TTM: An Efficient Mechanism to Detect Wormhole Attacks in Wireless Ad-hoc Networks

Phuong Van Tran[1], Le Xuan Hung[1], Young-Koo Lee[1]*, Sungyoung Lee[1], and Heejo Lee[2]

[1]Dept of Computer Engineering., Kyung Hee University, Korea, [2]Dept of Computer Science and Engineering, Korea University, Korea

{tvphuong, lxhung}@oslab.khu.ac.kr, yklee@khu.ac.kr, sylee@oslab.khu.ac.kr, heejo@korea.ac.kr

Abstract— **Important applications of Wireless Ad Hoc Networks make them very attractive to attackers, therefore more research is required to guarantee the security for Wireless Ad Hoc Networks. In this paper, we proposed a transmission time based mechanism (TTM) to detect wormhole attacks – one of the most popular & serious attacks in Wireless Ad Hoc Networks. TTM detects wormhole attacks during route setup procedure by computing transmission time between every two successive nodes along the established path. Wormhole is identified base on the fact that transmission time between two fake neighbors created by wormhole is considerably higher than that between two real neighbors which are within radio range of each other. TTM has good performance, little overhead and no special hardware required. TTM is designed specifically for Ad Hoc On-Demand Vector Routing Protocol (AODV) but it can be extended to work with other routing protocols.**

Keywords- Intrusion Detection, Wormhole Attacks, Wireless Ad Hoc Networks, AODV

I. INTRODUCTION

Wireless Ad Hoc Networks are becoming more and more popular because of their important applications ranging from health care and logistics, through agriculture, forestry, civil and construction engineering, to surveillance and military applications [1]. Wireless Ad Hoc Networks are formed by a set of hosts that communicate with each other over a wireless channel. Each node has the ability to communicate directly with another node (or several of them) in its physical neighborhood. Such Wireless Ad Hoc Networks have many attractive features including automatic self-configuration and self-maintenance, quick and inexpensive deployment, and the lack of the need for fixed network infrastructures or centralized administration. These features lead to important applications that can not be performed by traditional wired networks. The importance of Wireless Ad Hoc Networks is increasing rapidly with advances in technology that result in smaller, cheaper, and power-efficient devices.

However, beside the advantages also Wireless Ad Hoc Networks have many security challenges because of their lack of fixed infrastructure, topology changing unpredictably, and broadcast nature of wireless communication. There are many kinds of attacks focusing on vulnerabilities in routing protocols for Wireless Ad Hoc Networks. One of the most popular & serious attacks is wormhole. In wormhole attacks, one or two colluding malicious nodes (wormhole nodes) using some techniques try to lure other legitimate nodes to send data via wormhole nodes. Afterward, wormhole nodes could exploit the data in variety of ways: selectively dropping packets to interrupt communication, trying to crack communication keys, etc. Because wormhole nodes do not need to modify or create new packets so no cryptographic technique can prevent Wireless Ad Hoc Networks from wormhole attacks.

Some work has been done to detect wormhole attacks in Wireless Ad Hoc Networks [5, 6, 7, 8, 9, 10, 11, 12, 13] but they do not efficiently eliminate wormhole from the networks (Section III). In this paper, we proposed a more efficient mechanism named TTM (transmission time base mechanism) to detect and locate wormhole attacks on the Ad Hoc On-Demand Distance Vector (AODV), one of the most popular routing protocols in Wireless Ad Hoc Networks. Our technique tries to detect wormhole during route setup procedure by calculating the transmission time between each two successive nodes along the established route. A wormhole will be identified based on the fact that transmission time between two wormhole nodes is considerably higher than that between two legitimate successive nodes. Our major contribution lies in the simplicity, low computation overhead and the high effectiveness of the proposed mechanism.

The rest of the paper begins with an overview of wormhole attacks and two kinds of wormhole in Wireless Ad Hoc Networks in Section 2. Some pros and cons of other proposed mechanisms so far are discussed in Section 3 to identify the challenging issues in detecting and locating wormhole attacks. Our mechanism is designed for AODV so we should describes briefly AODV route setup procedure before go to the main part of the paper - our proposed mechanism in Section 4. The advantages and disadvantages of our proposal are discussed in Section 5. Finally, Section 6 gives some conclusion and future work.

II. WORMHOLE ATTACK & CLASSIFICATION

We can think of wormhole attack as a 2-phase process launched by one or several malicious nodes. In the first phase, these malicious nodes, called wormhole nodes, try to lure legitimate nodes to send data to other nodes via them. In the second phase, wormhole nodes could exploit the data in variety of ways such as trying to break the encryption key, modifying packets or simply dropping packets selectively to make some legitimate nodes unable to communicate with each others.

How to lure legitimate nodes to send data via wormhole nodes? This work can be done in many ways [12]. In the simplest case, wormhole attacks include two malicious nodes which are able to communicate directly with each other from far distance via an out-of-band channel. One node will overhear packets at its location and tunnel them to the second node which in turn replays tunneled packets into the network at

* Dr. Young-Koo Lee is the corresponding author.

1-4244-0667-6/07/$25.00 © 2007 IEEE

its location. Because two wormhole nodes can communicate with each other directly from far distance so packets sent via wormhole link will be faster than those sent via normal nodes and paths containing the wormhole link are likely shorter than normal paths. Therefore, more nodes will send their data via wormhole nodes.

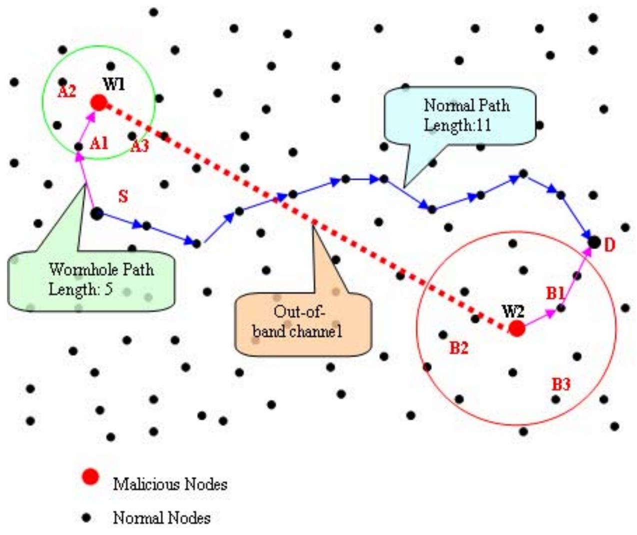


Figure 1: Wormhole Attack using out-of-band channel

For example, in figure 1, the path from S to D via wormhole link (W1, W2) has the length of 5 when the normal path has the length of 11. Therefore, in most routing protocols, S prefers sending data to D along the path with wormhole link.

However, the above method is difficult to deploy because it requires some special hardware to create an out-of-band channel. Another technique using encapsulation is more popular to launch wormhole attacks. Instead of using an out-of-band channel, the malicious node W1 encapsulates packets it overhears and sends them to the second malicious node W2 through the path exists between them. W2 decapsulates, gets the original packets and rebroadcasts them again. Because the original packets are encapsulated, they are not changed by intermediate nodes in the path between W1 and W2. By this way, W2 seems to get the packet directly from W1 with the same hop count although they are several hops far from each other.

Wormhole attack is serious to ad-hoc networks because it is easy to launch. The nature of wireless communication is broadcasting so wormhole nodes do not have to authenticate or communicate with legitimate nodes. They just overhear packets, tunnel them to the other node and replay into the network without any modifying or creating packets. So no encryption or authentication mechanism can protect Ad-hoc networks from wormhole attacks.

There are several ways to classify wormhole attacks. Here we divide wormhole attacks into 2 categories: hidden attacks & exposed attacks, depending on whether wormhole nodes put there identity into packets' headers when tunneling & replaying packets [13].

A. *Hidden Attacks*

Before a node forwards a packet, it must update the packet by putting their identity (MAC address) into the packet's header to allow receivers know where the packet directly comes from. However, in hidden attacks, wormhole nodes do not update packets' headers as they should so other nodes do not realize the existence of them. As showed in figure 1, a packet P sent by node S is overheard by node W1. W1 transmits that packet to node W2 which in turn replay the packet into the network. Because W1 & W2 do not change the packet header so D seems to get the packet directly from S. In this way, D & S are neighbors although they are out of radio range from each other (fake neighbors). General speaking, in hidden attacks nodes within W1's vicinity are "fake neighbors" of nodes within W2's vicinity and vice versa.

In this kind of attack, a path from S to D via wormhole link will be:

S → A1 → B1 → D

In the viewpoint of legitimate nodes, there is no existence of W1 & W2 in the path (hidden)

B. *Exposed Attacks*

In exposed attacks, wormhole nodes do not modify the content of packets but they include their identities in the packet header as legitimate nodes do (figure 1). Therefore, other nodes are aware of wormhole nodes' existence but they do not know wormhole nodes are malicious. In case of exposed attacks, the path from S to D (figure 1) via wormhole will be:

S → A1 → W1 → W2 → B1 → D

In hidden attacks, there are many fake neighbors created by wormhole link but there's no fake neighbor except (W1, W2) in this case. This difference leads to differences in detection mechanisms. Some mechanisms which can do well in detecting hidden attacks can not detect exposed attacks and vice versa.

C. *Related Work*

Some work has been done to detect wormhole in Ad Hoc networks. Most of them based on the fact that transmission time between two wormhole nodes or between two fake neighbors is much longer than that between two real neighbors which are close together. Because two wormhole nodes (or two fake neighbors) are far from each other and packets sent between two wormhole nodes maybe go through several intermediate nodes so it takes a longer time to transmit a packet between two wormhole nodes (or two fake neighbors) than between two real neighbors which are close together. By detecting this difference, we can identify wormhole attacks.

One of the first proposals for detecting wormhole is packet leashes [5]. Every time a node, say A, sends a packet to another node, say B, A has to put a time stamp (sending time) (temporal packet leashes) or the location of A and sending time (geographical packet leashes) into the packet. Based on this information, B can estimate the distance between A & B. If the estimated distance is longer than the possible radio range, B will reject the communication with A. These two mechanisms require tightly synchronized clocks (temporal packet leashes)

1-4244-0667-6/07/$25.00 © 2007 IEEE

or special hardware for location (geographical packet leashes) which is expensive to use widely. Therefore, we can say these two mechanisms are impractical with current technology.

In order to avoid using special hardware, Jane Zhen and Sampalli Srinivas try to detect wormhole using a so-called Round Trip Time (RTT) between two nodes [6]. A node, say A, calculates the RTT with another node, say B, by sending a message to node B requiring an immediate reply from B. The RTT between A and B is the time between A's sending the request message and receiving the reply message from B. In this mechanism each node (called N) will calculate the RTT between N and all N's neighbors. Because the RTT between two fake neighbors is higher than that between two real neighbors so by comparing these RTTs between A and A's neighbors, node A can identify which neighbors are fake neighbors and which neighbors are real neighbors. This mechanism do not require any special hardware and easy to implement but it can not detect exposed attacks because no fake neighbor is created in exposed attacks.

Another mechanism called DelPHI (Delay Per Hop Indicator), proposed by Hon Sun Chiu and King-Shan Lui [13], is able to detect both hidden and exposed wormhole attacks. In this mechanism, they try to find every available disjoint path between a sender and a receiver. Then, they calculate delay time & length of each path, computing Delay Per Hop value (average delay time per hop along each path). Delay Per Hop values of paths are used to identify wormhole: the path containing wormhole link will have greater Delay Per Hop value. This mechanism can detect both kind of wormhole but they can not pinpoint the wormhole location. Moreover, because lengths of paths are changed by every node (including wormhole nodes) so wormhole nodes could change the path length in a certain way to make them unable to be detected.

There are several other approaches which do not use transmission time to detect wormhole. In [10], the author proposed two statistical approaches to detect wormhole attack in Wireless Ad Hoc Networks. The first one called Neighbor Number Test bases on a simple assumption that a wormhole will increase the number of neighbors of the nodes (fake neighbors) in its radius. The base station will get neighborhood information from all sensor nodes, computes the hypothetical distribution of the number of neighbors and uses statistical test to decide if there is a wormhole or not. The second one called All Distance Test detects wormhole by computing the distribution of the length of the shortest paths between all pairs of nodes. In these two algorithms, most of the workload is done in the base station to save sensor nodes' resources. However, one of the major drawbacks is that they can not pinpoint the location of wormhole which is necessary for a successful defense.

In [9], another statistical approach called SAM (Statistical Analysis of Multi-path) was proposed to detect exposed wormhole attacks in Multi-path routing protocol. The main idea of the proposed scheme SAM is based on the observation that certain statistics of the discovered routes by routing protocols will change dramatically under wormhole attacks. Because wormhole links are extremely attractive to routing requests so it will appear in more routes than normal links. By doing statistics on the relative frequency of each link appear in the set of all obtained routes, they can identify wormhole attacks. This technique is only used to detect exposed attacks. It is unable to detect hidden attacks because in this kind of attack wormhole links does not appear in obtained routes.

To avoid the disadvantages of other proposed mechanisms, we set the goal of designing a mechanism which is able to detect both kinds of wormhole attacks, requiring no special hardware, locating wormhole location, having little overhead and good performance.

III. Proposed Mechanism

In TTM, we try to detect wormhole each time a route is requested. There is a two-fold benefit: first, we do not have to frequently check for wormhole which causes a lot of bandwidth and resource consuming and second, the wormhole will be identified before it can do any harm to the network because wormhole attacks have to interfere in the route setup before they can cause any damage. Our mechanism is designed specifically for AODV so we should go briefly into AODV route setup procedure first.

A. AODV route setup procedure

In AODV, when a node wants to communicate with another node and there is no valid route in its routing table, it broadcasts a route request packet (RREQ). A node receiving a RREQ for the first time will setup a reverse route to the source node in its routing table. If the node is the destination or has a valid route to the destination, it will unicast a route reply RREP along the reverse route back to the source node. Otherwise, it will increase the hop count in the RREQ by one and forward the RREQ to other nodes.

B. Transmission Time-based mechanism

In our mechanism, when a node establishes a route to another node, we will try to check whether there is a wormhole link in that route or not by calculating every Round Trip Time (RTT) between two successive nodes along the route. Each node in the established route will compute the RTT between it and the destination, then send this value back to the source node. The source node collects all of these RTT values, calculating RTTs between two successive nodes and identifying wormhole based on the fact that RTT between two fake neighbors or two wormhole links will be considerably higher than that between two real neighbors.

How to calculate the RTT values between two successive nodes along a route? There are various ways to do that job. In order to minimum the overhead, we will calculate these RTT values when the route is established. In AODV, a node will be in the route if it forwards a RREQ to the destination and receives a RREP from the destination later on. Therefore, we consider the time between an intermediate node sending the RREQ & receiving RREP as Round Trip Time between the intermediate node and the destination. Every node will save the time they forward RREQ & the time they receive RREP from the destination to calculate the RTT. Given all RTT values between nodes in the route and the destination, RTT between two successive nodes, say A and B, can be calculated as follows:

$RTT_{A,B} = RTT_A - RTT_B$

Where RTT_A is the RTT between node A and the destination

RTT_B is the RTT between node B & the destination

For example, the route from S to D includes:

$S \rightarrow A \rightarrow B \rightarrow C \rightarrow D$

During route setup procedure in AODV protocol, the time of sending RREQ & receiving RREP in each node along the route is described in Fig 2 (assuming that there is a wormhole link between B & C.)

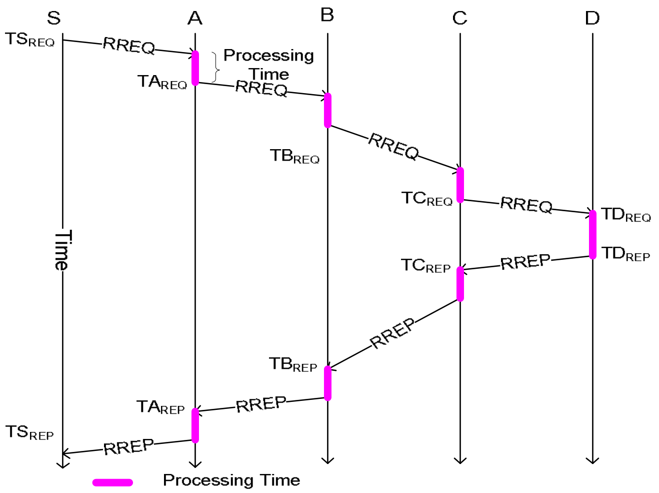


Figure 2 : Time of forwarding RREQ & receiving RREP

where TS_{REQ}, TA_{REQ}, TB_{REQ}, TC_{REQ} is the time the node S, A, B, C forward RREQ.

TS_{REP}, TA_{REP}, TB_{REP}, TC_{REP} is the time the node S, A, B, C forward RREP.

Then the RTT between S, A, B, C and D will be:

$RTT_{S,D} = TS_{REQ} - TS_{REP}$

$RTT_{A,D} = TA_{REQ} - TA_{REP}$

$RTT_{B,D} = TB_{REQ} - TB_{REP}$

$RTT_{C,D} = TC_{REQ} – TC_{REP}$

And the RTT values between two successive nodes along the path will be:

$RTT_{S,A} = RTT_{S,D} – RTT_{A,D}$

$RTT_{A,B} = RTT_{A,D} – RTT_{B,D}$

$RTT_{B,C} = RTT_{B,D} – RTT_{C,D}$

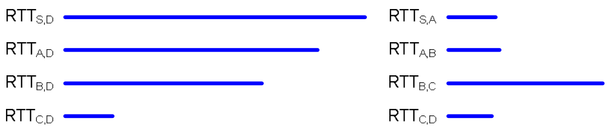


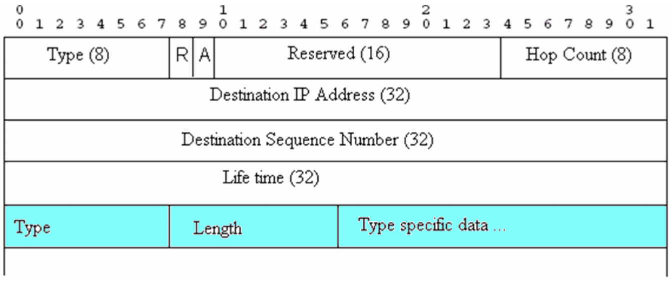
Figure 3 : Round Trip Time

Under normal situation, $RTT_{S,A}$, $RTT_{A,B}$, $RTT_{B,C}$, $RTT_{C,D}$ are similar but if there is a wormhole link between B & C, $RTT_{B,C}$ is considerably greater than $RTT_{S,A}$, $RTT_{A,B}$ & $RTT_{C,D}$.

C. *Sending RTT values back to the source node*

In our mechanism, the source node is in charge of collecting all RTT values between intermediate nodes along the established route and the destination, calculating RTT values between every two successive nodes then detecting wormhole. Every intermediate node along the route needs to send the RTT between them and the destination back to the source node. To reduce overhead, after receiving RREP, intermediate nodes will calculate the RTT and send the results along with the RREP back to the source node.

In AODV, it allows us to append user data to RREP packet (extensional part) in following format:



0 1 2 3 4 5 6 7 8 9 1 0 1 2 3 4 5 6 7 8 9 2 0 1 2 3 4 5 6 7 8 9 3 0 1

Type (8)	R	A	Reserved (16)	Hop Count (8)
Destination IP Address (32)				
Destination Sequence Number (32)				
Life time (32)				
Type	Length		Type specific data ...	

Figure 4 : RREP format

where the extensional part including: <Type, <Length> and <Type specific data>

When the destination node receives a RREQ, it will know how many intermediate nodes there are in the route (field <Hop Count>) so it will create a RREP with enough room in extensional part. Each intermediate node receive the RREP, calculating RTT value, putting that value into extensional part at right place then forwarding to the next hop along the reverse path. When the RREP reaches the source node, it contains all of the RTT values between intermediate nodes and the destination in the extensional part.

D. *Wormhole Detection*

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. As we know, a considerably higher RTT value between two successive nodes than others will indicate a wormhole link between those two nodes. The question is how much higher the RTT is considered a wormhole link. As in some other proposals, we used a threshold to make the decision. The threshold can be determined based on our simulation with appropriate parameters.

IV. Simulation Results

In this section, we evaluate the performance of our TTM by simulation using network simulator (ns2) [19]. In our experiments, the Ad-hoc network includes 50 nodes deployed randomly in a square field of 1000 meters by 1000 meters size. The transmission range is 250 meters. There is no movement of nodes and background traffic is generated randomly by a random generator provided by ns2. We created maximum of 8 CBR connections with the rate of 0.25 (4 packets per second). Packet size is 512 bytes.

Table 1: Simulation Parameters

Size	1000m x 1000m
Number of nodes	50
Transmission range	250m
Node movement	No
Background traffic	Light

Two wormhole nodes are randomly placed into the network. These two nodes, named W1 & W2, establish a tunnel between them using encapsulation. In all simulations, W1 & W2 do not put their identities into packet headers (hidden attack). In case of exposed attack, the performance is a little reduced because the distance between fake neighbors created by wormhole link in hidden attack is farther than the distance between two fake neighbors (two wormhole nodes) in exposed attacks. Note that in hidden attack, two wormhole nodes (n) hops far from each other will create pairs of neighbors which are (n + 2) hops far (n hops in case of exposed attack.)

Figure 5a shows the average RTT between two neighbors (the line with stars) and average RTT between two fake neighbors created by wormhole link (the line with cycles). With the light background traffic, the average RTT between two real neighbors is about 17ms. We can see that the RTT between two n hop far fake neighbors is about n times more than that between two real neighbors.

One of the important things in our mechanism is how to determine the threshold to detect wormhole which has a great effect on the performance of our mechanism. The threshold is proportional to false negative rate and disproportional to false positive rate. Figure 5b shows the relation between threshold and false detection rate. The threshold must be selected in order to minimum both false positive rate and false negative rate. Here, we set the threshold a value of 45ms (around the intersect of two lines)

With the threshold of 45ms, we run the simulation 1000 times with different wormhole length. Figure 5c shows the detection rate and false positive rate. The detection rate is proportional to the wormhole length. This is easy to understand because the more the wormhole length, the longer the transmission time between two fake neighbors and the easier to detect. The detection rate is 100% when the wormhole length is more than or equal to 6.

The simulation result shows a good performance of TTM. The reason is that we can calculate each RTT between two successive nodes along a path in individual and use these values to detect wormhole, not the average RTT Per Hop [13]. So the result is more accurate and we could pinpoint the wormhole location – the link having considerably higher RTT than the others.

V. Discussion

In this section, we are going to address the overhead of TTM in term of bandwidth and memory used.

In term of memory used, to calculate the RTT, every node needs to allocate memory for the information of each RREQ they get & forward. The information includes the destination of the RREQ (4 bytes) and the time the RREQ comes (4 bytes). After nodes get RREP & calculate the RTT, the memory allocated for the information of the according RREQ will be free. Be aware that before the information of the first RREQ is released, the node maybe gets several other RREQ so new memory is needed for new RREQ. Therefore, each node needs n x (4 + 4) bytes memory for this information where n is the maximum number of RREQ come to the node at the same time. The value of n depends on the traffic and the topology of the network. In our simulation, we set n as 4. That means each node is required just 4*(4+4) = 32 bytes memory to run our proposed mechanism.

In term of bandwidth used, the overhead is evaluated by comparing the number of bytes transmitted in the network in each route request when there is no wormhole prevention mechanism and when our mechanism is deployed. We set:

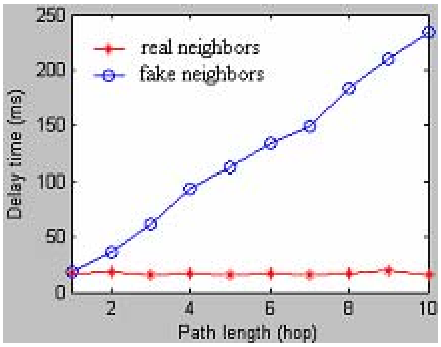


Fig 5a. Transmission time

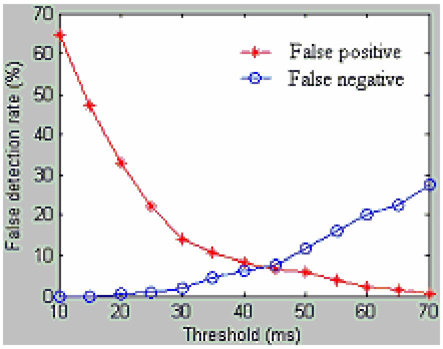


Fig 5b. False detection rate

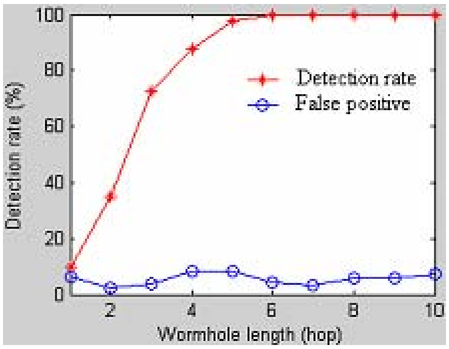


Fig 5c. Detection rate

N : number of nodes

L : length of the established route

$E0$: number of bytes transmitted in each route request when there's no wormhole prevention mechanism

$E1$: number of bytes transmitted in each route request when TTM is deployed.

In each route request in AODV, every node has to forward the RREQ once and the RREP is forwarded by nodes along the established route. Therefore, N RREQs and L RREPs are transmitted over the network. In AODV, the size of RREQ is 32 byte and size of RREP is 20. We have:

$$E0 = 32N + 20L$$

In our mechanism, every node in the established path is required to add the RTT into the RREP before forwarding it to the next hop. RTT is a 4 byte value so the length of RREP will be: 20 + 4L (bytes). We have:

$$E1 = 32N + L(20 + 4L)$$

In our simulation with $N = 50$, we get:

$$E0 = 1691.58878$$

$$E1 = 1775.47382$$

The overhead is inconsiderable to the network traffic. Note that this overhead happens only when a new route is requested.

VI. Conclusions and Future Work

In this paper, we proposed an efficient mechanism – TTM - to detect wormhole in Wireless Ad Hoc Networks using AODV routing protocol by calculating & comparing the Round Trip Time between every two successive nodes along that route during route setup protocol. TTM is able to detect both hidden & exposed wormhole attacks, locating the wormhole, requiring no special hardware. In table 1, we make a comparison between TTM and others in term of ability to detect exposed attacks, hidden attacks, to pinpoint wormhole location and to avoid using special hardware.

Table 2: Evaluation of some related work

	Exposed Attacks	Hidden Attacks	Pinpoint Location	No special HW required
Packet Leashes		✓	✓	
SAM	✓		✓	✓
DelPHI	✓	✓		✓
Neighbor Number Test		✓		✓
Neighbor Authentication		✓	✓	✓
TTM	✓	✓	✓	✓

The performance of TTM is also evaluated by simulation using network simulator. The simulation shows that the mechanism can detect wormhole attack with 100% accuracy when the wormhole length is large enough. Some future work also needs to be done to extend our mechanism to work in other routing protocols such as DSDV and DSR.

Acknowledgment

This work is financially supported by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energy (MOCIE) and the Ministry of Labor (MOLAB) through the fostering project of the Lab of Excellency.

References

[1] C.-K. Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems, Prentice Hall, 2002.

[2] W. Wang and B. Bhargava. Visualization of wormholes in sensor networks. In Proceedings of the ACM Workshop on Wireless Security (WiSe), 2004.

[3] Anthony D. Wood and John A. Stankovic: Denial of Service in Sensor Networks, IEEE Computer, October 2002, pp. 61-62.

[4] C. Karlof and D. Wagner: Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures, Ad Hoc Networks, vol. 1, pp. 293-315, 2003.

[5] Y. Hu, A. Perrig, and D. Johnson: Packet leashes: a defense against wormhole attacks in Wireless Ad Hoc Networks. In Proceedings of the IEEE Conference on Computer Communications (Infocom), 2003.

[6] J. Zhen and S. Srinivas. Preventing replay attacks for secure routing in ad hoc networks. Proc. of 2nd Ad Hoc Networks & Wireless (ADHOC-NOW'03), pp. 140--150, 2003.

[7] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. N. Levitt. A speci_cation-based intrusion detection system for AODV. In ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03)

[8] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In Proceedings of the IEEE Symposium on Network and Distributed System Security (NDSS), 2004.

[9] Lijun Qian, Ning Song, and Xiangfang Li. Detecting and locating wormhole attacks in Wireless Ad Hoc Networks through statistical analysis of multi-path. IEEE Wireless Communications and Networking Conference - WCNC 2005.

[10] Levente Buttyán, László Dóra, István Vajda: Statistical Wormhole Detection in Sensor Networks. Second European Workshop on Security and Privacy in Ad Hoc and Sensor Networks (ESAS 2005) Visegrád, Hungary, July 13-14, 2005: 128-141

[11] R. Poovendran and L. Lazos. A graph theoretic framework for preventing the wormhole attack in Wireless Ad Hoc Networks, to appear in ACM Wireless Networks.

[12] Issa Khalil, Saurabh Bagchi, Ness B. Shroff, LITEWORP: A Lightweight Countermeasure for the Wormhole Attack in Multihop Wireless Networks, International Conference on Dependable Systems and Networks (DSN 2005): 612-621

[13] Hon Sun Chiu King-Shan Lui, DelPHI: Wormhole Detection Mechanism for Ad Hoc Wireless Networks, International Symposium on Wireless Pervasive Computing ISWPC 2006.

[14] Y. Zhang, W. Lee, and Y. Huang, "Intrusion Detection Techniques for Mobile Wireless Networks", Wireless Networks, Vol. 9, No. 5, pp.545-556, Sep 2003.

[15] A. Mishra, K. Nadkarni and A. Patcha, "Intrusion Detection in Wireless Ad Hoc Networks", IEEE Wireless Communications, Vol. 11, No. 1, pp.48-60, February 2004.

[16] Y. Hu and D. Johnson, "Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks", Proceedings of MobiCom, pp.231-242, Aug 2000.

[17] Tran Van Phuong, Hung Le Xuan, Seong Jin Cho, Young-Koo Lee, Sungyoung Lee, An Anomaly Detection Algorithm for Detecting Attacks in Wireless Sensor Networks. ISI 2006: 735-736