Defending Against Wormhole Attack in OLSR

HONG Liang  HONG Fan  FU Cai

ABSTRACT  OLSR (optimal link state routing) is one of the four basic routing protocols used in mobile ad hoc Networks by the MANET working group of IETF (Internet engineering task force). OLSR, a proactive routing protocol, is based on a multipoint relaying flooding technique to reduce the number of topology broadcast. OLSR uses periodic HELLO packets to neighbor detection. As introduced in Reference [1], the wormhole attack can form a serious threat in wireless Networks, especially against many ad hoc Network routing protocols and location-based wireless security systems. Here, a trust model to handle this attack in OLSR is provided and simulated in NS2.

KEYWORDS  mobile ad hoc Network; wormhole attack; OLSR; securing routing

CLC NUMBER  TP393

Introduction

In a multihop wireless ad hoc Network, mobile nodes cooperate to form a Network without using any infrastructure such as access points or base stations. Instead, the mobile nodes forward packets for each other, allowing communication among nodes outside wireless transmission range. The nodes’ mobility and the fundamentally limited capacity of the wireless medium, together with wireless transmission effects such as attenuation, multipath propagation, and interference, combine to create significant challenges for routing protocols operating in an ad hoc network.

Several routing protocols for ad hoc Networks have been developed, particularly in the MANET working group of IETF. OLSR [1] was proposed in 2003, which belongs to the proactive class of routing protocols. OLSR is an optimization of the classical link state algorithm tailored to the requirements of a mobile wireless LAN. The key concept used in the protocol is multi-point relays (MPRs). MPRs are nodes selected in charge of forwarding broadcast messages during the flooding process. This technique substantially reduces the message overhead as compared with a classical flooding mechanism, where every node retransmits each message when it receives the first copy of the message. So this protocol is particularly suitable for large and dense Network.

In Reference [2], attacks on ad hoc protocols generally fall into one of two following categories: routing-disruption attacks and resource-consumption attacks. Wormhole attack is classified into routing-disruption attacks. In the wormhole attack, an attacker records packets (or bits) at one location in the Network, tunnels them to another location, and relays them there. Due to the nature of wireless transmission, the attacker can create a wormhole even for packets not addressed to itself, since it can overhear them in wireless transmission and tunnel them to the colluding attacker at the opposite end of the wormhole.

The OLSR’s neighbor discovery mechanisms
rely heavily on the reception of HELLO packets to neighbor detection, so it is extremely vulnerable to this attack. When an attacker tunnels through a wormhole to a colluding attacker near node B all HELLO packets transmitted by node A, and likewise tunnels back to the first attacker all HELLO packets transmitted by B, then A and B will believe that they are neighbors, which would cause the routing protocol to fail to find routes when they are not actually neighbors. Furthermore, the attacker is invisible at higher layers, unlike a malicious node in a routing protocol, which can often easily be named, the presence of the wormhole and the two colluding attackers at either endpoint of the wormhole are not visible in the route.

In this paper, we introduce a trust model to evaluate the trustiness of "a node is a neighboring node". By dynamical built-up trust, we can avoid choosing the route caused by wormhole attackers. In Section 1 we review related works, Section 2 describes our solution, and Section 3 provides an evaluation of our scheme applying in OLSR. In Section 4 we present conclusions.

1 Related works

The important approach for preventing wormhole attacks is presented in References [3, 4]. The main idea is that by authenticating either an extremely precise timestamp or location information combined with a loose timestamp, a receiver can determine if the packet has traversed an unrealistic distance for the specific network technology used. Temporal leashes rely on extremely precise time synchronization and timestamps in each packet. But to construct a temporal leash, all nodes must have tightly synchronized clocks, which in fact are not easy to achieve in MANET. Geographical leashes rely on all nodes knowing its own location and having loosely synchronized clock. In that paper, the authors also point out that in some circumstances, bounding the distance between the sender and receiver, cannot prevent wormhole attacks.

Another method of preventing wormhole attacks is known as RF watermarking, which authenticates a wireless transmission by modulating the RF waveform in a way known only to authorize node. But if the radio band in which communications are taking place is known, then an attacker can attempt to tunnel the entire signal from one location to another. Some authors also propose using intrusion detection to handle the wormhole attack, but intrusion detection is difficult to isolate the attacker in a software-only approach.

2 Security scheme

2.1 Trust model

We use an adaptation of the trust model configured by Marsh for use in pure ad hoc Networks. Marsh's model computes situational trust in agents based upon the general trust in the trustor and in the importance and utility of the situation in which an agent finds itself. General trust is basically the trust that one entity assigns another entity based upon all previous transactions in all situations.

In our model each node have a trust evaluator which gathers data from the neighbor's events in all states, filters it, assigns weights to each event and computes different trust levels based upon them. The trust evaluator has three functions: trust derivation, quantification, and computation.

At first, in OLSR the trust can come from the information about the successful transmission of any packet that is relayed by the neighboring node, such as some acknowledgments. Second, the neighboring node's HELLO packet received on schedule can also conduce to the trust. These events can be categorized into data and control packet types, and in each event there are two states: success and fail, which record the number of successful events and failed events respectively. All these are shown in Table 1.

| HELLO packets received in time(Ca) | Success | $H_n$ |
|----------------------------------|---------|------|
| Fail                             |         | $H_f$ |
| Data forwarding(Ca)              | Success | $D_n$ |
| Fail                             |         | $D_f$ |
In trust quantification process, we represent trust from -1 to 1 signifying a continuous range from complete distrust to complete trust. Trust computation involves an assignment of weights to the event that were monitored and quantified. We use the continuous range from 0 to 1 for representing the significance of a certain event from unimportant to most important. The higher weights represent the event more important. We define the trust $T_x(y)$ to the neighboring node $y$ by the node $x$, and it is given by the following equation:

$$T_x(y) = \sum_{i=1}^{n} [W_x(i) \times T_x(i)]$$

where $W_x(i)$ is the weight of the $i$th trust category to $x$ and $T_x(i)$ is the situational trust of $x$ in the $i$th trust category. The $n$ represents the number of category.

From above equation, we can get the following equations:

$$C_x = \frac{H_s - H_p}{H_s + H_p} \text{ for } H_s + H_p \neq 0 \text{ else } C_x = 0$$

$$C_d = \frac{D_s - D_p}{D_s + D_p} \text{ for } D_s + D_p \neq 0 \text{ else } C_d = 0$$

Negative values represent that more failed events occur than successes. Hence, a value of -1 represents complete distrust, a value of 0 implies a non-contributing event and a value of +1 means absolute trust in a particular event.

Now the node $x$ can get the whole trust $T$ to the neighboring node $y$:

$$T_x(y) = W_x(C_x) \times T_x(C_x) + W_x(C_d) \times T_x(C_d)$$

2.2 OLSR review

The procedure of OLSR is as follows. Every node broadcasts HELLO messages that contain one-hop neighbor information periodically. The TTL of HELLO messages is 1, so they should not forwarded by its neighbors. With the aid of HELLO messages, every node obtains local topology information.

A node (also called selector) chooses a subset of its neighbors to act as multi-point relaying nodes for it is based on the local topology information, which are specified in the periodic HELLO messages later. MPR nodes perform two tasks: 1) when the selector sends or forwards a broadcast packet, only its MPR nodes among all its neighbors forward the packet; 2) the MPR nodes periodically broadcast its selector list throughout the MANET (again, by means of MPR flooding). Thus every node in the Network knows through which MPR nodes every other node could be reached.

With global topology information stored and updated at every node, a shortest path from one node to every other node could be computed with Dijkstra's algorithm, which goes along a series of MPR node.

2.3 Extension to OLSR

The framework of extension to OLSR is shown in Fig. 1.

When the node receives a new sender's HELLO message, it will make two new records, <node, positive, negative, event>, to record separately the event of this sender's HELLO message's coming in time or not, and the event of data forwarding successfully or not. Then in information collection there are two tables to record every possible neighbor's events.

These tables are the inputs of trust calculation. By trust calculation, every possible neighbor will get a value which represents the probability of the neighbor relationship. The tuples <neighbor, probability> will be recorded in Neighbor Set. Some OLSR information repositories and packets' format should be modified.

When the node broadcasts the HELLO message, it contains its neighbor information inclu-
The recommendation about the probability of neighbor relationships. From receiving others HELLO messages, every node obtains local topology information. When choosing MPR nodes, the node will take the node’s recommendation as an important factor.

When nodes exchange the TC messages which contain the information about the neighbor relationship’s probability, every node would get global topology information which can construct a weighted directed graph. The weight on the edge represents the evaluation of edge start point on the link existence between itself and the end point.

Then from the weighted directed graph of the global topology, we can use Dijkstra algorithm to calculate the routing table. In this process, the probability of the “being a neighbor” is considered as the weight.

3 Simulation

We used NS2 for the simulation. The NS2 simulator has been used extensively in evaluating the performance of ad hoc Network routing protocols. The parameters for the simulations are shown in Table 2.

| Parameter              | Value       |
|------------------------|-------------|
| Number of nodes        | 12          |
| Number of attackers    | 2           |
| Simulation duration    | 90          |
| Transmission range     | 100 m       |
| Attacker’s transmission range | 250 m |
| Traffic type           | CBR(UDP)    |
| Data payload           | 512 bytes   |
| Packet rate            | 20 kb/s     |
| Movement model         | No movement |

The node’s coordinates are shown in Fig. 2.

In the simulations we set up an UDP connection between No and N11 with time 20 s, No will start send packets to N11. In the network there are two attacking nodes, which are A1 and A2 in the figure. A1 and A2, which are the tunnel’s two ends, will execute the wormhole attack. A1 will tunnel all it’s hearing HELLO packets to A1, A2 will also tunnel all the hearing HELLO packets to A1, then both of them will replay the HELLO packets.

We simulate the originate OLSR protocol and the revised protocol under the same condition. The results are shown in Fig. 3. From the figure, we can see that the lower line is a zero line which simulate the originate OLSR protocol. The zero means that N0 can not find a right route to send the packet to N11, so N11 receives nothing from N0. All these happened cases are caused by wormhole attackers making N0 misbelieving N11 being its neighbor.

The upper line is the result of simulating the revised OLSR protocol, we can found at first N0 also can not find the right route to N11, but after evaluating some neighbor’s trustiness, N0 start to choose another route to send the packet, after many times trying and evaluating, N0 finally find a stable route to N11, so in the figure it shows that the transmitting rate is going to keep stable with time, and after 20 s, it keeps about 2.0 kb/s.

4 Conclusions

Because of the wireless medium’s openness,
every node can hear the neighbor’s radio without being detected. When two or more malicious nodes construct one or more wormholes, they can destroy the entire Network by disrupting the routing protocol, especially to OLSR protocols.

In this paper we introduced a trust model to evaluate the trustiness of “a node is the neighbor” in OLSR protocol. From the trustiness calculating, the node can get the right route instead of choosing the route caused by wormhole attack. This scheme can run with no need for network synchronization and GPS devices. But the scheme is based on trust evaluation, which predicts the future events by collecting the past events, so the trust evaluated by the node lags behind the attacks.

In future work, we will work on how to secure the trustiness message transmission and how to get the recommended path in trust graph. We also take the node’s mobility into consideration, because when the network topology changing fast, the route will change fast, which means the trust model should keep track with it.

REFERENCES

[1] Clausen T, Jacquet P (2003) Optimized link state routing protocol (OLSR) [OL]. http://www. ietf. org/rfc/rfc3626. txt

[2] Hu Yihchun, Perrig A (2004) A survey of secure wireless ad hoc routing[J]. IEEE Security and Privacy, 2(3): 28-39

[3] Hu Yihchun, Perrig A, Johnson D B (2003) Packet leashes: a defense against wormhole attacks in wireless Networks[C]. The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, San Francisco, CA, USA

[4] Hu Yihchun, Perrig A, Johnson D B (2003) Rushing attacks and defense in wireless ad hoc Network routing protocols[C]. The Workshop on Wireless Security, San Diego, CA, USA

[5] Hu Lingxuan, Evans D (2004) Using directional antennas to prevent wormhole attacks[C]. Network and Distributed System Security Symposium, San Diego, California

[6] Mash S (1999) Formalizing trust as a computational concept[D]. U.K., Department of Mathematics and Computer Science, University of Stirling

(Continue from Page 228)