This function returns TRUE if its input integer arguments are within communication range of each other, otherwise it returns FALSE.
9. received[n]: This counter counts the packets received by the node n.
10. forward[n]:This counter counts the packets forwarded by the node n.
11. D represents the difference of packets received and packets forwarded by a node, i.e. number of packet drops.
12. destination_node: It is the ID of the destination node in the network
13. sequence.access(P): It is a function which returns the sequence number of the packet P.

Algorithm to Detect Packet Drops and Prevent Multiple Update of Trust Values

```
1. tap(Packet P) //packets overheard in promiscuous mode enter this function
2. {
3. Initialize integer seqno with -1;
4. Initialize integer ph with -1;
5. Initialize int next_hop with the next hop of the packet;
   // function nexthop.access(P);
6. Initialize int prev_hop with the previous hop of the packet;
   //function prevhop.access(P);
7. IF node_id is not present in the array monitors[i] to monitors[j] THEN RETURN;
8. ELSE IF next_hop is the neighbor of the monitor node THEN Increment the
   received packet counter of the next_hop by one AND Increment the forward
   counter of the prevhop by one one in the table of monitor node currently
   executing the tap function.
```

// Using monitors[k].received[next_hop]++; and monitors[k].forward[previous_hop]++; Where monitors[k] corresponds to node_id i.e. the ID of the node currently executing the tap function.

9. IF integer ph is not equal to the previous hop of the packet THEN Increment the trust of the previous hop of the packet and THEN Initialize ph with current previous hop of the packet
10. IF next_hop of the packet is not the destination node THEN calculate the difference between packets received and sent by the next_hop
    // Using D(difference)=monitors[k].received[next_hop]- monitors[k]. forward[next_hop];
11. IF that packet difference is greater than C AND IF integer seqno is not equal to the sequence number of the packet THEN decrease the trust of the next_hop in the table of the monitor node AND Initialize seqno with the current sequence number of the packet.
    //Where C is a constant and function used is dec_trust(next_hop , dec_amt);

Explaination of Algorithm. In the above algorithm, monitor nodes update the trust table of their neighbors depending on the packet forwarding behavior. The trust value of a node is incremented by an amount inc_amt, if a node shows benevolent behavior by forwarding the packet that it was supposed to forward. If a node drops a packet, the neighboring monitor node decrements its trust value by an amount dec_amt. In above algorithm we took two variables "seqno" and "ph" which contains sequence number and the previous hop of the packet respectively. We took the sequence number as the parameter to differentiate between the packets, for instance if a trust value is decreased on a particular sequence number by a monitor node, then other monitor nodes cannot decrease the value of trust on the same sequence number, it has to be another sequence number as multiple monitor nodes come across the activity of same packet drop or packet modification. A similar logic is applied when there is a need to increase the trust values in case a node forwards a packet without any malicious, in such a case variable "ph" prevents multiple updates to the trust table. As multiple monitor nodes overhear the packets forwarded by the nodes in the path, therefore there must be some mechanism to control multiple updates hence, for a particular forwarding hop i.e. previous hop ("ph") if trust is increased by a particular monitor node, then other monitors cannot increase the trust for the same hop and the same packet forward.

3.4 Route Selection

Source node computes the route trust of a route as:

$$RT_i = \frac{\sum T_i}{n_i} \tag{1}$$

Where RT_i is the route trust of route i, $\sum T_i$ is the sum of trust values of all the nodes in route i and n_i is the number of hops in route i and n_i. The route trust is directly proportional to the average trust of the nodes in the route and inversely proportional to the number of hops. The source node makes use of two threshold values for the selection of most trustworthy path between source and destination. The thresholds are termed as Mal_threshold and RT_threshold as elucidated below:

1. Mal_threshold: It is a value at or below which a node is declared as malicious.
2. RT_threshold: It is a value below which route is not considered as optimum.

If the trust values of all the nodes in the network surpass the Mal_threshold then the route trust for that route is now checked for RT_threshold. If the route satisfies these two criteria, then that route is selected for packet forwarding, otherwise route is rejected.

4 Simulation

4.1 Setup

We used Network Simulator NS-2.34 to evaluate the effectiveness of the proposed security scheme. We simulated and compared the results of the proposed protocol PS-DSR with standard DSR routing protocol. The simulation parameters are listed in Table 1.

Table 1. Simulation parameters

Parameter	Simulation Value
Simulator	NS-2.34
Examined Protocol	DSR and EESDSR
Simulation time	140 seconds
Simulation area	1500 x 300 m
Number of nodes	60
Transmission range	250 m
Movement model	Random Waypoint
Maximum speed	20 m/s
Pause time	0 seconds
Traffic type	CBR (UDP)
CBR rate	0.2 Mbps
Packet size	1000 bytes
Maximum malicious nodes	25
Initial energy	160 J
rxPower	1 W
txPower	1 W
idlePower	1 W

4.2 Metrics

The following metrics are used to evaluate the performance of the proposed security scheme through simulations:

1. Packet Delivery Ratio (PDR): It is the ratio of the total number of data packets received by the destination node to the total number of data packets sent by the source node.
2. Packet Loss Percentage: It is the percentage of the packets that were dumped by malevolent nodes to the total number of packets.
3. Average end-to-end latency: It is the average of time (including buffer delays during route discovery, queuing delays at interface queues, re-transmission delays at MAC layer and propagation time) taken by the data packets from source to destination.
4. Routing packet overhead: It is the fraction of the total number of control packets to the total number of data packets.
5. Path optimality: It is the proportion of the total number of hops in the shortest route to the total number of hops in the route selected by the protocol for transmitting data packets.

4.3 Trust Parameters

The values of the trust parameters taken for the simulation are listed in Table 2.

Table 2. Trust parameters

Parameter	Simulation Value
Trust Range	0.0 to 8.0
Mal_threshold	4.0
Initial trust value	6.0
inc_amt	0.02
dec_amt	0.05
RTh	1.05

4.4 Results and Analysis

In this section, we present the performance results for the proposed PS-DSR and that of the standard DSR protocol, in the presence of varying number of malicious nodes. Figure 1 shows a network scenario chosen for the implementation of PS-DSR and standard DSR. Green coloured nodes depict the nodes having high remaining energy, yellow coloured nodes represent the nodes having moderate energy and red coloured nodes represent the nodes having low remaining energy.

Figure 2 shows the number of monitor nodes selected by PS-DSR at different times, indicating periodic and on demand selection of monitor nodes. As shown in the graph, new set of monitor nodes are selected after every 50 seconds. The time interval of 50 seconds has been taken as it has been observed that the topology of the network changes drastically within 50 seconds and this requires the selection of new set of monitor nodes as previous monitor nodes do not cover the whole network. The set of 10 new monitor nodes selected at time 120 seconds is on demand as remaining energy of few monitor nodes fall below the required energy to work as a monitor node. The monitor node set selection algorithm does not consider energy deficient nodes and hence new set of monitor nodes that have enough remaining energy are selected to observe all network nodes in their neighborhood. The size of the MN set depends upon the density of the network and the communication range. For our simulations, we considered 250m communication range and the size of MN set varies, as the density of the mobile network varies with time. Figure 3 show that, the packet delivery ratio using the proposed scheme PS-DSR is higher than the standard DSR in the presence of 25 malicious nodes. This can be attributed to the fact that the later does not take into account the routes free from malicious nodes.

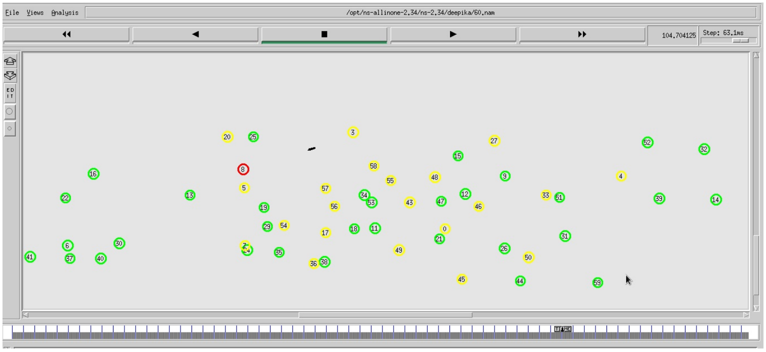

Fig. 1. Network topology of 60 nodes taken for the simulation

Figure 4 show that, the packet loss percentage using the proposed scheme PS-DSR is less than the standard DSR. PS-DSR selects the most trustworthy path avoiding the malicious nodes. This deviation from the routes selected by standard DSR leads to a raise in the packet overhead as shown in figure 5.

The average end to end delay of PS-DSR is more than the standard DSR, this can attributed to the delay in the route discovery process and the buffer delays. Average end to end delay is shown in figure 6. The path optimality presented in figure 7 for PS-DSR and the standard DSR comes out to be same in our simulation results.

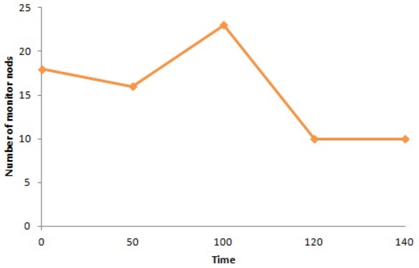

Fig. 2. Size of MN set selected by PS-DSR at different time intervals

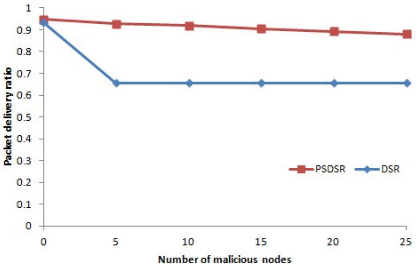

Fig. 3. Packet Delivery Ratio of PS-DSR and standard DSR for varying number of malicious nodes

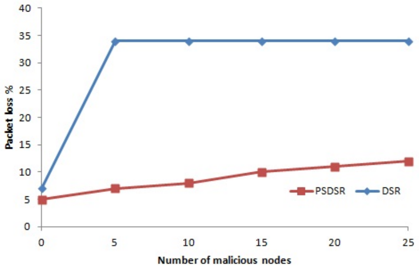

Fig. 4. Packet Loss Percentage of PS-DSR and standard DSR for varying number of malicious nodes

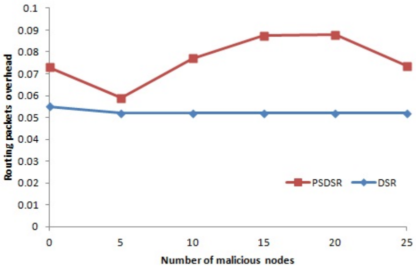

Fig. 5. Routing packet overhead for PS-DSR and standard DSR for varying number of malicious nodes

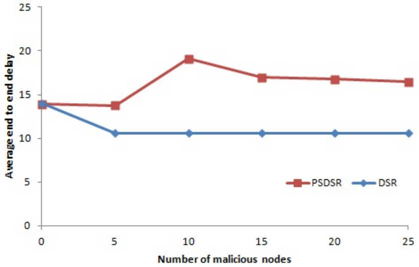

Fig. 6. Average end to end delay for PS-DSR and standard DSR for varying number of malicious nodes

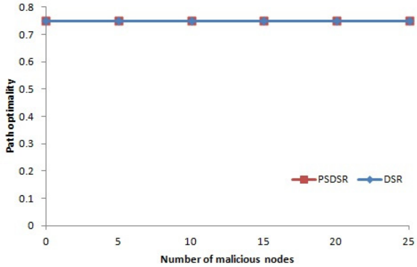

Fig. 7. Path optimality for PS-DSR and standard DSR for varying number of malicious nodes

5 Conclusion

PS-DSR selects a more reliable path as compared to standard DSR. A monitor node cannot intentionally abuse its neighbors as each monitor is also being monitored by its neighboring monitor node.There is a significant increase in the packet delivery ratio, hence packet loss is less, and moreover, there is a marginal increase in the routing overhead. Using PS-DSR, the source node selects a new route free from malicious nodes without any delay after the detection of malicious node/s in the current route. PS-DSR is a power saving protocol as nodes working as monitoring nodes change their status from monitor to regular nodes and vice versa as and when required. PS-DSR is adaptable to dynamic topology of the network as new monitor nodes are selected from time to time. PS-DSR is using a completely innovative approach for detection of packet drops, the algorithm is designed from the scratch and PS-DSR considers energy factor as well as Trust factor during its operation.

References

1. Johnson, D.B., Maltz, D.A.: Dynamic Source Routing in Ad-Hoc Wireless Networks. In: Imielinski, T., Korth, H. (eds.) Mobile Computing, pp. 153–181. Kluwer (1996)
2. Kukreja, D., Singh, U., Reddy, B.V.R.: A Survey of Trust Based Routing Protocols in MANETs. In: Fourth International Conference on Electronics Computer Technology (ICECT 2012), pp. 537–542. IEEE Press (2012)
3. Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks. In: Proceedings of Sixth Ann. Int'l Conf. Mobile Computing and Networking (MobiCom), pp. 255–265 (2000)
4. Buchegger, S., Boudec, J.: Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes-Fairness in Distributed Ad Hoc NeTworks. In: Proceedings of IEEE/ACM Workshop Mobile Ad Hoc Networking and Computing (MobiHOC), pp. 226–236 (2002)
5. Pirzada, A.A., Datta, A., McDonald, C.: Trust-Based Routing for Ad-Hoc Wireless Networks, pp. 326–330. IEEE (2004)
6. Wang, C., Yang, X., Gao, Y.: A Routing Protocol Based on Trust for MANETs. In: Zhuge, H., Fox, G.C. (eds.) GCC 2005. LNCS, vol. 3795, pp. 959–964. Springer, Heidelberg (2005)
7. Pirzada, A.A., McDonald, C.: Deploying trust gateways to reinforce dynamic source routing. In: Proceedings of the 3rd International IEEE Conference on Industrial Informatics, pp. 779–784. IEEE Press (2005)
8. Pirzada, A.A., McDonald, C.: Trust Establishment In Pure Ad-hoc Networks. Wireless Personal Communications 37, 139–163 (2006)
9. Chuanhe, H., Yong, C., Wenming, S., Hao, Z.: A Trusted Routing Protocol for Wireless Mobile Ad hoc Networks. In: IET Conference on Wireless, Mobile and Sensor Networks (CCWMSN 2007), pp. 406–409 (2007)
10. Narula, P., Dhurandher, S.K., Misra, S., Woungang, I.: Security in mobile ad-hoc networks using soft encryption and trust based multipath routing. Sci. Direct Comput. Commun. 31, 760–769 (2008)
11. Dhurandher, S.K., Mehra, V.: Multi-path and message trust-based secure routing in ad hoc networks. In: Int. Conf. Advances in Computing, Control and Telecommunication Technologies (ACT 2009), pp. 189–194 (2009)

12. Dhurandher, S.K., Obaidat, M.S., Verma, K., Gupta, P., Dhurandher, P.: FACES: Friend-Based Ad Hoc Routing Using Challenges to Establish Security in MANETs Systems. IEEE Systems Journal 5(2), 176–188 (2011)
13. Li, Y., Peng, S., Chu, W.: An Efficient Algorithm for Finding an Almost Connected Dominating Set of Small Size on Wireless Ad Hoc Networks, pp. 199–205. IEEE (2006)
14. Kukreja, D., Miglani, M., Dhurandher, S.K., Reddy, B.V.R.: Security enhancement by detection and penalization of malicious nodes in wireless networks. In: International Conference on Signal Processing and Integrated Networks, pp. 275–280. IEEE (2014)