Consideration on Monitoring Scheme to Secure Link State Routing against Byzantine Attacks

Babatunde Ojetunde[1,a)] Naoki Shibata[1,b)] Juntao Gao[1,c)] Minoru Ito[1,d)]

Abstract: Secure communication is essential in most applications such as battlefield and disaster management applications. Most existing protocols adopt cryptography based approach, trust based approach (reputation of nodes) and incentive based approach to detect and prevent attacks in such applications. However, such protocols are still subjected to drawbacks like expensive overheads, difficulty in maintaining secure key and session management, unsecured routes against Byzantine attacks and so on. In this paper, we introduce a monitoring scheme to secure packets route in link state routing protocol against Byzantine attacks. Specifically, each node creates an event-driven monitoring block which is used to record all activities of an intermediate node when receiving and forwarding packets. Our schemes provide mutual monitoring in which nodes in the network can validate the monitoring blocks of other nodes and report malicious activities. Also, our scheme uses a statistical method to predict the probability that a node is not benign. The proposed scheme provides protection against colluding attacks and other Byzantine attacks.

1. Research Background

Routing security is an important aspect of wired and wireless networks, that has been given wide consideration over the years. The significant improvement in the role that such security features play in the whole network cannot be over emphasized. This secure routing allows easy packet transmission between nodes without any form of compromise. Several security mechanisms have been proposed for various routing protocols in the past, however, routing protocols are still subjected to one form of attacks or the other.

An example of such attacks is the Byzantine attacks. In such attacks, a node can interrupt route discovery, impersonate destination node, corrupt routing information, completely drop packet or inject fake packets into the network. Thereby preventing timely delivery of packets from source to destination. These types of attacks can either be carried out by a malicious node from outside the network or within the network. Although these forms of attacks can be easily detected in a wired network, but ad-hoc networks are still more vulnerable to such threats.

Many works already carried out on route security adopt three main approaches: the cryptography based approach, trust based approach and incentive based approach [1]. In the cryptography based approach various cryptography mechanisms such as private and public key encryption schemes, digital signature, hash function and end-to-end authentication are adopted to secure the packet in routing protocols. The major drawbacks of this approach are the high computation overheads, maintaining secure key management and session management. While in trust based approach, nodes participating in the routing of packets are assumed to be trustworthy, thereby the security mechanism provided focuses more on the information being exchanged among nodes. Also, packets lost are often attributed to poor link quality which is often not the case as malicious nodes may drop packets. Another approach adopted is the incentive based approach, in which nodes participating in routing are given some form of incentives to report malicious nodes.

In this paper, we propose a monitoring scheme to secure the link state routing protocol against Byzantine attacks. First, we show the result of our survey on securing routing protocol against Byzantine attacks. Then we briefly discuss our approach which focuses on analyzing the actions of each node within the network using the link state routing protocol. Our proposed scheme will guarantee the communication among connected benign nodes in the network while detecting malicious nodes by a statistical method.

2. Related Work

Geetha *et. al.* [1] classified routing protocols into three distinctive types: proactive, reactive and hybrid protocols. They described proactive protocols as a protocol where nodes frequently exchange the network topology information and constructs routing tables to send packets from source to destination. An example of such protocols is Optimized Link State Routing protocol (OLSR), and DSDV. Also, reactive protocols are described as a protocol that ensure packets are sent from source to destination only when the need arises. Ad-hoc On Demand Vector (AODV) and Dynamic Source Routing are examples of reactive protocols. While hybrid protocols can be realized by adopting both proactive and reactive protocols. The route discovery makes use of the

[1] Nara Institute of Science and Technology
[a)] ojetunde.babatunde.nq3@is.naist.jp
[b)] n-sibata@is.naist.jp
[c)] jtgao@is.naist.jp
[d)] ito@is.naist.jp

proactive protocol scheme while the reactive protocol scheme is adopted for sending packets. Zone Routing Protocol (ZRP) and Fisheye State Routing (FSR) are few examples of hybrid protocols.

Harshavardhan [2] surveyed security issues in ad-hoc routing protocols and identified how to mitigate such security threats. Harshavardhan first stated the properties of an ad-hoc routing protocol as distributed operation, loop free, demand based operation, a unidirectional link support, security, quality of service support, multiple routes and power conservation. Then, they used findings from some related works to summarize different ad-hoc routing protocols before analyzing various security threats and techniques to mitigate them. Some of the security threats stated are as follows: impersonation or spoofing, black-hole attack, sinkhole attack, wormhole attack, etc. While solutions to these attacks are classified under the following categories: trust values, wormhole detection method, intrusion detection systems (IDS), credibility management and routing test, multi-factor authentication techniques and so on.

Ali *et. al.* [3] also surveyed security challenges in mobile adhoc networks, they introduced three important security parameters and further divided security aspects into two, which are security services and attacks. The security services are classified into five important services which are used to protect the network before attacks happen, while attacks are the threats to the network. In addition, they analyzed and discussed the mitigating approaches against the attacks in MANETs. Mojtaba *et. al.* [4] investigated routing attacks and various solutions to such attacks. They highlighted vulnerable security attacks that MANETs routing protocols are subject to and identified mechanisms such as cryptography scheme, key management and special hardware using GPS as some of the solutions to such attacks.

Allegedly *et. al.* [5] proposed a new malicious nodes detection scheme to detect packet faking by malicious node. In this type of attack malicious nodes drop one or more packets and inject another packet to replace the dropped packet. They introduced the hash chain technique to detect the attack and trace the malicious nodes. There approach is compared against the acknowledgment based mechanisms and the network coding based mechanism. Kannhavong *et. al.* [6] surveyed routing attacks in mobile ad hoc networks. They investigated various security issues in MANETs and examined routing attacks, such as flooding, black hole, wormhole, replay, link spoofing, and colluding attacks, as well as solutions against such attacks in MANETs. They identified the advantages and drawbacks of the solutions reviewed, then recommended the improvement of the effectiveness of the security schemes they surveyed.

Papadimitratos *et. al.* [7] proposed a secure link state protocol (SLSP) for mobile ad hoc networks to secure neighbor discovery and adopted neighbor lookup protocol to further strengthen their system against DoS attacks. In addition, the proposed SLSP restricted the forwarding of packets within a cluster of the source node and adopted the use of public and private keys to validate that packets are only forwarded within the cluster. Unlike our proposed monitoring scheme, their protocol only focused on securing the topology discovery and protected the link state update packets but does not secure the routing of packets. Our newly proposed scheme addresses routing security using a monitoring mechanism to protect packets and also guarantees the communication of benign nodes. Another main difference in our work is that our proposed scheme secures the routing protocol against colluding attacks.

To secure packet route, however, Papadimitratos *et. al.* [8] proposed a different mechanism from their previous work to secure message transmission in MANETs. Their mechanism is based on four main schemes: secure end-to-end transmission of packets and feedback, dispersion of a packet, multi-path routing of packets and adaptation to topology changes. In their protocol, the source node will first select node disjointed paths that are valid (referred to as an active path set (APS)). Then splits the packet into a number of pieces, which is transmitted simultaneously across the selected multiple APS. The destination node after receiving a sufficient number of pieces of the divided packet will then reconstruct the packet, even when a fraction of the pieces is dropped or invalid. In a situation where a piece of the packet is not received by the destination, such route is considered as broken or compromised. In addition, their mechanism also introduced path rating based on the feedback from the destination node. Paths that are below the given threshold are discarded from the network. Their secure protocol focused on detecting unsecured routes unlike our approach in which the actual malicious nodes in a selected route are detected and discarded from further relaying of packets.

Although some of the proposed schemes successfully mitigate routing attacks, however, they are either too expensive for resource constrained networks or the solution provided is not applicable to colluding attacks from malicious nodes. Also, it is possible for malicious nodes to drop packets and attribute the cause to poor communication links. Therefore, we propose a mechanism to analyze the action of all nodes in the network. Specifically, our schemes will focus on mitigating Byzantine attacks in link state routing protocols.

3. Overview of Routing Protocol and Byzantine Attacks

In this first section, we describe the link state routing protocols (LSR) and Byzantine attacks.

3.1 Link State Routing Protocols (LSR)

Link state routing (LSR) protocols are proactive protocols in which a node creates a topology of the network and position itself at the root of the tree. LSR protocols are based on Shortest Path First (SPF) algorithm, (also known as Dijkstra's Algorithm) to find the best path to a destination. There is no hop count limit in LSR protocols. Examples of an LSR protocols are open shortest path first (OSPF) and intermediate system to intermediate system (IS-IS).

In LSR, each node finds out the status and the cost of their neighbors' links. Then creates a map of the network showing how nodes are connected to each other. The information is then broadcast to the entire network. Using the best logical path, each node, then calculates its best path to every possible destination. A node can then collect its best paths to each destination to form its routing table. When a node link status change, a routing update called a Link-State Advertisement (LSA) is exchanged between nodes. When each node receives an LSA routing update, the link-state algorithm is used to recalculate the shortest path to affected destinations.

Detecting an outsider attack on the packets exchanged by neighbor can be avoided by using cryptography schemes such as digital signature, hash chain, etc. However, it is difficult to prevent insider attacks such as falsifying routing information, packets dropping, packets faking and other Byzantine attacks.

3.2 Byzantine Attacks

Byzantine attacks can be described as an attack in which malicious nodes take control of the network resources, and disrupt the network performance [1]. The malicious nodes can either selectively drop packets, corrupt routing information or send packets on non-optimal paths. These types of attacks are difficult to detect when carried out by a fully authenticated node in the network. Some of the Byzantine attacks are as follows:

3.2.1 Corruption of Routing Table Attacks

Here the goal of a malicious node is to corrupt the routing table information by falsifying neighbor information or capture and modify the neighbors' link information broadcast by the benign node. Thereby causing the routing protocols to maintain wrong information in the routing tables which now includes the malicious nodes in almost all routes to destinations.

3.2.2 Wormhole Attacks

In this form of attack, a malicious node advertises an artificial route as the best path to the destination node and tunnel the packets to another malicious node, thereby causing the source node to ignore the genuine route. Such malicious nodes can either drop the packet, or selectively drop the packet. Therefore, preventing timely delivery of packets and causes packet lost in the network. This is also a form of colluding attack. An example of a Byzantine wormhole attack is illustrated in Figure 1.

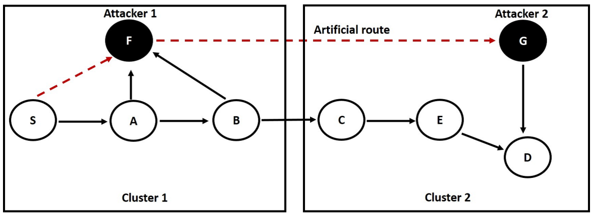


Fig. 1 Wormhole attack using cluster

3.2.3 Black Hole Attacks

In this form of attack, a malicious node injects fake routing information to attract all packets to itself and drop all of them, modify some packets or selectively drop packets. Although to avoid detection such malicious node sometimes actively participate in the routing of packets to the destination in a normal way. Therefore other nodes in the network will find it difficult to detect such malicious node action.

3.2.4 Sink Hole Attacks

In sink hole attack, a malicious node attracts all packets to itself by claiming to have shortest path to all destinations in the network. Other intermediate nodes relay their packets through such malicious node. The malicious node can either modify, fabricate or eavesdrop on the packets.

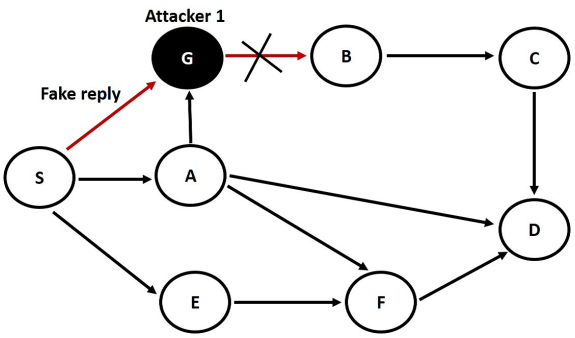


Fig. 2 Black hole attack

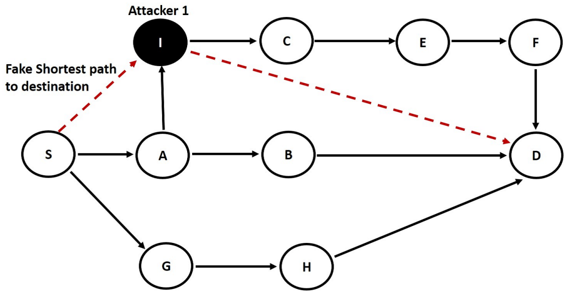


Fig. 3 Sink hole attack

3.2.5 Falsifying Location Information Attacks

In this attack a malicious node forges a position in the network which is completely different from its actual position and report the forged position to other nodes in the network. This causes other benign nodes in the network to get wrong status and cost of such malicious node's link which leads to invalid information in the routing table and hence packets lost.

These types of Byzantine attacks are difficult to detect or prevented, especially when carried out by an insider attacker. Therefore, we adopt a monitoring scheme to secure routing in LSR protocol.

3.2.6 Colluding Attacks

In a colluding attack, a group of nodes collaborates to carry out a Byzantine attack by dropping or modify packets. One of the nodes will advertise itself as having the shortest path to the destination. The shortest path may or may not include other collaborating nodes to complete the attack. This form of attack is hard to detect, especially when the nodes align each other as neighbors. For example, in Figure 4, the source node S decides to send a packet to destination D. The best shortest path is S - F - B - D, however, since node F is malicious and colluding with another malicious node C, the route will be S - F - C - E - D with the intermediate nodes F and C colluding.

4. Proposed Secure Routing Protocol Based on Monitoring Scheme

In this section we describe our secure routing protocol using

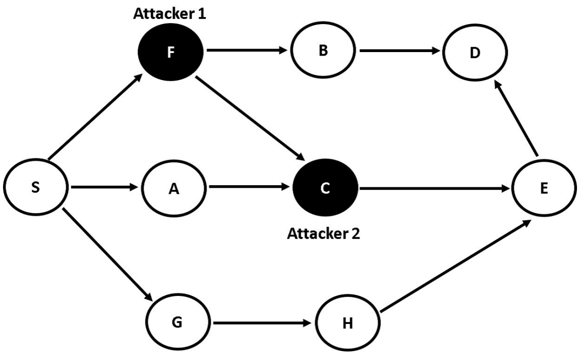


Fig. 4 Colluding attack

monitoring based scheme to protect against colluding malicious nodes. First, we explain how a valid routing table is formed, then we describe the statistical method for detecting malicious nodes and the monitoring scheme to secure LSR protocols. Finally, we explain how our proposed scheme mitigates Byzantine attacks in LSR protocols.

4.1 Assumptions

We make the following assumptions about our monitoring scheme.

- All benign nodes are connected in the network topology.
- All nodes have public and private keys.
- All links are not stable. i.e. Not all packets are received by the neighboring nodes.
- All benign nodes know the link state of other benign nodes.
- Change of probability for packet dropping is low.
- A packet is dropped with probability q due to wireless channel fading during transmission between two benign nodes.

4.2 Routing Table Formation

In our LSR protocol, each node broadcasts hello messages to its neighbors periodically at every 10 seconds with a dead interval of 30 seconds. The dead interval is used to confirm if a node is still alive. If a node fails to receive hello messages from a particular neighbor within a dead interval, then that neighboring node is considered as being disconnected from the network. The hello message includes the node's ID *(ID)*, digital signature $ID_{K^{-1}}$, node's position ID_{GPS}, the node's hello message hash value $ID_{H(h)}$, predicted packet dropping threshold PPD_T, number of packets sent P_S, number of packets received P_R, number of packets forwarded P_F, a timestamp T_S, the list of neighbors. In addition, each node collects hello message from their neighbor, calculates the hash value of the collected hello message and include the hash value, the neighbor's ID, timestamp, the neighbor's digital signature to their own hello message. Figure 5 shows a typical example of the additional information in the hello message which we specifically introduced for achieving routing security.

Each node floods the link state information of its neighbor to other nodes in the network. As part of flooding, we use acknowledgment and retransmission because links are not reliable. Each node maintains its routing table through the neighbor information in the hello message. Since benign nodes are connected, all neighbor information reach all nodes. After a node collects all topology information, each node calculates the best logical path to every possible destination with the information collected from the hello messages. Then it uses the best paths to each destination to form its routing table.

Hello Message

ID
ID_K^{-1}
ID_{GPS}
$ID_{H(h)}$
PPD_T
P_S
P_R
P_F
T_S
<u>Neighbor List:</u>
$ID_{N1} - H(h) - T_S$
$ID_{N2} - H(h) - T_S$
$ID_{N5} - H(h) - T_S$
$ID_{N7} - H(h) - T_S$

Fig. 5 An example of hello message

After neighbor nodes receive hello messages, each neighbor node responds to the hello message by sending an acknowledgment to confirm receiving such hello message. Within the replies, each neighbor node identifies itself with its node ID, digital signature, and position. The node that initiates the hello message can use the information from the neighbors to confirm that of the hello messages collected.

A node will record the information of neighbors it is directly connected to into its hello message and gets the sub-neighbor list through the flooded link state information of other nodes it receives. For example, let's consider a scenario using the network in figure 6 below, for node *S* to get the complete network topology, node *S* first will form a partial network, which include its direct neighbors *A* and *C* while node *A* will have only node *B* and node *C* with *B* and *E*. The same steps apply to other nodes in the network. After receiving the flooded hello messages, node *S* will get the full network topology which includes all its sub-neighbors and adds them in its hello message.

Also, each node authenticates each other with public and private key scheme. The keys can be safely generated by any nodes. The public and private keys are unique to each benign node and the private key is kept secret by each node. Benign nodes exchanged public keys beforehand. When a new hello message is received by a node from its neighbor, the node after authenticating such neighboring node with its public key and node's ID will then check the timestamp and the hash value of its last hello message with the new hello message to confirm if the old hello message is not replayed.

4.3 Monitoring Scheme for LSR Protocol

In LSR protocol, to send a packet from a source to destination the routing protocol finds the shortest path to such destination using the information in the source node's routing table. However,

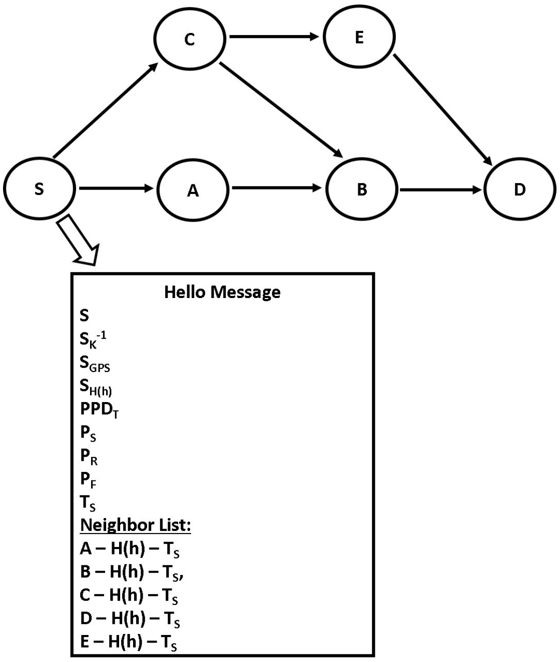


Fig. 6 An hello message with complete network information

the route may be subjected to an attack from a malicious node that is included in the route to the destination. To prevent such attacks by detecting such malicious node we introduce a statistical method and a mutual monitoring scheme.

4.3.1 Detecting Malicious Nodes

Let's consider a situation as in Figure 7, in which node S is sending a packet to destination *D* with *S* - *F* - *G* - *D* as the shortest path to destination.

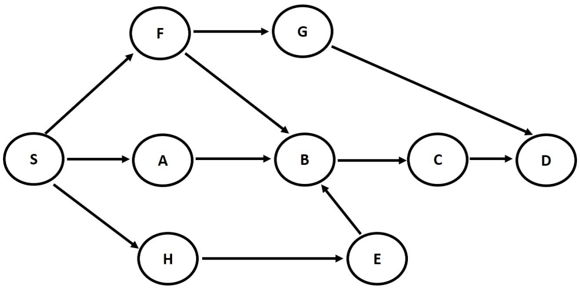


Fig. 7 Packet route in a network

To detect a malicious node in the network, we use an event-driven monitoring block to record the packet route history of packets, which will allow other nodes to know the past events of the packets sent. The details of the event-driven monitoring block is explained in 4.3.3. For example, to route a packet from node *S* to destination *D*, node *F* sign its signature ($S_{K_F^{-1}}$) on the monitoring block for receiving and forwarding the packet to the next hop node *G*. Similarly, node *G* sign its signature for receiving the packet from node *F* and for forwarding the packet to destination node *D* on the monitoring block. The destination node *D* however, will only sign its signature for receiving the packet. However, it is possible for a node to carry out a malicious action, such as dropping packets and claimed otherwise. Therefore, we record the IDs of nodes in the monitoring block when they accept and forward packets. An example of an event driven monitoring block is illustrated in Figure 8.

Packet Route	Next Hop Node ID	Receiving Node Sig	Forwarding Node Sig	Receiving HOP ID	Forwarding HOP ID
F	↓	↓			
G	↓	↓			
D					

Fig. 8 An example of monitoring block

In addition, we use a statistical method, that is explained in the next section, to confirm a node is not a benign node. In this method, nodes monitoring packets detect a node that drops packets too many as a malicious node. The aim of our statistical method is to analyze the packet dropping behavior of each node while the event-driven monitoring block is to confirm at which node the malicious action is carried out. These methods are used to detect malicious node in the network.

Other nodes (e.g. *S*, *B* and *G*) in the network will first observe the packet from node *F* as described in 4.3.2. Then monitors the monitoring block (see Figure 9) to verify if it is correctly signed by node *F*. The same process applies to packets from node *G* to destination node *D*. In a situation where node *F* accepts the packet and fails to forward such packet to node *G*, node *F's* ID for forwarding the packet will not be recorded in the monitoring block and this can be detected by the monitoring nodes. If the statistical method confirms node *F* is not a benign node and the block is not correctly signed, the result of the monitoring is reported to other nodes in the network. Therefore, the malicious action of node *F* is detected in the network.

Packet Route	Next Hop Node ID	Receiving Node Sig	Forwarding Node Sig	Receiving HOP ID	Forwarding HOP ID
F	G	$S_{K_F}^{-1}$	$S_{K_F}^{-1}$	F	F
G					
D					

Fig. 9 Monitored block

4.3.2 Observing packet dropping

A monitor node observes the packet dropping behavior of a monitored node and adopts the approach of statistical hypothesis testing to tell if the monitored node is a malicious node.

The approach of statistical hypothesis testing: First, the monitor node makes a null hypothesis H_0 that the node being monitored is a benign node and sets the value of significance level α (as a common practice $\alpha = 5\%$). Second, the monitor node observes the monitored node for N packets and counts the number n_d of packets dropped by the monitored node. Third, the monitor node calculates the P value p using the following formula

$$p = \sum_{i=n_d}^{N} \binom{N}{i} q^i (1-q)^{N-i}. \tag{1}$$

If $p \leq \alpha$, the monitor node rejects the null hypothesis H_0, meaning that the monitored node is detected as a malicious node. Otherwise, the monitor node accepts the null hypothesis H_0.

The whole process is summarized in Algorithm 1.

Algorithm 1 Malicious node detecting

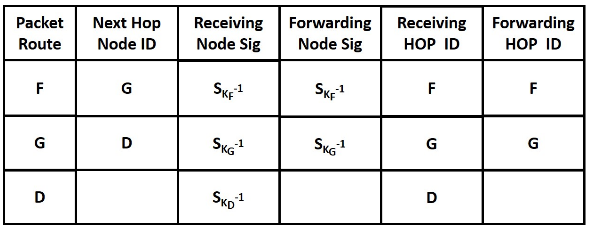
```
Input: $q$ : the probability of a packet being dropped
       $\alpha$ : level of significance
       $N$ : sample size of observed packets
Variables: $n_d$ : the number of dropped packets
       $p$ : P value
       $j$ : counter
Output: Reject $H_0$ or Accept $H_0$
$n_d \leftarrow 0$;
$j \leftarrow 1$;
while $j \leq N$ do
    The monitor node observes how the monitored node handles a received packet not destined for himself;
    $j \leftarrow j + 1$;
    if The monitored node drops the received packet then
        $n_d \leftarrow n_d + 1$;
    end if
end while
Calculate $p$ according to (1);
if $p \leq \alpha$ then
    return Reject $H_0$;
else
    return Accept $H_0$;
end if
```

4.3.3 Event-Driven Monitoring Block

Each data packet maintains a route history in an event-driven monitoring block, which records all events occurring to these packets, like packet receiving, forwarding, etc. Each neighboring node can overhear this packet and check the validity of information in this block. In our scheme to secure the routing protocol we monitor the action of every node in the network. To achieve this a node will create an event driven monitoring block which is added to the packet header. The event-driven monitoring block consist of the packet route, next hop node ID, receiving node signature, forwarding node signature, receiving hop ID and the forwarding hop ID. Intermediate nodes on the route to destination append their signatures on the event-driven monitoring block when they receive the packet and before forwarding the packet to the next hop intermediate node. The signature serves as a confirmation for accepting and forwarding a packet.

However, it is possible for an intermediate node to sign the block without forwarding the packet. To avoid this we introduce an event driven marking of the intermediate node's ID on a separate field created in the block. The hop ID marking is triggered by the intermediate node's action. Accepting a packet will trigger the marking of the receiving hop ID field while forwarding a packet trigger the marking of the forwarding hop ID field. The monitoring block also contains the source node signature and each field used for the signature of the intermediate node is time stamped. This will allow the monitoring nodes to know the delay at each node and to prevent modification (see Figure 10 for a completely filled monitoring block based on the example in section 4.3.1).

Apart from recording each intermediate node signature and IDs in the event-driven monitoring block, we maintain the packet route information (selected shortest path) in the event-driven monitoring block to prevent a malicious node from using a non-optimal route to forward packets. Also, each intermediate node record the next hop node that the packet is forwarded to. This will prevent forwarding of packets to the wrong next hop node. In addition, the history of each node activity (packet dropping, the packet sent and the packet forwarded) is observed and added to the hello message.



Packet Route	Next Hop Node ID	Receiving Node Sig	Forwarding Node Sig	Receiving HOP ID	Forwarding HOP ID
F	G	$S_{K_F}{}^{-1}$	$S_{K_F}{}^{-1}$	F	F
G	D	$S_{K_G}{}^{-1}$	$S_{K_G}{}^{-1}$	G	G
D		$S_{K_D}{}^{-1}$		D	

Fig. 10 A valid monitoring block

4.4 Preventing Byzantine Attack

In this section we explain how our monitoring scheme prevents Byzantine attacks. Here we only focus on packet dropping, the corruption of the routing table and colluding attacks.

4.4.1 Preventing Packet Dropping

If a node actively selected to take part in the routing of packets from source to destination decides to carry out a wormhole attack as shown in Figure 1 by tunneling the packet to another node or other malicious node in the network which eventually drop such packet or selectively drop the packet. The source node *S*, node *A*, and node *B* will detect such action by observing how node *F* handles the received packet, then calculates the P value and compares the P value with the observed packet drop rate of node *F*. The monitoring nodes further checks the monitoring block signed by the malicious node to know the past activities of the malicious node. Also, the next hop node ID is confirmed against the packet route field to validate if the node the packet is forwarded to is part of the original route selected. If the observed packet dropping rate is greater than the P value and the next hop ID is not valid or not included in the rate (packet route field), then the monitoring nodes can share the information of the malicious node *F* to other nodes. Hence the malicious node is removed from subsequent routing.

In addition, if the malicious node decides to selectively drop, the forwarding hop ID will not be marked and monitoring nodes can detect such action. The same process applies to the other Byzantine attack that involves dropping of packet or selective dropping of packet such as black hole attack.

4.4.2 Corruption of Routing Table

In a situation where a malicious node tries to corrupt the routing table information by falsifying neighbor information or decide not to add a node as a neighbor in its hello message. Other benign nodes in the network will get a conflicting neighbor information from the nodes that are connected to such malicious node. To prevent this type of attack, benign nodes that receive the conflicting information can first check the neighbor list in its hello message against the neighbor list they have added to the hello message in the past to confirm if there is a previous connection between such malicious node and the excluded node. Also, benign nodes can compare the hash value of the last hello message received from the malicious node against the hash value of the last hello message the excluded node received from the malicious node. If the excluded node is in the neighbor list of the malicious node and the hash value of the last hello message received by the benign nodes and the excluded node from the malicious node is the same, this confirms that there is a previous connection between the malicious node and the excluded node. Then the malicious node is detected for altering the neighbors information in its hello message. Hence, the malicious node is excluded from the route.

4.4.3 Preventing Colluding Attacks

In our scheme, we only address colluding attacks in which only one of the colluding parties is part of the selected packet route and the malicious node forwards the packets to another node on a non-optimal path. To detect and prevent such colluding attacks in which a group of nodes collaborate to forward a packet to each other, we use the packet route and next hop ID. The nodes that serve as monitoring nodes can confirm if the packet is forwarded to a valid next hop node by checking the next hop node ID against the pre-selected packet route in the event-driven monitoring block. In addition, if the colluding nodes are in the original valid route and the packet is dropped, the event monitoring block will show at which hop the packet is dropped. Colluding malicious nodes cannot modify the event driven block to change the next hop node's ID or mark the receiving and forwarding hop ID as the process is handled by the monitoring scheme algorithm and each field is time stamped.

5. Evaluation

In this section, we run simulations to evaluate the performance of our monitor scheme. Specifically, we focus on the following common performance metrics.

Successful detection ratio: It is the ratio between the number of successfully detected malicious nodes and the number of all malicious nodes.

False positive ratio: It is the ratio between the number of honest nodes falsely detected as malicious nodes and the number of all honest nodes. It is desirable to have a low false positive ratio.

Delivery ratio: It is the ratio between the number of packets successfully delivered to destinations and the number of packets generated by sources. Not all packets can be successfully delivered to destinations due to reasons like malicious node dropping packets, buffer overflow, etc.

Packet delay: It is the time interval from the time a packet is generated to the time the packet is delivered to its destination.

6. Conclusion

In this paper, we proposed monitoring scheme to secure link state routing against Byzantine attacks. According to this scheme, nodes can mutually monitor each to detect misbehavior in the network by checking the predicted packet dropping rate against the actual packet dropping rate. Also, the monitoring nodes confirm each node signature on the event driven monitoring block to detect if a node drop a packet or tunnel a packet to other malicious nodes. Thereby detecting the Byzantine action of malicious nodes on the LSR routing protocol.

Future work includes detecting and preventing other colluding scenarios such as when malicious nodes are neighbors to each other and all are part of the selected packet route, source node or destination node is malicious or both are malicious. Other attacks to be considered are packet delay in which an attacker advertises non-optimal paths, packet delay by colluding malicious nodes and false reporting of a benign node as an attacker.

Acknowledgment

This work is supported by the JSPS KAKENHI Grant Numbers 16K01288, 16K12421, 15K15981 and Otsuka Foundation scholarship reference number 16-S38.

References

[1] Geetha, A., and Sreenath, N.: Byzantine Attacks and its Security Measures in Mobile Adhoc Networks, *(IJCCIE 2016)*, Int'l Journal of Computing, Communications and Instrumentation Engineering, Vol. 3, Issue 1, 2016.

[2] Harshavardhan, K.: A Survey on Security Issues in Ad Hoc Routing Protocols and their Mitigation Techniques, International Journal of Advanced Networking and Application, Vol. 03, Issue 05, pp. 1338-1351, March-April, 2012.

[3] Ali, D., Seyed, K., Esmaeil, K.: Security Challenges in Mobile Ad hoc Networks: A Survey, *(IJCSES 2015)* International Journal of Computer Science and Engineering, Vol. 6, No. 1, February, (2015).

[4] Mojtaba, S., Imran, G., Aida, H., and Seung, J.: Routing Attacks in Mobile Adhoc Networks: An Overview, Science International (Lahore), Vol. 25, No. 4, pp. 1031–1034, 2013.

[5] Alajeely, M., Ahmad, A., and Doss, R.: Malicious Node Detection in OppNets using Hash Chain Technique, 4th International Conference on Computer Science and Network Technology (ICCSNT 2015), Vol. 1. IEEE, 2015.

[6] Kannhavong, B., Nakayama, H., Nemoto, Y., Kato, N., and Jamalipour, A.: A survey of routing attacks in mobile ad hoc networks, in IEEE Wireless Communications, Vol. 14, no. 5, pp. 85–91, October 2007.

[7] Papadimitratos, P., and Haas, Z., J.: Secure link state routing for mobile ad hoc networks, in Proceedings of the IEEE Workshop on Security and Assurance in Ad hoc Networks, in conjunction with the 2003 Symposium on Applications and the Internet, pp. 379–383, January, 2003.

[8] Papadimitratos, P., and Haas, Z., J.: Secure data transmission in mobile ad hoc networks, in Proceedings of the 2nd ACM workshop on Wireless security *(WiSe '03)*, ACM, New York, NY, USA, 41–50, 2003.