Journal of Network and Computer Applications 36 (2013) 582–592

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Securing DSR against wormhole attacks in multirate ad hoc networks

Shams Qazi *, Raad Raad, Yi Mu, Willy Susilo

University of Wollongong, Australia

ARTICLE INFO

Article history:
Received 4 June 2012
Received in revised form
21 October 2012
Accepted 12 December 2012
Available online 8 January 2013

Keywords:
Ad hoc networks
Wormhole attacks
Dynamic source routing (DSR)
Round trip time (RTT)

ABSTRACT

A wormhole attack is one of the hardest problems to detect whereas it can be easily implanted in any type of wireless ad hoc network. A wormhole attack can easily be launched by the attacker without having knowledge of the network or compromising any legitimate nodes. Most existing solutions either require special hardware devices or make strong assumptions in order to detect wormhole attacks which limit the usability of these solutions. In this paper, we present a security enhancement to dynamic source routing (DSR) protocol against wormhole attacks for ad hoc networks which relies on calculation of round trip time (RTT). Our protocol secures DSR against a wormhole attack in ad hoc networks for multirate transmissions. We also consider the processing and queuing delays of each participating node in the calculation of RTTs between neighbors which to date has not been addressed in the existing literature. This work provides two test cases that show that not taking multirate transmission into consideration results in miss identifying a wormhole attack.

Contents

* Corresponding author. Tel.: +61 421964493.
E-mail addresses: shams@uow.edu.au (S. Qazi), raad@uow.edu.au (R. Raad), ymu@uow.edu.au (Y. Mu), wsusilo@uow.edu.au (W. Susilo).

1084-8045/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2012.12.019

1. Introduction

With the advancements in wireless communication, ad hoc wireless networks are becoming more popular platforms for different types of scenarios especially where it is expensive or infeasible to setup network infrastructure. These networks are threatened by many attacks because of their open architecture. These attacks could involve message tampering, identity spoofing, eavesdropping, blackhole attack (Hu and Perrig, 2004) and the rushing attack (Hu et al., 2004). Since many multihop wireless environments are resource-constrained (e.g., bandwidth, power, or processing), providing detection and countermeasures to such attacks often turn out to be more challenging than in their wired counterparts (Khalil et al., 2005).

One of the severe attacks is a *wormhole attack*, which has been introduced in the context of ad hoc networks (Hu et al., 2006; Wang et al., 2006; Capkun et al., 2003). In this attack, a malicious node captures packets from one location in the network, and "tunnels" them to another malicious node at a distant point, which replays them locally. The tunnel can be established in many different ways, e.g., through an out-of-band hidden channel (e.g., a wired link), a packet encapsulation, or a high powered transmission. This makes the tunneled packet arrive either sooner or with a lesser number of hops compared to the packets transmitted over normal multihop routes. This creates the illusion that the two end points of the tunnel are very close to each other. A wormhole tunnel can actually be useful if used for forwarding all the packets. However, in its malicious incarnation, it is used by attacking nodes to subvert the correct operation of ad hoc and sensor network routing protocols. The two malicious end points of the tunnel may use it to pass routing traffic to attract routes through them. They can then launch a variety of attacks against the data traffic flowing on the wormhole, such as selectively dropping the data packets. The wormhole attack can prevent two nodes from discovering legitimate routes greater than two hops away and thus disrupt network functionality. In addition, it may affect data aggregation and clustering protocols and location-based wireless security systems. It is important to note that the wormhole attack can be launched even without having access to any cryptographic keys or compromising any legitimate node in the network (Hu et al., 2006; Wang et al., 2006).

Our protocol secures DSR against wormhole attacks in ad hoc networks with the help of RTT calculations between intermediate nodes, which are participating in the route. Our protocol also calculates the processing times involved at each participating node while processing route request and route reply packets. The main differences between our protocol and other existing protocols (Tran et al., 2007; Alshamrani, 2011) is the consideration of processing time and multirate transmission. The overwhelming majority of wireless protocols support different transmission rates at the physical layer, it is not possible to detect a wormhole attack correctly in a wireless environment using algorithms defined in Tran et al. (2007) and Alshamrani (2011) as they assume the transmission rate between nodes to be constant. If links are faster or slower than the RTT between those nodes will be considerably different, therefore, it is hard to say whether this difference in RTT is because of wormhole or transmission rate.

1.1. Paper organization

Section 2 of this paper presents the wormhole attack, its different modes and DSR protocol. In Section 3, we present the related work done by the different authors. Section 4 presents our proposed protocol including system assumptions, notations, protocol run, attack model and examples. In Section 5, we present the security analysis of our protocol against wormhole attacks and also we compare it with existing solutions and Section 6 presents the conclusion.

2. Background

Wormhole attacks are mainly severe against wireless ad hoc network routing protocols, such as dynamic source routing (DSR) (Johnson et al., 2003) and Ad hoc On-Demand Distance Vector (AODV) (Perkins et al., 2003). In this section, we first discuss the working of DSR protocol and then different ways to launch a wormhole attack against DSR.

2.1. Dynamic source routing (DSR)

Dynamic source routing (DSR) protocol, is an on-demand routing protocol based on the concept of source routing, which means that the initiator knows the complete hop-by-hop route to the destination. This specific feature brings efficiency, but also results in the scaling of routing message overhead. To perform DSR, each node is required to maintain a route cache which contains the topology information of the network. The route cache is consistently updated to reflect the current status of the network.

DSR consists of two major phases: route discovery and route maintenance. In case of route recovery, source node generates routing request (RREQ) and broadcasts its to neighbors. The receiving node will append its own address to the RREQ packet and rebroadcasts it, if it is not the destination. On reception of RREQ packet at destination, node generates route reply (RREP) packet and forward back to the source, as shown in Fig. 1.

2.2. Modes of wormhole attacks

There are different ways to launch wormhole attacks in a wireless network environment which include using high power transmission, tunneling using encapsulation, tunneling using out-of-band channels, packet relay or protocol deviation (Khalil et al., 2005). In the following, we will discuss these in detail.

2.2.1. Wormhole using high power transmission

In this mode, a single malicious node can create a wormhole attack without the help of any colluding node. When a malicious node gets a route request, it broadcasts the request with high power as compared to normal nodes. Any node that hears the high-power broadcast, rebroadcasts it towards the destination. By this method, the malicious node increases its chance to be in the routes established between the source and the destination.

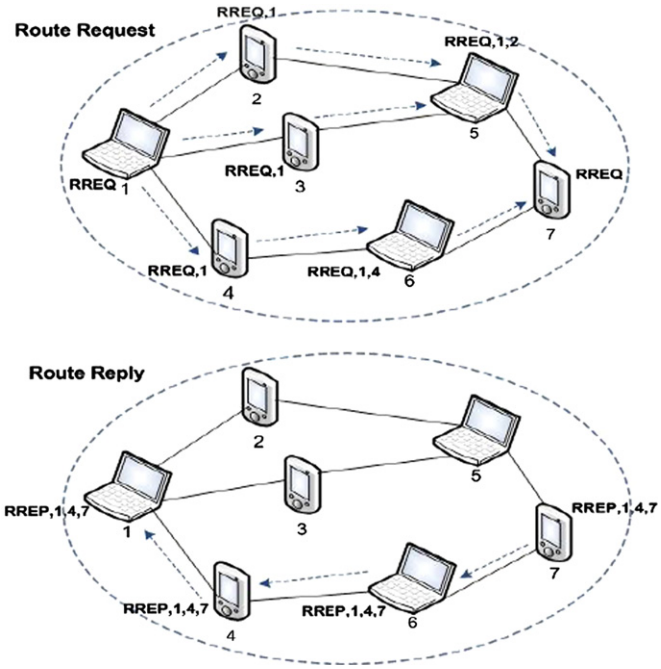

Fig. 1. Route discovery in DSR protocol.

2.2.2. Wormhole tunnel using encapsulation

In this mode of attack, two or more malicious nodes participate to create a tunnel between them and give false illusion that the route through them is the shortest, even though they may be far away. They create a tunnel with the help of normal nodes using encapsulation. Due to encapsulation, hop count does not increase during the traversal through intermediate nodes of tunnel, which launches wormhole attack between source and destination.

2.2.3. Wormhole tunnel using out-of-band channel

In this mode of attack, an out-of-band high-bandwidth channel between the malicious nodes is used to create a wormhole tunnel. This channel can be a long-range directional wireless link or a direct wired link. This type of attack requires specialized hardware, therefore, it is more difficult to launch as compared to encapsulation attack.

2.2.4. Wormhole using packet relay

In this mode of attack, a malicious node tries to convince two far nodes that they are neighbors by relaying packets between them. Even one malicious node can do this and if more malicious nodes are available then this can expand the neighbor list of victim nodes to several hops. Consider that node X and node Y are two non-neighbor nodes with a malicious neighbor node $M1$. Node $M1$ can relay packets between nodes X and Y to give them the illusion that they are neighbors.

3. Related work

In this section, we review some existing solutions against wormhole attacks in wireless ad hoc networks. These solutions can be divided into different categories, based upon type of solution. In some of the solutions, authors used additional hardware or softwares and some solutions are based upon calculation of RTT between intermediate nodes.

Hu et al. (2006) introduced the concept of geographical and temporal packet leashes to detect wormhole attacks in wireless networks. They added a leash to the packet with extra information to defend against wormholes. According to the authors, each node needs to know its own location and all nodes have loosely synchronized clocks. The geographical leashes ensure that the distance between sender and recipient is within certain limits. The temporal leashes ensure that all packets have an upper bound on its lifetime, which restricts the maximum travel distance. They require that all nodes have tightly synchronized clocks. An implicit assumption is that packet processing, sending, and receiving delays are negligible. Both geographical and temporal leashes need to add authentication data to each packet to protect the leash, which add significant processing and communication overhead. In addition, a large amount of storage is needed at each node since a hash tree based authentication scheme (Merkle hash trees) is used in Merkle (1980).

Capkun et al. (2003) presented wormhole-attack detection method without requiring any clock synchronization through the use of MAD (Mutual Authentication with Distance Bounding). Each node i estimates the distance to another node j by sending it a one bit challenge, which node j responds instantaneously. Using the time of flight, node i detects if node j is a neighbor or not. The approach uses special hardware module that can temporarily take over the control of radio transceiver unit of the node to immediately respond to one-bit challenge without the delay imposed by the usual way of processing messages. Khalil et al. (2005) present a simple lightweight protocol, called LITEWORP, to detect and mitigate wormhole attacks in static ad hoc and sensor wireless networks. LITEWORP uses secure two-hop neighbor discovery and local monitoring of control traffic to detect wormhole nodes and also provides a countermeasure that isolates the malicious nodes from the network. LITEWORP does not require specialized hardware, such as directional antennas or fine granularity clocks. It does not require time synchronization between the nodes in the network. LITEWORP does not increase the size of packet, it only incurs negligible bandwidth overhead, during initialization and detection of a wormhole. Detection and isolation of wormhole in LITEWORP is done wisely to minimize the possibility of victimizing nodes due to false alarms caused by natural collisions in the wireless medium or due to malicious framing.

Khabbazian et al. (2006) made two assumptions, the first is that there are a lot of distributed nodes in the network and the second is that the distance between the two nodes $\leq T$, where T is the transmission range. The authors also protected a packet hop count by a hash chain to stop attackers from reducing the hop count. In addition, they divide their work on different distances and examine where the two legitimate nodes are, and the distance and the number of hops in between. The results show that this analytic model is able to explain how the effect of the wormhole can be measured. However, this approach does not cope well when the attackers only intend to analyze the network traffic rather than disturbing any network traffic by dropping data packets.

Su (2010) proposed a routing protocol named WARP to defend ad hoc networks against wormhole attacks. WARP is basically modified form of AODV routing protocol by adopting link disjoint multi-path routing between source and destination. In WARP each node records all of its neighbor's anomaly values (number of times it forms path from different source to destination). Due to wormhole node's great ability to grab routing paths, if the occurrence of one links exceeds the threshold value, the two ends of this link may be wormhole nodes. If anomaly values of a node exceed a threshold value then its neighbor will discard all request for forming route containing that node in the path.

Yu et al. (2010) proposed a new Routing Security Scheme based on Reputation Evaluation (RSSRE) to secure ad hoc networks. Authors considered the case of hierarchical ad hoc network based on roles and functions of participating nodes. The reputation relation is built based

on the behaviors and correlation of the node. They chose relatively secure nodes by reputation evaluation in routing and updated the reputation through nodes relationship. In this paper, they considered AODV as routing protocol and claimed their protocol can be used in any routing protocol to safeguard routing security.

Qian et al. (2007) proposed an approach focusing upon analysis of routing statistics named as Statistical Analysis of Multipath (SAM). Through analysis of an ensemble of multipath routes obtained at the base station, suspicious links appearing with much higher frequency than expected can be excluded in favor of more diverse alternative pathways. The approaches presented provide resilience in case the wormhole alters route establishment messages, and allow easier extension to multi-sink scenarios as detection state is implicitly shared. However, if multipath routing is not specifically required by the application, then the additional overheads of it could render it unnecessary.

Garcia and Robert (2009) proposed a new routing protocol based on a modification of the Split Multipath Routing (SMR) protocol (Qian et al., 2007). The modified protocol allows intermediate nodes to forward repeated copies of a RREQ message, as long as their hop counts are not larger than the hop counts of already received copies. The destination should receive numerous copies of the RREQ message. Thus, the destination should be able to build a list of available paths from the source; this information gives a partial view of the network that would be used by the WIM–DSR protocol in the discovery of possible wormhole attacks. In this protocol, the destination chooses a path and broadcasts it towards the source. The intermediate nodes should rebroadcast only one copy of a given RREQ message. This step should allow intermediate nodes to validate the information.

An algorithm WRTTGDD is introduced in Prasannajit et al. (2010). This algorithm works on calculating the RTT and geographic distance. The WRTTGDD's operation can be divided into two steps: using a hop counting technique and RTT between each successive node. Then, every node must collect the set of hop counts of its neighbor nodes. In addition, the Dijkstra algorithm is used by each node to find the shortest route for every pair based on the RTTs and hop count. Furthermore, by using multi-dimensional scaling (MDS), a local map will be reconstructed. Then, distortions in local maps will be detected by the use of a diameter feature (hop counting). Further, the highest value of RTT belongs to the fake link that is created by the attackers, because in a normal network without wormholes, the authors claim that all the RTTs are nearly the same. This method helps to detect the wormhole attacks because it gives every node significant information about the nodes that are able to communicate directly. Although, this algorithm can detect wormhole attacks, it is not stated how to isolate malicious nodes to avoid future wormhole attacks.

Znaidi et al. (2008) introduced a new algorithm to detect a wormhole attack. This algorithm is applied on each given node to compute specific coefficients (CS) for its neighbor. Authors assume each node obtains the list of one and two-hop neighbors. Each node will send a HELLO message including its identity; therefore every node which hears the HELLO message must add this node to its neighboring list and then send a reply message to the sender of the HELLO message. After this, every two successive nodes share their neighbor lists with each other. The last process in this protocol is that after node (i) has received the neighbor list of node (j), it has to compare it with its own neighbor list. Thus, if there is at least one common neighbor, node (i) will consider the (j) node is a normal node. Otherwise, it will consider it as a suspicious node, and put it in its red list. Therefore, node (i) has to broadcast a message to inform all nodes that (j) is a suspicious node. Therefore, the black alert message will be sent to all neighbors to delete the malicious node by a node that has received a number of alert messages. The results show this algorithm works well in detecting the existence of a single wormhole in classical networks.

In Dong et al. (2009), a topological based scheme is proposed to analyze the wormhole issue and by observing the inevitable topology deviations introduced by wormholes. Authors classified the wormholes according to their impact on the network and propose a topological approach. This approach solely relies on topological information of the network and detects wormholes by detecting non-separating loops (pairs). They formally proved the correctness of this design in continuous geometric domains and extend it into discrete domains.

In Tran et al. (2007), Transmission Time based Mechanism (TTM) is used to detect wormhole attacks on AODV routing protocol, which is the closest work to the one presented in this paper. In (TTM), round trip time (RTT) is calculated between two successive nodes throughout the route. The RTT can be calculated by subtracting the RREQ forwarding time from the RREP receiving time. When the sender generates the RREQ, it records the sending time. When the node receives the RREQ, it processes the RREQ and then rebroadcasts it and further, records its sending time as well, and so on until the RREQ reaches the target destination. Each node participating in the route receives the RREP generated by destination later on. Thus, every participating node records the RREP receiving time. Then, each node calculates its RTT with the destination and appends it to the extensional part in the RREP which is already created by the destination. When the source node gets the RREP, it triggers the detecting process to check if the established route is valid or not. The source node will calculate RTTs between every two successive nodes along the path based on RTT values in the extensional part of RREP. The authors believed that if the difference between the RTTs of successive nodes is higher than the threshold (which they assumed 45 s based upon simulation results) value then there is a wormhole.

Figure 2 shows the complete time-line, of how the RREQ travels through all the nodes, as well as the RREP in the reverse direction. In order to calculate the RTT, each node records the RREQ forwarding time TN_{REQ} and the RREP receiving time TN_{REP}, and calculates the RTT between destination and itself. All these calculated results forwarded to source S with RREP packet, which was generated by the destination. Finally, the source S calculates the RTT between each two successive nodes. According to Fig. 2, we obtain four RTT values. The first value is $RTT_{S,A}$, the second value is $RTT_{A,B}$, the third value is $RTT_{B,C}$, and the last value is $RTT_{C,D}$.

The authors also mentioned about the processing time required at each node which can affect the value of RTT and they

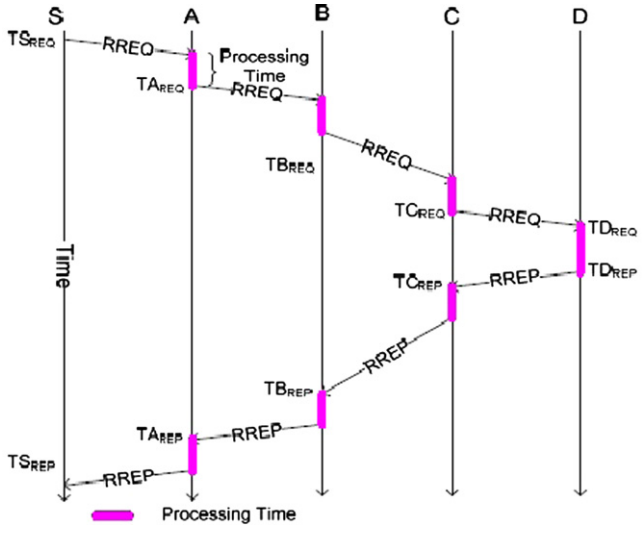

Fig. 2. Time of RREQ and RREP packets.

proposed a mechanism that instead of calculating the RTT between two nodes by measuring once, it is measured several times, say k times, afterward to calculate the average value of RTT. The authors considered that this average RTT value gives better results in detection of wormhole but in actual it does not really work because of difference in transmission time due to congestion in the network at different times and also difference in processing time at different time intervals.

The following are the possible threats which can affect the performance of TTM:

1. In TTM, the authors only considered single or fixed rate transmission whereas in wireless networks, the transmission rate can vary from one point to other depending upon the capacity of node and the wireless conditions. In fact, we will show through an example how TTM wrongly identifies wormholes due to different rates of transmissions on the wireless link. Therefore, it is really important to provide a solution for multirate transmission.
2. The second disadvantage in TTM is the longer RTT without the presence of a wormhole. This longer RTT may be due to processing or queuing delay at any participating node, which is not considered by TTM. In TTM, authors calculate the RTT several times to obtain average value and consider it as an accurate value. But in reality it is hard to get accurate values by calculating the average because there is a need to reduce the number of route requests.
3. The third disadvantage in TTM is that each node has the right to record the forwarding time of RREQ, and the receiving time of RREP as well, we may think malicious nodes will record fake times, unlike the time they use in the transmission. By doing this the source may not be able to detect the wormhole link and may not be able to recognize that the network is under an attack.
4. The fourth disadvantage that makes the TTM mechanism inefficient to detect and locate the exposed wormhole attack, is the ability of the malicious nodes to delay forwarding both the RREQ and the RREP packets. By doing this, the source will not be able to pinpoint the wormhole link and the source will have more than one RTTs value which are larger than the average.
5. Another possible threat in TTM is that malicious nodes can change the RTTs forwarded by neighboring nodes because all the RTTs attached with RREP packet are in normal text. Hence, malicious nodes can easily change these values to distract the source.
6. TTM is also not secure against wormholes created by packet relay and high power transmission.

In our proposed protocol, we consider all these threats and our protocol secures DSR against wormhole attacks in an ad hoc network.

4. Proposed protocol

In this section, we propose a secure DSR protocol against Wormhole attacks in ad hoc networks which support multirate speeds at their physical layer. In the following sub sections, we present system assumptions, the notations used in our protocol and the definition to prove our protocol's security against wormhole attacks.

4.1. System assumptions, notations and definitions

We consider an ad hoc network consisting of N nodes and are communicating over a shared wireless medium. Links between nodes are assumed to be bidirectional, i.e. given a link $L(A,B)$ between nodes A and B in an ad hoc network there exists the link $L(B,A)$.

We use a directed graph $G(N,E)$ to model an ad hoc network where N is a finite set of nodes and E is a finite set of bi-directional wireless radio link between the nodes. Each node $N_i \epsilon N$ has unique ID (IP address) and moves randomly. Each mesh node N_i has transmission radius R, according to wireless transmission mode. N_j is the neighbor of N_i, if node N_j is in the transmission range R of node N_i and there is a bi-directional wireless link $E(i,j)$ and $E(j,i)$ between the two nodes, as assumed earlier. We assume that M is a finite set of malicious nodes present in the network to create wormhole attack whereas M must be greater than 1 and less than $(N-1)$.

We use dynamic source routing (DSR) protocol as the routing protocol over the IEEE 802.11g medium access control protocol. All the nodes in the network are in promiscuous mode because in dynamic source routing environment, each node examines every packet it receives. As the node examines the addresses in each packet, it learns where other nodes are located relative to the node examining packets. Due to this, nodes do not need to transmit periodic routing advertisements, such as Routing Information Protocol (RIP) transmissions that are used to inform other nodes about the state of network.

IEEE 802.11g supports bandwidth up to a maximum of 54 Mbps and approximately 22 Mbps on average, and it operates in the 2.4 GHz ISM band. Importantly and of relevance to our protocols, IEEE 802.11g supports rates at 6, 9, 12, 18, 24, 36, 48 and 54 Mbps and 5.5 and 11 Mbps when working with IEEE 802.11b. IEEE 802.11g is backwards compatible with 802.11b, meaning that 802.11g access points will work with 802.11b wireless network adapters and vice versa but important point here is that if any of the participating node is working with 802.11b then the whole transmission through that node will be 802.11b with lower bandwidth as compared to 802.11g. While we are only considering IEEE 802.11g for our examples, our protocol can be applied over any Multi-rate MAC such as IEEE 802.11n.

Data rate, packet size and processing time at each node play an important role in our protocol because we calculate the round trip time between the nodes and compare it with the data rate offered by IEEE 802.11g to check whether there exists a wormhole or not. It is important to note that other protocols that attempt to do a similar function do not consider the case of multirate transmissions. In our proposed protocol, a request packet is divided into two parts, fixed and dynamic (depending upon no. of hop count), size of which can be calculated as below

$$RREQ\ size = 24 + (4 \times\ no.\ of\ hop\ count)$$

for example if the hop count=4 then RREQ size=24+16=40 bytes

$$RREP\ size = 24 + (4 \times\ no.\ of\ hop\ count) + (18 \times\ no.\ of\ hop\ count)$$

whereas 18 bytes are used to carry the request packet receive/forward, reply packet receive/forward time, request packet size

Table 1
Notations.

TN_{RREQ_R}	Route Request receiving time of Node N
TN_{RREQ_F}	Route Request forwarding time of Node N (noted by neighbors)
$RREQ_{SN}$	RREQ packet size at specific node N
TN_{RREP_R}	Route Reply receiving time of Node N
TN_{RREP_F}	Route Reply forwarding time of Node N
$RREP_{SN}$	RREP packet size at specific node N
$RTT_{N_iN_j}$	Round Trip Time between nodes N_i and N_j
PT_{N_i}	Processing time at node N_i
c	Speed of light (3×10^8 m/s)
d	Distance between two nodes
R	Maximum range of wireless node (300 m)
PD	Propagation delay equal to 0.001 ms
μ	μ is equal to 2 ms (limit for RTTs between participating nodes)

and reply packet size at each node (4 bytes to store each time stamp and 1 byte to store packet size in bytes).

In our proposed protocol, the source node calculates the round trip time between intermediate nodes depending upon the times received in the reply packet and calculates the processing time (if required) and then compares it with the existing data rate to detect the presence of wormhole attack which is discussed in a later section. The notations used in our proposed protocol are summarized in Table 1.

In our protocol, μ is a threshold value which is used to compare the difference between expected and measured values of RTT. The difference between expected and measured RTTs should ideally be zero but in case of lower or higher values than μ indicate the detection of a wormhole in our protocol. We assumed μ equal to 2 ms considering the factors involved in real time environment like congestion, etc.

4.2. Protocol run

In our proposed protocol, we are calculating the RTTs between the participating nodes but the most important thing we are considering the case of multirate transmission between them. In our mechanism, during the establishment of a route between source S and destination D, the source is responsible for calculating RTTs between all the intermediate nodes and processing time at each node whereas all participating nodes including the destination are responsible to forward their timestamps TN_{RREQ_R}, TN_{RREQ_F}, TN_{RREP_R} and TN_{RREP_F} to the source along with the route reply packet. As we already assumed that all the nodes are working in promiscuous mode, therefore, neighboring nodes can monitor and note down the time when their next hop neighbor forwards the same request packet. This is another important difference in our protocol that the request forward time of each node is monitored/noted by the neighboring node so there are less chances that malicious node alter request forward time to create illusion that delay is because of processing or queuing. After the calculation of RRTs between all nodes, the source compares RTTs and identifies a wormhole (if it exits) based on a threshold function. The fact that the expected RTT of two fake neighbors or two node wormhole tunnel will be considerably much higher or much lower than the measured RTT.

In DSR, when a source node forwards a RREQ to find out the route for the destination, it receives a RREP from the destination after some time through the help of intermediate nodes. Therefore, RTT is the time between forwarding the RREQ packet and receiving the corresponding RREP packet. Each node taking part in the route can also overhear when the neighboring node forwards the same request packet after processing. Each node along the route stores the time when it receives RREQ and the time when it receives RREP, whereas neighboring node stores the time when the same RREQ packet is forwarded by the next hop node. In Tran et al. (2007), each participating node calculates the RTT and forwards it to the source with RREP packet, whereas in our protocol, all participating nodes forward their request receiving time, reply receiving time, reply forwarding time and request forwarding time of a neighboring node to the source with RREP packet. Now at the source node all calculations are being done which is more secure as compared to mechanism discussed in Tran et al. (2007) because in our protocol, the source has all the information and can compare the request receiving and request forwarding times of specific node to calculate processing time involved at that node. The source then selects the best possible route and starts communication with the destination (usually the shortest path). The source also broadcasts a message to all nodes about the malicious nodes (if any exist).

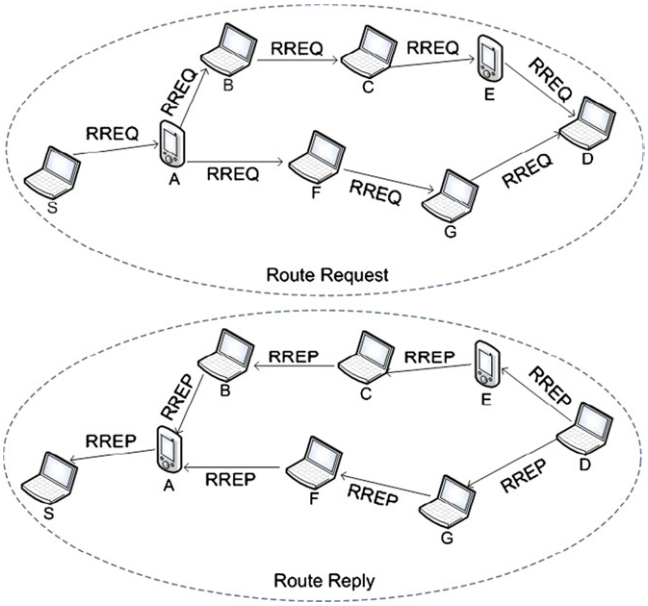

Fig. 3. Route request in the absence of wormhole attack.

4.2.1. Calculation of RTT and processing time

In this section, we discuss the calculation procedure of RTT between neighboring nodes and processing time (PT) at each participating node. Let us assume that node S wants to communicate with node D and S does not have routing information for D in its routing table/cache as shown in Fig. 3. To find out the best possible route, S broadcasts a route request RREQ with some alteration according to our protocol as mentioned below.

.As shown in Fig. 3, there are two possible routes available from source S to destination D. One is ($S \rightarrow A \rightarrow B \rightarrow C \rightarrow E \rightarrow D$) and second route is ($S \rightarrow A \rightarrow F \rightarrow G \rightarrow D$). Source node S receives replies from both routes with all the corresponding values as mentioned below.

1. $S \rightarrow * : RREQ, D, TS, SR\{S\}$
2. $A \rightarrow * : RREQ, D, TS, SR\{S,A\}$
3. $B \rightarrow * : RREQ, D, TS, SR\{S,A,B\}$
4. $F \rightarrow * : RREQ, D, TS, SR\{S,A,F\}$
5. $C \rightarrow * : RREQ, D, TS, SR\{S,A,B,C\}$
6. $G \rightarrow * : RREQ, D, TS, SR\{S,A,F,G\}$
7. $E \rightarrow * : RREQ, D, TS, SR\{S,A,B,C,E\}$
8. $D \rightarrow G : RREP, S, TS, SR\{S,A,F,G,D\}, TD_{RREQ_R}, RREQ_{SD}, TD_{RREP_R}, TD_{RREP_F}, RREP_{SD}$
9. $G \rightarrow F : RREP, S, TS, SR\{S,A,F,G,D\}, TD_{RREQ_R}, TD_{RREQ_F}, RREQ_{SD}, TD_{RREP_R}, TD_{RREP_F}, RREP_{SD}, TG_{RREQ_R}, RREQ_{SG}, TG_{RREP_R}, TG_{RREP_F}, RREP_{SG}$
10. $F \rightarrow A : RREP, S, TS, SR\{S,A,F,G,D\}, TD_{RREQ_R}, TD_{RREQ_F}, RREQ_{SD}, TD_{RREP_R}, TD_{RREP_F}, RREP_{SD}, TG_{RREQ_R}, RREQ_{SG}, TG_{RREP_R}, TG_{RREP_F}, RREP_{SG}, TF_{RREQ_R}, RREQ_{SF}, TF_{RREP_R}, TF_{RREP_R}, RREP_{SF}, TG_{RREQ_F}$
11. $A \rightarrow S : RREP, S, TS, SR\{S,A,F,G,D\}, TD_{RREQ_R}, TD_{RREQ_F}, RREQ_{SD}, TD_{RREP_R}, TD_{RREP_F}, RREP_{SD}, TG_{RREQ_R}, RREQ_{SG}, TG_{RREP_R}, TG_{RREP_F}, RREP_{SG}, TF_{RREQ_R}, RREQ_{SF}, TF_{RREP_R}, TF_{RREP_R}, RREP_{SF}, TG_{RREQ_F}, TA_{RREQ_R}, RREQ_{SA}, TA_{RREP_R}, TA_{RREP_F}, RREP_{SA}, TF_{RREQ_F}$

Similarly source node S receives the second route reply. Now the source needs to calculate the RTT and transmission time between the intermediate nodes to detect the existence of a wormhole attack.

Table 2
RTT between participating nodes and destination.

Node	TN_{RREQ_R}	TN_{RREQ_F}	$RREQ_{SN}$	TN_{RREP_R}	TN_{RREP_F}	$RREP_{SN}$	RTT_{ND}
S	TS_{RREQ_R}	TS_{RREQ_F}	$RREQ_{SS}$	TS_{RREP_R}	TS_{RREP_F}	$RREP_{SS}$	$TS_{RREP_R}-TS_{RREQ_F}$
A	TA_{RREQ_R}	TA_{RREQ_F}	$RREQ_{SA}$	TA_{RREP_R}	TA_{RREP_F}	$RREP_{SA}$	$TA_{RREP_R}-TA_{RREQ_F}$
F	TF_{RREQ_R}	TF_{RREQ_F}	$RREQ_{SF}$	TF_{RREP_R}	TF_{RREP_F}	$RREP_{SF}$	$TF_{RREP_R}-TF_{RREQ_F}$
G	TG_{RREQ_R}	TG_{RREQ_F}	$RREQ_{SG}$	TG_{RREP_R}	TG_{RREP_F}	$RREP_{SG}$	$TG_{RREP_R}-TG_{RREQ_F}$

Table 3
RTT between intermediate nodes.

$RTT_{SA}=RTT_{SD}-RTT_{AD}$
$RTT_{AF}=RTT_{AD}-RTT_{FD}$
$RTT_{FG}=RTT_{FD}-RTT_{GD}$

Source S calculates the RTT between participating nodes on the basis of values received with RREP packet and creates a timing Table 2 which includes the following information.

After the RTT calculation of all the participating nodes with the destination, the source node S calculates the RTT between the intermediate nodes, as shown in Table 3.

We have considered the case of multirate transmission in our protocol whereas the state of the art only considered constant data rate which cannot detect wormhole as illustrated in our example in the later section. According to TTM, if there is a wormhole tunnel involved in the network then the time between the wormhole tunnel end points is much greater or much smaller as compared to normal nodes, which is only true when there is constant transmission rate throughout the network (which is not a practical assumption).

Once the source node has calculated the RTT between neighboring nodes, the source has to compare all the actual RTTs with expected RTTs based upon the transmission rate between the neighboring nodes to check whether there exists a wormhole tunnel or not. For this purpose, the source runs an algorithm as shown below:

Algorithm 1. Wormhole checking between intermediate nodes.

Assume that N nodes are randomly placed in an ad hoc network and source calculated the RTTs of all the neighboring nodes involved in the route.
Calculate PT for RREQ and RREP Packets
Calculate TT for RREQ and RREP Packets
Calculate $RTT=(TT_{N_i}+PT_{N_i}+PD)$
Compare actual RTT with expected RTT
if $|A(RTT_{N_iN_{i+1}})-E(RTT_{N_iN_{i+1}})|\le|\mu|$ then
NO Wormhole
else
Wormhole Detected between N_i and N_{i+1}
end if

As shown in the algorithm above, source first calculates the processing time at each node and expected transmission time based upon packet size and available bandwidth between two nodes and then compares the actual RTTs with the expected RTTs of all the participating nodes and if the difference is less than or equal to μ then the route is considered to be safe, otherwise source flags an alert about wormhole detected between nodes N_i and N_{i+1}. To calculate excepted transmission time TT, the source can use following equation:

Table 4
Processing time calculations.

$PT_{RREQ_{N_i}}$	$PT_{RREP_{N_i}}$
$TN_{iRREQ_F}-TN_{iRREQ_R}$	$TN_{iRREP_F}-TN_{iRREP_R}$

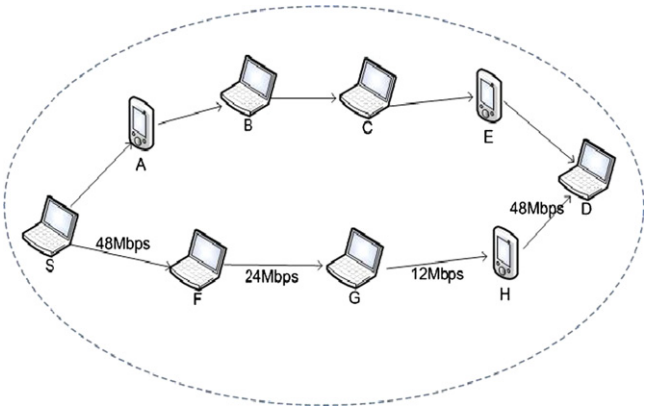

Fig. 4. Route request from source S to destination D.

$$TT=\frac{Packet\ Size(bits)}{Bandwidth(bps)} \tag{1}$$

Now the source has to calculate the processing time while processing RREQ and RREP packets simultaneously as shown in Table 4 of each intermediate node. As we assumed that our network is in promiscuous mode, therefore, TN_{RREQ_F} is monitored and forwarded by the neighboring node which is considered to be more secure as malicious node cannot change that value.

The source calculates the expected transmission time (TT) of RREQ and RREP packet using Eq. (1). Packet sizes are being forwarded by each node therefore the source can easily calculate the transmission time for two neighboring nodes as mentioned below

$$TT_{N_iN_{i+1}}=\frac{Packet\ Size\ (in\ this\ case\ its\ RREQ)}{Bandwidth}$$

$$TT_{N_{i+1}Ni}=\frac{Packet\ Size\ (in\ this\ case\ its\ RREP)}{Bandwith}$$

$$Therefore,\ RTT_{N_iN_{i+1}}=TT_{N_iN_{i+1}}+TT_{N_{i+1}Ni} \tag{2}$$

But in our calculation above RTT between two nodes does not include the processing time of RREP packet so we have to add processing time and propagation delay as well in this equation and then the source can compare the expected RTTs and actual RTTs. The generalized form of the calculation is as follows:

$$RTT=\sum_{i}^{2N-1}(TT_i+PT_i+PD) \tag{3}$$

$$Hence,\ RTT=\sum_{i}^{2N-1}\left(\left(\frac{Packet\ Size}{Bandwidth_i}\right)+PT_i+PD\right) \tag{4}$$

Once the source completes all the calculations, the source node can easily detect a wormhole attack by comparing the expected RTT values (calculated based upon transmission rate between the corresponding nodes and packet size) and actual RTT values (calculated based upon values received from corresponding nodes). The source can then avoid malicious nodes and choose the best possible route to communicate with the destination.

Table 5
RTTs with destination in TTM.

Node	TN_{REQ}	TN_{REP}	RTT_{ND}
S	0	34	34
F	0.5	30.5	30
G	3.5	23.5	20
H	8.5	15.5	7

Table 6
RTTs between intermediate nodes in TTM.

$RTT_{SF} = 4$
$RTT_{FG} = 10$
$RTT_{GH} = 13$
$RTT_{HD} = 7$

Table 7
RTT between participating nodes and destination.

Node	TN_{RREQ_R}	TN_{RREQ_F}	$RREQ_{SN}$	TN_{RREP_R}	TN_{RREP_F}	$RREP_{SN}$	RTT_{ND}
S	0	0	28	34	34	112	34
F	0.5	2.5	32	30.5	32.5	94	28
G	3.5	5.5	36	23.5	26.5	76	18
H	8.5	10.5	40	15.5	17.5	58	5

Table 8
RTT between intermediate nodes.

$RTT_{SF} = 6$
$RTT_{FG} = 10$
$RTT_{GH} = 13$
$RTT_{HD} = 5$

Table 9
Processing times at intermediate nodes.

Node	PT_{RREQ}	PT_{RREP}
F	2	2
G	2	3
H	2	2

Table 10
Expected and actual RTTs.

Nodes	Expected RTT	Actual RTT
RTT_{SF}	4.91	6
RTT_{FG}	8.3	10
RTT_{GH}	11.34	13
RTT_{HD}	4.04	5

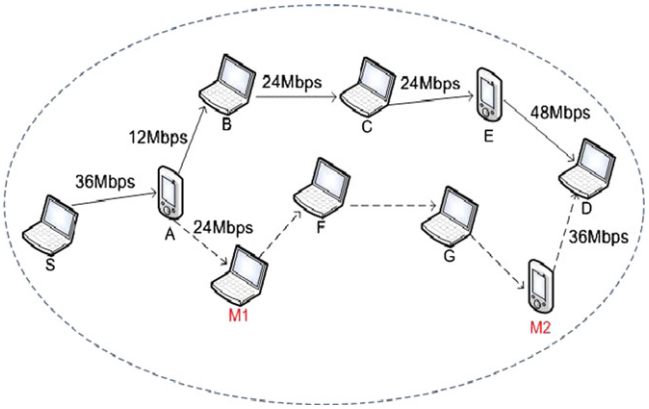

Fig. 5. Route request under wormhole tunnel with encapsulation.

4.3. Attack model

In this section, we consider two different examples to demonstrate how our proposed protocol works for the detection of a wormhole attack within multirate transmission and how algorithms that assume constant rate transmission such as TTM (Tran et al., 2007) provide inaccurate results. We consider that a wormhole is launched by malicious nodes.

4.3.1. First example

Let us assume that an ad hoc wireless network is established as shown in Fig. 4 and node *S* wants to communicate with node *D*. *S* does not have routing information for *D* in its routing table. DSR is the routing protocol whereas IEEE 802.11g is the MAC and physical layer protocol with multirate data transmission between the nodes as mentioned in Fig. 4. To find out the best possible route, *S* broadcasts a route request RREQ with some alteration according to our protocol.

4.3.2. Working of TTM (Tran et al., 2007)—multirate transmission example

In this subsection, we present, how TTM works for multirate transmission to detect the wormhole attack. The source *S* broadcasts the route request for the destination *D* and all the nodes participating in the route request appends the request packet by adding TN_{REQ} request time. Once the destination receives the request packet, it then prepares a reply packet and transmits it back to the same node from which it received the request. All the participating nodes append their route reply receiving time as well with the reply packet.

Upon reception of route reply, the source node calculates and creates the RTT tables as (Table 5).

After the calculation of RTT of all the participating nodes with the destination, now source node *S* calculates the RTT between the intermediate nodes, as shown in Table 6.

As shown in Table 6, RTT_{FG} and RTT_{GH} are large numbers as compared to other RTTs. According to TTM, nodes with the larger RTT are malicious and are part of a wormhole tunnel. Hence nodes *G* and *H* are wrongly identified as malicious. This occurs because there is a transmission rate deferential between the hops. TTM works if we assume the transmission rate at each node is constant. This is the main drawback of TTM and motivation for our work as our protocol works for both constant and multi transmission rate. In next subsection, we present how our protocol works for this example.

4.3.3. Working of proposed protocol—multirate transmission example

As discussed in the previous section, the source *S* broadcasts the route request for the destination *D*. The next node in the network receives that request packet and rebroadcasts it to its neighbors after performing necessary processing. All the neighboring nodes receive that request packet and rebroadcast it until it reaches the destination *D*. Then *D* prepares a reply packet and forwards it back to the same route from which it received the request. *D* replies to all the requests received from different

Table 11
RTT between participating nodes and destination.

Node	TN_{RREQ_R}	TN_{RREQ_F}	$RREQ_{SN}$	TN_{RREP_R}	TN_{RREP_F}	$RREP_{SN}$	RTT_{ND}
S	0	0	28	40.5	40.5	112	40.5
A	1	2	32	36	37	94	34
*M*1	3.5	5.5	48	30	32	84	24.5
*M*2	12.5	14.5	40	19	21	58	4.5

Table 12
RTT between intermediate nodes.

$RTT_{SA} = 6.5$
$RTT_{AM1} = 9.5$
$RTT_{M1M2} = 20$
$RTT_{M2D} = 4.5$

Table 13
Processing times at intermediate nodes.

Node	PT_{RREQ}	PT_{RREP}
A	1	1
*M*1	2	2
*M*2	2	2

Table 14
Expected and actual RTTs.

Nodes	Expected RTT	Actual RTT
RTT_{SA}	4.889	6.6
RTT_{AM1}	7.250	9.5
RTT_{M1M2}	13	20
RTT_{M2D}	3.722	4.5

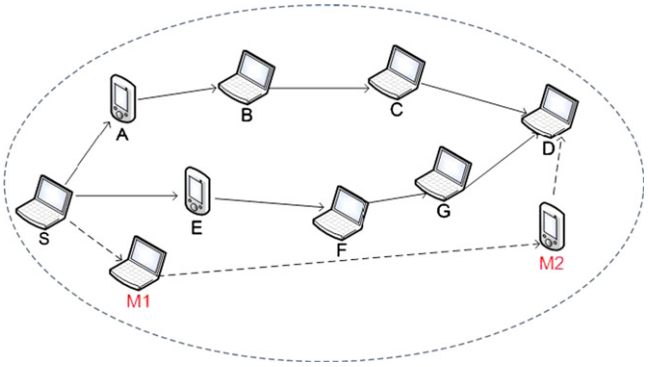

Fig. 6. Wormhole tunnel using out-of-band channel.

routes after fulfilling all the requirements mentioned in our protocol. All the nodes participating in the route forward back the route reply to their next hop until it reaches the source node *S*.

Upon reception of route reply, the source node calculates and creates the RTT tables as (Table 7).

After the calculation of RTT of all the participating nodes with the destination, the source node *S* calculates the RTT between the intermediate nodes, as shown in Table 8.

Now the source node calculates the processing time of RREQ and RREP packet at each node based upon the values stored in Table 4 as shown in Table 9.

Now the source node needs to calculate the expected RTTs based upon the link bandwidth and packet data size. The source calculates the expected RTTs as discussed earlier. Table 10 presents the expected and calculated RTTs of all the intermediate nodes.

As shown in Table 10, the difference between the actual RTTs (calculated based upon values received with reply packet) and expected RTTs (calculated based upon available bandwidth and data size) is less than threshold μ which is equal to 2 ms as discussed earlier. Ideally, this difference should be equal to zero but due to wireless environment, we considered it safe when it is less than or equal to μ. Hence, according to our protocol, there is no wormhole in this route and the longer delay in transmission is because of the different transmission rates between the nodes. But TTM has detected a wormhole attack in the same scenario.

4.3.4. Second example

Let us assume that an ad hoc network is established as shown in Fig. 5 and node *S* wants to communicate with node *D*. *S* does not have routing information for *D* in its routing table. DSR is the routing protocol whereas IEEE 802.11g is the MAC and physical layer protocol with multirate data transmission (minimum transmission rate is 12 Mbps) between the nodes as mentioned in Fig. 5. To find out the best possible route, *S* broadcasts a route request RREQ with some alteration according to our protocol.

Upon reception of route reply, the source node calculates and creates the RTT tables as (Table 11).

After the calculation of RTT of all the participating nodes with the destination, the source node *S* calculates the RTT between the intermediate nodes, as shown in Table 12.

Processing times of RREQ and RREP packets at each node are as shown in Table 13.

The source node needs to calculate the expected RTTs based upon the link bandwidth and the packet data size. The source calculates the expected RTTs as discussed earlier. Table 14 presents the expected and calculated RTTs of all the intermediate nodes.

As shown in Table 14, the difference between the actual RTT and expected RTT of node *M*1 and *M*2 is much greater than threshold μ. We assumed that the transmission rate between *M*1 and *M*2 is 12 Mbps which is our minimum transmission rate, even then difference is much greater. Hence, according to our protocol, there is a wormhole tunnel between *M*1 and *M*2. Source node broadcasts this information to all other nodes and discards this route. The source node checks for alternate routes and after successful checking, it selects the best possible route for communication with destination.

5. Security analysis

In this section, we present security analysis of our proposed protocol based upon the different wormhole attack modes as discussed in Section 2.

5.1. Security against packet encapsulation wormholes

As discussed in earlier sections, our protocol is secure against packet encapsulation wormhole attacks. In Fig. 5, *M*1 and *M*2 are two malicious nodes and they have created a wormhole tunnel between them with the help of packet encapsulation. Therefore, one route between source S and destination *D* is ($S \rightarrow A \rightarrow M1 \rightarrow M2 \rightarrow D$) and the other route is ($S \rightarrow A \rightarrow B \rightarrow C \rightarrow E \rightarrow D$). According to our protocol, we compare the expected RTT values and actual RTT values to check both the routes. Therefore, according to our protocol, the source node *S* discards route 1 and selects route 2 for communication with the

destination and hence, our protocol is secure against packet encapsulation wormholes.

5.2. Security against out-of-band wormholes

Our proposed protocol secures DSR against out-of-band wormhole attacks as well. As shown in Fig. 6, Node S sends a route request for node D, whereas $M1$ and $M2$ are malicious nodes having an out-of-band channel between them.

Node $M1$ tunnels the route request to $M2$, which is a legitimate neighbor of D. Node $M2$ broadcasts the packet to its neighbors, including D. D gets three route requests ($S \rightarrow M1 \rightarrow M2 \rightarrow D$), ($S \rightarrow A \rightarrow B \rightarrow C \rightarrow D$) and ($S \rightarrow A \rightarrow E \rightarrow F \rightarrow G \rightarrow C \rightarrow D$).

Once the source node S receives all these three routes replies, it calculates the RTTs between the consecutive nodes for all three routes and then decides which route to choose for communication. In case of route 1, the hop count is less when compared to other two routes but the difference between expected RTT and actual RTT of $M1$ and $M2$ is considerably smaller than all other neighboring nodes because $M1$ and $M2$ have high speed wired or wireless link. Our protocol is checking for all abnormal RTTs whether very high or very low. Therefore, route 1 is not selected, any other route can be selected based upon time and hop counts.

5.3. Security against high power transmission wormholes

This type of wormholes can be detected using the assumption of bi-directional links/channels. Suppose a malicious node say $M1$, tries to use high power transmission to forward a packet $P1$ to its final destination, or to cross-multiple hops to introduce itself in the shortest path. But on receiving a reply packet from all possible routes, the source node calculates the RTTs for all neighboring nodes. Based upon the RTT of all the consecutive nodes, a malicious node can be detected easily and the source does not select the route which contains the malicious node.

5.4. Security against packet relay wormholes

As in DSR, all nodes participating in active routes have the list of their neighbors, therefore if a malicious node $M1$ tries to relay a packet between two non-neighbor nodes A and B and deceives them that they are neighbors. Both nodes detect the malicious behavior of $M1$ since they know that they are not neighbor and they also calculate the RTT between them. Then this RTT can be compared with the RTT of two neighboring nodes to confirm whether there is a wormhole or not.

5.5. Security against TTM (Tran et al., 2007) threats

1. In our protocol, we also considered transmission and processing times to avoid any wrong detection as we discussed earlier in case of TTM when working with multirate transmission.
2. Our proposed protocol works in multirate transmission environment as well which was not covered in the literature. As shown in example, our proposed protocol identifies that the delay is not because of wormhole whereas it is because of slow transmission rate between intermediate nodes.
3. In our protocol, each node has to forward the request forwarding and reply receiving/forwarding time instead of RTT, therefore, the source can compare all the consecutive nodes' request and reply timings to make the decision about correctness of timings. This feature also helps us in taking care of the queuing delay involved at each node. Hence if a node stores corrupted data, it will be detected by the source.
4. According to our protocol, if any of the malicious node delays the RREQ or RREP packet that can be detectable by the source based of transmission time calculation and comparison of RTT of consecutive nodes.

6. Performance analysis

In this section, we present performance analysis of our proposed protocol in comparison with other existing solutions discussed in Section 3. As we have already mentioned earlier, in our proposed protocol, there is no requirement of any special hardware or any complex calculation or statistical analysis. Our protocol calculates RTTs based upon the values received through RREQ and RREP packets during route discovery process. According to our protocol, nodes require some additional memory to store RTTs of corresponding nodes and some extra processing time required to perform linear calculation to find out RTT between corresponding nodes and it depends upon the number of hops participating in that route. In other existing solutions, complex calculations or statistical analysis is required which is time consuming and also require extra memory.

If we compare performance in terms of memory and processing of our protocol with existing DSR protocol, there is not much difference because in our protocol, every participating node needs to add 18 bytes of extra data with RREP packet and all the calculations to find out RTTs is being done at the source. Therefore, our protocol does not create much difference as compared to DSR but in the end by using our protocol, we are able to safeguard our routing protocol against wormhole attacks.

Another important performance metric of our protocol is that we focused on multirate transmission problem which is not covered by existing solutions mentioned in Section 3. It is clearly shown in our examples that without considering multirate transmission, wormhole attacks may be detected wrongly or may not be detected properly.

7. Conclusions

In this paper, we discussed different modes of wormhole attack against wireless ad hoc networks and proposed a protocol to secure DSR against these attacks considering the multirate transmission environment. Most of the existing solutions discussed in Section 3, considered the case of fixed rate transmission and AODV as routing protocol but in our proposed solution, we considered DSR as routing protocol and considered multirate transmission environment which is a very important factor. We provided two different examples, one with fixed rate transmission and other with multirate transmission to explain the difference of our protocol with other existing protocols. Our proposed protocol is not only limited to DSR protocol, it can be widely used for any routing protocol with slight modifications to secure protocol against wormhole attacks. Furthermore, it can rapidly isolate the malicious nodes to improve the performance of routing protocols. Another benefit of our protocol over the existing solutions discussed in Section 3 is that our protocol does not require any special hardware or any complex calculations.

As future work, we intend to propose a generic solution to secure routing protocols against wormhole attacks in ad hoc networks in multirate transmission environment without assuming data rates between links.

References

Alshamrani AS. PTT: packet travel time algorithm in mobile ad hoc networks. In: Workshops of international conference on advanced information networking and applications; 2011.

Capkun S, Buttyán L, Hubaux J-P. SECTOR: secure tracking of node encounters in multi-hop wireless networks. In: ACM workshop on security of ad hoc and sensor networks (SASN); 2003. p. 21–32.

Dong D, Liu Y, yang Li X, Liao X, Li M. Topological detection on wormholes in wireless ad hoc and sensor networks. 2009.

Garcia L, Robert J-M. Preventing layer-3 wormhole attacks in ad-hoc networks with multipath DSR. In: Ad hoc networking workshop, 2009. Med-Hoc-Net 2009. 8th IFIP Annual Mediterranean; 2009. p. 15–20.

Hu Y-C, Perrig A. A survey of secure wireless ad hoc routing. IEEE Security and Privacy 2004;2(3):28–39 ISSN 1540-7993.

Hu Y-C, Perrig A, Johnson DB. Rushing attacks and defense in wireless ad hoc network routing protocols. In: ACM workshop on wireless security (WiSe); 2004. p. 30–40.

Hu Y-C, Perrig A, Johnson D. Wormhole attacks in wireless networks. IEEE Journal on Selected Areas in Communications 2006;24(2):370–80.

Johnson DB, Maltz DA, Hu Y-H. The dynamic source routing protocol for mobile ad hoc networks (DSR). Technical report, IETF MANET Working Group; 2003.

Khabbazian M, Mercier H, Bhargava V. NIS02-1: wormhole attack in wireless ad hoc networks: analysis and countermeasure. In: Global telecommunications conference; 2006. GLOBECOM '06. IEEE; 2006. pp. 1–6.

Khalil I, Bagchi S, Shroff NB. LITEWORP: a lightweight countermeasure for the wormhole attack in multihop wireless networks. In: In the international conference on dependable systems and networks (DSN); 2005; p. 612–21.

Merkle RC. Protocols for public key cryptosystems. In: IEEE symposium on security and privacy; 1980. p. 122–34.

Perkins CE, Royer EB, Das SR. Ad hoc on demand distance vector (AODV) routing. RFC 3561. Technical report, IETF; 2003.

Prasannajit B, Venkatesh, Anupama S, Vindhykumari K, Subhashini S, Vinitha G. An approach towards detection of wormhole attack in sensor networks. In: 2010 first international conference on integrated intelligent computing (ICIIC); 2010. p. 283–9.

Qian L, Song N, Li X. Detection of wormhole attacks in multi-path routed wireless ad hoc networks: a statistical analysis approach. Journal of Network and Computer Applications 2007;30(1):308–30.

Su M-Y. WARP: a wormhole-avoidance routing protocol by anomaly detection in mobile ad hoc networks. Computers and Security 2010;29(2):208–24.

Tran PV, Hung LX, Lee YK, Lee S, Lee H. TTM: transmission time-based mechanism to detect wormhole attacks. In: IEEE computer society; 2007. p. 172–8.

Wang W, Bhargava B, Lu Y, Wu X. Defending against wormhole attacks in mobile ad hoc networks. In: Wireless communication and mobile computing; 2006.

Yu Y, Guo L, Wang X, Liu C. Routing security scheme based on reputation evaluation in hierarchical ad hoc networks. Computer Networks 2010;54(9):1460–9.

Znaidi W, Minier M, Babau J-P. Detecting wormhole attacks in wireless networks using local neighborhood information. In: IEEE 19th international symposium on personal, indoor and mobile radio communications, 2008. PIMRC 2008; 2008. p. 1–5.