Detection and Avoidance of Wormhole Attacks in Connected Vehicles

Sami S. Albouq and Erik M. Fredericks
Oakland University
Rochester Hills, MI 48309
{ssalbouq,fredericks}@oakland.edu

ABSTRACT

The routing protocol in connected vehicles (CVs) is vulnerable to wormhole attacks where attackers can deceive legitimate nodes and purport them as if they are immediate or close neighbors. The Ad hoc On-Demand Distance Vector (AODV) protocol is a routing packet protocol designed for mobile nodes. However, AODV may not have been explicitly developed with security issues, specifically wormhole attacks, in mind, thereby requiring a detection algorithm to secure route establishment. This paper introduces the Wormhole-Protocol-Detector (WPD), a lightweight protocol for detecting and mitigating wormhole attacks. WPD is designed to work on a highway that is clustered into equal segments and equipped with road side units (RSUs) in predefined locations to monitor each segment. WPD consists of three phases: monitoring and detection of out-of-range packets, estimating the hop count between source and destination nodes, and identification of nodes participating in a wormhole connection. Together, these phases enable legitimate nodes to avoid the wormhole link and obtain secure routing paths between CVs. To validate our approach, we apply WPD to a CVs simulation where different types and lengths of wormhole, including a new wormhole attack method, can be applied to the CV network. Our experimental results suggest that WPD can detect wormhole attacks with a high detection rate and minimum false positives.

CCS CONCEPTS

• **Connected vehicle communication** → **Routing security**;

KEYWORDS

Wormhole attacks; Connected vehicle; security; routing

1 INTRODUCTION

Connected vehicles (CVs) comprise an ad-hoc network [3] similar to mobile ad-hoc networks (MANET). However, characteristics such as high mobility and frequent updates to a dynamic network typology can introduce additional network complexities [16]. Wireless communication channels are vulnerable to a wide range of security attacks due to their open nature, lack of fixed infrastructure in which nodes can form a network autonomously without the need for pre-deployed communication infrastructures (e.g., base-stations and access points), and the hostile environments where adversaries may exist. A wormhole is one such security attack that has been proven to have serious consequences on many proposed routing protocols in ad-hoc networks [2][10][19]. The wormhole attack typically requires the presence of at least two colluding nodes that are geographically separated and connected via a tunnel, where the intent is to deceive legitimate nodes and purport them as if they are neighbors. In this paper, we present the Wormhole-Protocol-Detector (WPD), a lightweight protocol to automatically detect and mitigate wormhole attacks.

It is generally difficult to detect the wormhole attack while in an ad-hoc network because it is infrastructure-less and decentralized [5][18]. The use of either strong encryption or authentication will not solve the problem since the attacker can function as a legitimate node [10]. To this end, different techniques, such as leashes or specialized hardware, can be used to detect a wormhole attack. A leash functions by adding information to a packet in order to restrict the distance or time that the packet is allowed to travel [18][27]. However, leash techniques may require clock synchronization on location accuracy and therefore cannot always detect physical layer wormholes [18]. Other techniques rely on special hardware, such as radio-frequency, ultrasound, or directional antennas to determine the location of the attacker [13][26]. Protocols that rely on directional antennas add complexity and require additional customization. As a result, CVs require a feasible technique that can oversee packets and automatically detect a wormhole attack.

In this paper, we introduce WPD, a lightweight protocol that monitors packets and measures the lowest possible number of nodes that can connect a source to destination by using road side units (RSUs). An RSU is a stationary device that verifies routing packets and can be used to identify nodes connected to a wormhole, thereby avoiding the attack. The protocol does not require any additional requirements such as specialized hardware or clock synchronization to ensure that it is both simple and easy to adapt. WPD works purely based on local connectivity information that is maintained by each node including neighborhood information (i.e, immediate neighbors) where nodes may belong to several neighborhoods. Thereafter, this information can be requested and verified by a specific node in the routing path to detect and avoid the wormhole links that have been established between a source and a destination.

WPD leverages packet monitoring, lower greatest hop bound measurement, and secure path verification to detect and avoid wormhole attacks. Packet monitoring enables legitimate nodes to inspect incoming packets and detect suspicious packets, where a suspicious packet indicates a wormhole establishment attack. In

DIVANet'17, November 21–25, 2017, Miami, FL, USA

ACM ISBN 978-1-4503-5164-5/17/11…$15.00
https://doi.org/10.1145/3132340.3132346

this case, the node can request assistance from the RSU for further verification. The RSU can provide additional information to boost detection accuracy by finding both source and destination locations and determining the possible hops between them. Next, nodes exchange their local neighborhood information, such as routing data, to detect unsecured neighbors by checking the misconstrued (i.e., the tunnel) path that the wormhole created. When the misconstrued path is detected, WPD suggests different nodes to reconstruct some of the paths in their routing table to avoid all suspicious links.

We demonstrate the use of WPD by applying it to a highway-based CVs simulation that uses the ad hoc on-demand distance vector (AODV) protocol for establishing a route between two non-neighbor nodes and a malicious attacker that uses its position to deceive legitimate nodes. Experimental results suggest that WPD can detect and avoid the wormhole attacks with a minimum number of false positives. The remainder of this paper is organized as follows. Section 2 provides background information on CVs, AODV, and wormhole attacks. Section 3 describes the WPD technique. Section 4 then describes our experimental setup and results. Following, Section 5 overviews related work. Lastly, Section 6 discusses our findings and presents future directions.

2 BACKGROUND

This section presents background material on CVs, the AODV routing protocol, and the wormhole attack specifications we consider in this paper.

2.1 Connected Vehicles

CVs are a subclass of MANET where the type of networks are adaptive, self-organizing, infrastructure-less, and do not require any centralized administration. CVs have distinctive characteristics, such as nodes with high mobility and varying network traffic patterns [3]. There are generally two main entities in CVs: vehicles and RSUs. Vehicles are intelligent mobile nodes equipped with different sensors. For example, global positioning systems (GPS) provide positioning information and distance sensors measure distances between the vehicle and other objects. RSUs are stationary entities that are located at predefined positions to enhance communication and provide additional computing power. CVs provide two types of communication between nodes: vehicle to vehicle (V2V) and vehicle to RSU (V2U). Thus, nodes of CVs depend on themselves for implementing any network functionality due to the decentralization and infrastructure-less networks [2]. As a result, many routing protocols were proposed to maximize connection periods between nodes, such as AODV and Dynamic Source Routing (DSR) [14]. However, this paper explicitly focuses on the AODV protocol.

2.2 Ad Hoc On-Demand Distance Vector

AODV is a reactive routing protocol that has been developed for mobile ad-hoc networks and has been adopted by CVs [23]. In order to use the AODV protocol in CVs, links between nodes must be bidirectional to initiate routing operations. Thus, nodes do not lie on an active path; they either do not maintain routing information or do not participate in any periodic routing table exchange. A node does not need to discover and maintain a route to another node until the two nodes plan to communicate, or if a former node offers its service as an intermediate forwarding terminal. When a local connection is needed to a particular node, all neighboring nodes can become aware of each other through the use of several techniques, including a broadcast message known as a *HELLO* message or through a route request (RREQ) message.

Using a RREQ message, the source node sends a message containing the IDs of both the originator and destination. The RREQ is broadcast through the network until a node that has a known route to the destination or the destination itself, responds with a route reply (RREP) message. The most important field in the RREQ for the AODV protocol is the hop count for node discovery, where hop count indicates the number of hops traversed by the RREQ message from source to destination and is used to determine the shortest path to the destination node leading to faster communication.

2.3 Wormhole Attack Specification

A wormhole attack typically requires the presence of at least two colluding nodes that are in separate physical locations to establish a wormhole link. This link is used to tunnel packets between legitimate nodes via attackers to convince them that they are close neighbors. Attackers can be either *passive* or *active* while launching attacks, where the malicious node may choose to include themselves or not in the routing operations, respectively. We next describe common methods of wormhole attacks [2]: packet encapsulates, out-of-band channel, high power transmission, and packet relay.

2.3.1 ***Packet Encapsulation.*** In this method, the wormhole attackers try to prevent intermediate nodes from modifying the hop count field in the RREQ and ensure this packet looks attractive by having fewer hop counts upon arriving at the destination. Figure 1(a) demonstrates the packet encapsulates attack, where a malicious Node W_1 encapsulates a packet received from its neighboring Node A and sends the packet to its colluding malicious Node W_2. Once the packet is received by Node W_2, it decapsulates it and broadcasts to its neighborhood. Thus, the original packet is not modified by the intermediate nodes between Node W_1 and Node W_2, as the intermediate nodes cannot open the packet due to the encapsulation. The only modification occurs when the packet is sent again from Node W_2 to Node B.

2.3.2 ***Out-of-Band Channel (OBC).*** In this method, malicious nodes are equipped with long-range high-bandwidth wireless links that operate at different frequencies to transmit packets from source to destination without interference from intermediate nodes [23]. Figure 1(b) shows the established tunnel between Node W_1 and Node W_2, where the malicious nodes have specialized hardware for packet exchange. This method is more difficult in terms of the establishment compared to other wormhole methods (e.g., packet encapsulation) as it requires additional hardware and complex configurations to launch an attack. However, attackers will not include their information in the packet and just retransmit the packet to Node B. This wormhole serves as a fast link between the source and destination to trick the nodes into thinking that they are one hop away. As a result, Node B assumes that Node A is its immediate neighbor when it receives a RREQ from it through the wormhole tunnel, which looks faster with a shorter hop count.

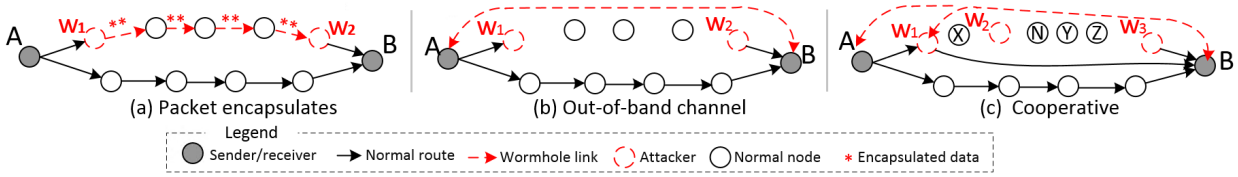


Figure 1: Demonstration of wormhole attacks.

2.3.3 ***Packet Relay.*** In this method, a malicious node relays packets between two nodes to convince them that they are immediate neighbors. This type of attack can be established by either one or more malicious nodes using the same channel and without the need for a special hardware as in the OBC method. More malicious nodes in the path can incur maximum damage due to the expanding of the neighbors list of a victim nodes. This type of attack can be launched when the malicious Node W_1 is located within transmission range of the legitimate nodes A and B, where Node A and Node B are not immediate neighbors [19]. Node W_1 tunnels packets between Node A and B to invisibly purport them as if they are neighbors. The attacker does not advertise its identity in the routing operations.

2.3.4 ***High Power Transmission.*** In this method, the wormhole attacker uses a strong power transmission to send a long range signal that could reach to the destination or a node closer to the destination. This increases the chance to be on the routes established between the source and destination. The goal of the attacker is to minimize number of hops and lower the latency of packet delivery to attract network traffic. This type of attack can be launched using one malicious node. In this method, the attacker is visible to the source during the routing establishment [15][18][20]. Note this method may not work correctly if the attacker cannot receive the RREP.

2.4 Cooperative Wormhole Attack

We now introduce a new type of wormhole, a cooperative wormhole attack, that maximizes the damage to the network and makes attacking nodes appear more attractive. In this attack, the malicious nodes use two or more different methods to establish the wormhole link between source and destination nodes. The establishment of the type of wormhole link could be prepared between colluded attackers or unintentionally exist between malicious nodes. To the best of our knowledge, we are the first who introduce the cooperative wormhole attack. In this type of attack, the attacker can be either passive or half-passive in term of sharing their identities. If the attackers use the passive technique, then both ends that connect to the wormhole link exchange packets as if they are one hop away, because the attackers serve as a link between nodes. When attackers choose to be half-passive, one of the attackers will participate in the routing as a legitimate node and use the wormhole link to deliver the packets with a smaller number of hops.

For example, a high power transmission attacker may not be able to establish a wormhole link using the AODV protocol. This happens when the attacker sends a very long range RREQ, but cannot receive the RREP due to the different transmission range. Thus, it may cooperate with other attackers (e.g., OBC) to form the wormhole link. For instance, in Figure 1(c), Node A wants to establish a route to Node B. Node A will broadcast a RREQ to all its adjacent neighbours. When Node W_1 receives the RREQ, it uses its high power transmission and broadcasts the RREQ to the destination. The RREQ will arrive to Node W_2 and Node B. Note that the RREQ arrives to Node B in a short amount of time. When the RREQ arrives to Node W_2, Node W_2 will tunnel the packet to its colluded Node W_3 that then transmits the packet to Node B. When the RREQ is transmitted through the tunnel, it may arrive later than the high power transmission packet. If the packet uses this path, its latency arrival will increase as it will have been transmitted by two attackers, whereas the high power transmission only uses one. However, Node W_1 will be the faster transmitter and its packets will arrive sooner to Node B, and therefore Node B will discard other RREQs. Once Node B receives the RREQ, it will send back a RREP. Node W_3 hears the RREP and transmits it to Node W_1 via Node W_2, which then sends it to the destination node. Note this type of attack may confuse existing detection techniques that rely on time analysis [6][16][25], especially if the attackers use a cut-through method (i.e, the packet is sent before the whole frame has been received) [21].

3 APPROACH

This section introduces the WPD technique. First, the assumptions are stated for the network, type of nodes, and attackers. Then, a description is provided of how WPD works to automatically detect a wormhole attack and identify nodes that connect to the wormhole link. We also describe a mechanism that assists nodes in avoiding the wormhole.

3.1 Assumptions and Network Model

In this section, we state our assumptions about the network specifications, type of nodes, and establishment methods.

Network: Communications between nodes are bidirectional to correctly exchange RREQ/RREP packets. For example, if Node X can hear Node Y, then Node Y can also hear Node X. As a result, nodes in the network have the same communication range $\tilde{r}$.

Nodes: We assume there are two types of nodes in CV networks: vehicles and RSUs. Vehicles are mobile nodes with various speeds and built-in intelligent features (c.f., Section 2.1.). Vehicles can determine their positions with normal accuracy errors that range between 3-6m [8]. RSUs are stationary devices that are located in roads to form sequence of clusters that ensure a connection between nodes and RSUs. RSUs connect to each other via high speed links and can perform operations such as calculations and packet verification. According to the Dedicated Short Range Communications (DSRC) specification, both types of node can provide a transmission range of up to 1000m for a channel [12][4].

Attack Model: We assume that malicious nodes can establish the wormhole link at arbitrary locations in the network and choose the type of the wormhole. The wormhole attackers do not use their

IDs or MAC addresses when they use a passive-attack method; otherwise acting as a normal node with more powerful capabilities than normal nodes.

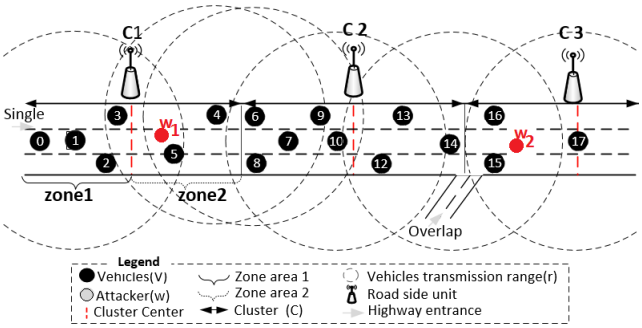


Figure 2: Demonstration model of a highway network.

Connected Vehicles Network Model: In this paper, we consider a highway scenario for building a proposed solution for the network. We assume the highway is built on a number of static clusters where an RSU node, called a cluster head (CH), represents each cluster. Every cluster consists of a CH and a number of cluster members (i.e., vehicles) that are within the range of the CH. All clusters have the same fixed size and CHs positioned at predefined locations. Figure 2 presents a section of highway, comprising overlapping and single areas, two CHs, and several vehicles. For example, if there is a highway of length d, then the minimum number of CHs required to cover the entire highway are $\mu = \frac{d}{\tilde{r}}$, meaning the placement of CHs would be sequential on the highway. As result, every CH covers a fixed segment of the road, and every node belongs to at most one cluster. This model is suitable for a highway similar to I-95 in the US. It starts from the south coast and ends at the northeast coast. It is not necessary to have all the highway equipped with RSUs, but possible in the areas where the traffic is high. For example, the I-95 highway between New Jersey and New York has a length of 43 miles and the average annual daily traffic flow ranges between 150,000 to 250,000 vehicles [1].

A vehicle may join a new segment of a road from a single or an overlapped zone (c.f., Figure 2). For example, when a vehicle enters a cluster from a single zone area, it only needs to send a join request (JREQ) packet to the CH. Then, the CH will accept the request and send with a join reply (JREP) packet that contains three main attributes: a cluster head ID, location (e.g., latitude and longitude) and the immediate adjacent-cluster IDs. The location attribute is used by vehicles to identify their positions in the cluster by calculating their distance from the CH center and determine their position. Every packet is required to include the cluster ID and node's position during routing operations. When a vehicle enters the highway from an overlapped zone, it is required to broadcast a JREQ to all CHs in the overlapped zone. This packet includes the vehicle's identity, speed, position and direction, which helps an appropriate CH to identify the vehicle that will join it soon and replies with a JREP. After a period of time, a node may arrive at a point where it needs to leave the cluster and join another cluster. Hence, the node must send two types of packets: leaving-cluster and JREQ. When the node sends a leaving-cluster packet, the CH removes the node from its routing table and includes it in its history members table.

Notations: Consider a CV network as shown in Figure 2, consisting of clusters represented by $C = \{c_1, c_2, ..., c_y\}$. Each $c_y \in C$ has a CH position c_y^{pos}, divided into two zones $c_y^{zone1} < c_y^{pos} \leq c_y^{zone2}$, and has vehicles denoted by the set $V = \{v_0, v_1, v_2, ..., v_i\}$. Let P denote the set of pairs from source to destination $P \subset V \times V$; note that there are at most $v \times (v-1)$ pairs in P. For each $(i,j) \in P$, we let $R_{i \to j}^{q}$ denote the set of v that AODV selected during the routing discovery period and q represents a specific node from that set. When v_i wants to communicate with v_z that is not in the communication range $\tilde{r}$, it initiates an AODV route discovery packet to be processed and possibly retransmitted by other nodes. During packet transmission, every node needs to include two important attributes beside AODV's required attributes. These attributes are the cluster ID (c_y) and vehicle position (v_i^{pos}) that inform the next destination that the packet is coming from a close neighbor of the same or an adjacent zone. To determine v_i^{pos}, a vehicle needs to know its position in $c_y = \{c_y^{zone_1}, c_y^{zone_2}\}$. If the node position is $c_y^{zone_1}$, then $v_i^{pos} = c_y^{zone_1}$; otherwise $v_i^{pos} = c_y^{zone_2}$. For the wormhole attack, we defined W_z to represent the attacker.

3.2 Wormhole Detection

In this section, we describe our protocol technique comprising three phases: monitoring packets, lower greatest hop bound, and path verification. We next describe each phase in detail.

3.2.1 ***Monitoring Packets.*** In this phase of the protocol, each vehicle monitors packets that have been received from neighboring nodes to discover wormhole packets that come from unexpected zones and notify its CH. As mentioned earlier, every node needs to include two important attributes (i.e., (v_i^{pos}, c_y^{id})) when they send a packet to allow the next hop to know from where it originated. Consequently, before considering any received packets in routing operations, a node is required to check these attributes of the packet to ensure it is sent from a valid zone cluster. During monitoring, a node expects to receive packets from either the same cluster or an adjacent cluster's zone depending on the node's position in the cluster. Figure 3 demonstrates the monitoring packets. For example, if a node's position is in CH position c_y^{pos}, it will only communicate with nodes that are also in the same cluster. However, if the node moves in any direction towards a new cluster, it will slightly lose communication coverage from its current cluster and start covering a zone in an adjacent cluster.

If a node received a packet from a non-adjacent zone, it is considered to be a malicious attack as it violates the expected traveling path of the packet. Thus, the legitimate node needs to send an alarm to the CH for further inspection. Typically, legitimate nodes send packets within the standard communication range (i.e., $1000m$), unlike the wormhole attackers [12]. For example, assume a highway as shown in Figure 2 consists of $C = \{c_1, c_2, c_3\}$, and V=$\{v_0, ..., v_{17}\}$. Notice that every c_i has a set of v_i that are located in different positions of each $c_i^{zone\#}$; for instance, c_3=$\{v_{15}^{pos=1}, v_{16}^{pos=1}, v_{17}^{pos=2}, W_1^{pos=1}\}$. Assume Node v_1 wants to communicate with Node v_{17} that is not in the communication range. Before Node v_1 creates a RREQ packet, it needs to determine v_1^{pos} to include it in the RREQ packet. Next, Node v_1 can create the RREQ that includes AODV

attributes, including $v_1^{pos=zone1}$ and c_1^{id}, to broadcast it to adjacent neighbors (ignore W_1 and W_2 for now). Once the RREQs are received by legitimate neighbors such as Nodes (v_0, v_2, v_3), they check the two attributes of the RREQs to ensure they come from known zones. If the RREQs come from a familiar zone, then the receiver node accepts the RREQs as normal packets. If we assume the attackers exist and use one of the wormhole methods described in Section 2.3, the attacker will transmit the RREQs through the wormhole directly, skipping the intermediate nodes. *This means that all legitimate nodes that connect to the wormhole link will not participate in creating the routes.* Hence, Node v_{17} will receive the RREQs from a non-adjacent cluster, indicating a malicious attack. When Node v_{17} checks the RREQs, it will find them coming from unexpected zones that are not from the same cluster or an adjacent cluster's zone. Consequently, Node v_{17} will report this issue to its CH. To demonstrate this process, we will next present sample scenarios for monitoring wormhole attacks.

Out-of-Band Channel example: Assume that two attackers W_1 and W_2 wish to establish a wormhole link between Node v_1 and Node v_{17} by transmitting the RREQ packet over a high-bandwidth wireless link. When Node W_1 receives the RREQ from Node v_1, whose attributes are $v_1^{pos=zone1}$ and c_1, it forwards the RREQ to Node W_2 that is not in an adjacent cluster. Node W_2 rebroadcasts the RREQ to the final destination v_{17}. When the RREQ packet is received by Nodes v_{15}, v_{16} and v_{17}, they check the attributes $v_1^{pos=zone1}$ and c_1 to find that the RREQ comes from an unfamiliar cluster and then report that to their CH.

High Power Transmission example: This method can be detected using the normal travel path violation and the assumption of the symmetric bidirectional link where the destination can only hear from the attacker. When the attacker uses the high power transmission method, a malicious node W_1 tries to receive the RREQ packet and forwards the packet to its final destination, or to cross multiple hops between a source and destination nodes to introduce itself in the path. Let us consider the previous scenario, but omit Node W_2 from the network, as this method requires only one node to establish. When Node v_1 sends the RREQ packet to Node v_{17}, Node W_1 will receive the RREQ and send it to nodes that are in cluster c_2 or c_3. Once the RREQ is received by one of the appropriate nodes (i.e., v_{12},..., v_{17}), they will recognize the RREQ coming from a non-adjacent zone or non-adjacent clusters such in c_3 that is considered as a malicious attack and reported to the CH. In this method, the attacker cannot establish a route to the destination. Similarly, the cooperative wormhole method can be detected using the normal travel path violation because the destination can reply packets to the attacker through a hidden wormhole link. Thus, the destination and attacker have bidirectional links and can be detected.

Wormhole placement: The placement of the wormhole attacks will not affect the monitoring detection in most cases if the wormhole is long. In the OBC, High Power Transmission, Cooperative, or Relay Packet wormhole, attackers possibly place themselves either in adjacent zones or non-adjacent clusters. When the attacker places themselves in adjacent zones, legitimate nodes that are in the same cluster will not find the wormhole link attractive since the normal path may differ from the wormhole link in at most one hop, even if we place nodes as far apart as possible in the cluster. However, when two attackers place themselves in non-adjacent clusters, they form a link that is attractive and violate the normal path construction that can easily be detected similar to the scenario in Figure 2. In case of the single attacker, the intuition is similar to the previous ones. The attacker wants to attract as many packets as possible from the network traffic by retransmitting long distance packets that then violate the normal traveling path, leading to detection by WPD.

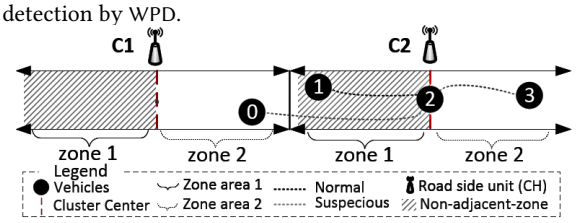


Figure 3: Monitoring packets.

3.2.2 ***Lower Greatest Hop Bound.*** In this phase of the protocol, the CH verifies the existence of the wormhole attack by estimating the lower greatest hop bound (LGHB) between a source and a destination, where LGHB is the minimum number of hops between nodes. Once the CH receives a wormhole-alarm RREQ packet from the cluster members, it extracts the source and destination IDs and composes them in a new location verification packet that is then sent to all adjacent CHs. The CH whose routing table contains the source or destination IDs sends a location inquiry to that particular (i.e., source or destination) node. Thereafter, this node needs to reply with its most recent location to its CH. Then, the CH will forward the location to the destination's CH to complete the calculation. When the CH of the destination receives the location of both nodes, it will calculate the distance and LGHB on the relation between the distance and the standard communication range to compare it with the wormhole route path length. This will ensure detection of the wormhole attack by discovering if the tunnel distance is longer than the LGHB.

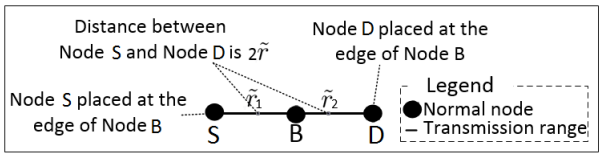


Figure 4: The minimum number of hops on a good path.

Assume L_s denotes the source location and L_d denotes the destination location (ignoring the accuracy error for now). The traveled distance of a packet from the source to the destination node locations is $L_{sd} = \|L_s\text{-}L_d\|$. Since the communication range is standard for all nodes in the CVs, LGHB for a packet to travel from the source to destination is $h=\lceil L_{sd}/\tilde{r} \rceil$. Consequently, if $h < \lceil L_{sd}/\tilde{r} \rceil$, then there is a wormhole on the path.

THEOREM 3.1. *If $h < \lceil L_{sd}/\tilde{r} \rceil$, then a wormhole exists on the route.*

PROOF. *We will prove the existence of a wormhole link by proving the minimum hop count on a good path is at least $\lceil L_{sd}/\tilde{r} \rceil$. The minimum number of hops on a good path occurs when nodes are placed on a straight line and at the boundary of the communication ranges as shown in Figure 4. Since two adjacent nodes cannot be placed further than $\tilde{r}$, total distance traveled is at most $h\tilde{r}$, which implies*

$h \geq \lceil L_{sd}/\tilde{r} \rceil$. *Thus, if* $h < \lceil L_{sd}/\tilde{r} \rceil$, *then there is a wormhole link of length greater than* $\tilde{r}$ *on the path.*

Consider the previous scenario on Figure 2 where Nodes v_1 and v_{17} needed to communicate with each other by sending the RREQ through the wormhole link and the packet is monitored and detected as an attempted attack. c_3 first requests both nodes' locations to calculate the LGHB between Node v_1 and Node v_{17}. Assume c_3 received and calculated the distance between Node v_1 and Node v_{17} to be $L_{sd} = 2250m$ and $\tilde{r} = 500m$. Consequently, the packet required at least $\lceil 2250m/500m \rceil = 5$ hops to travel between Node v_1 and Node v_{17}. If the calculation result value is lower than 5 hops, then c_3 needs to inform the destination and CH that there is a wormhole in the path and needs to create a RREP packet that collects affected node IDs.

The effect of the location accuracy error is negligible. In very few cases some good short paths may not be discovered. Let h be the hop count, L_{sd} is the real distance, and $\hat{L}_{sd}$ is the distance with the maximum error accuracy between source and destination. If L_s and L_d are the recorded locations, then $||L_s - L_d||$ - $2\epsilon \leq \hat{L}_{sd} \leq ||L_s - L_d|| + 2\epsilon$, where ϵ denoted the maximum error accuracy. If $L_{sd} = \hat{L}_{sd}$ - 2ϵ, then we are lowering the bound of the normal path that leads to an undetected wormhole link as there may be a hop count greater than $\hat{L}_{sd}$ - 2ϵ. On the other hand, if $L_{sd} = \hat{L}_{sd}$ + 2ϵ, then we are increasing the bound of the normal path leading to minor false positives.

3.2.3 ***Path Inspection.*** In this phase of the protocol, we employ path inspection to detect the attacker's exact location and identify non-neighboring nodes to employ mitigation strategies. The proposed solution exploits hop count variations between neighboring nodes along a path from a source S to a destination D to identify the start and end of the wormhole link, including locating the specific node(s) that connect to the wormhole link. This technique consists of two processes: neighbor discovery and neighbor verification.

Neighbor discovery: The node that performs the detection is called a *verifier* and discovers all one-hop neighbors that are within communication range. For example, in Figure 2, the normal one-hop neighbors of Node v_1 are (v_0, v_2, v_3, v_5). The established wormhole link between Node W_1 and Node W_2 purports nodes to think that they are immediate neighbors as they do not know the source location of the packets. However, because the highway is divided into clusters and every node belongs to a specific CH, the neighboring list will not contain neighbors from non-adjacent clusters (i.e., nodes that are behind the wormhole link). The clustering technique and the additional attributes in the packets allows nodes to identify their immediate neighbors. As a result, the number of inspected nodes will minimally increase due to the exclusion of the most nodes that are within the wormhole transmission range, excepting the *verifier*. For example, when Node v_1 broadcasts a *HELLO* packet to determine its one-hop neighboring list, every node that receives the *HELLO* packet will reply with its ID. If no wormhole link exists, the neighboring list will be inclusive to all immediate neighbors. If a wormhole link does appear in the network, it is possible that non-neighboring nodes respond to the request. Depending on the location of the *verifier* from the wormhole, some nodes may be additionally included in the neighborhood list.

There are two scenarios for receiving packets that depend on the wormhole existence and its type. For example, consider Figure 2. If an OBD wormhole link exists, then Nodes (v_{16}, v_{15}, v_{17}) will reply their IDs to v_1, that will then be discarded due to violation of the expected, "normal" path. However, if the cooperation wormhole link exists, it may add some nodes to the neighborhood list that are not within the communication range of v_1, but are in one of the adjacent clusters. This will not affect the *neighbors verification process* because these nodes will result in a smaller hop count that is not valuable enough to produce a discrepancy between routes.

Neighbors verification: The *verifier* specifies either a one- or two-hops away *examiner node* E (c.f., Figure 5), where $E \in R_{S \to D}$, and searches for an alternative route that does not pass through the wormhole link to the examiner. This process will show a significant variation in hop count between nodes if a wormhole link exists. The selection of Node E depends on the distance in hop count of the $verifier$ to the destination. Thus, if a node connects to a wormhole link in one end and needs to verify the path route against the possibility of wormhole existence, it specifies the examined node (which could be in either end of the wormhole link) and performs the *neighbors verification* to identify the variation of hop count between nodes and where a wormhole link is shorter than normal links. Thus, the start and end of the wormhole link can be identified in a route based on the discrepancy between normal and attacker's hop count.

Figure 5 demonstrates how both processes work together, let Nodes a, b, $f \in R_{S \to D}$–they are nodes on the path from S (i.e., source node) to D (i.e., destination node) obtained from the RREP. Let β_i be the set of one-hop neighbors of the Node i and $h_{i,j}$ length of hops between Node i to j. Let the wormhole link of $W_1 \leftrightarrows W_2$ connects a and b, where $a \in \beta_{W_1}$ and $b \in \beta_{W_2}$. Let f be the next hop from b on the path $R_{S \to D}$. Clearly, a and b are separated by a distance greater than $\tilde{r}$ and connected via a wormhole link. To start inspecting nodes over the path, the *verifier* (sender or source node) needs to determine the examiner node E in the path. There are two situations for selecting the examiner node E: one-hop or two-hops away from the *verifier*. If $E = D$, then the *verifier* selects E as a one-hop node; otherwise two-hops away (c.f.,Figure 5). In the illustrative example:

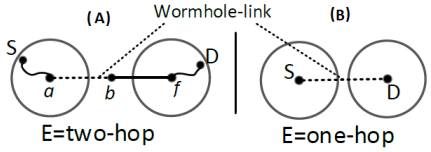


Figure 5: Examiner node selection.

(1) The *verifier* sets E as a two-hops away node (see Figure 5), i.e., $E = R^3_{S \to D}$.
(2) The *verifier* performs the process of the *neighbors discovery* and obtains β_S.
(3) Then, the *verifier* will broadcast the list β_S and E to all immediate nodes and ask every node $\{\beta_S$ - $R_{S \to D}\}$ to find a route to E where the route does not include nodes in β_S. Thus, every node will determine $h_{i,E}$ and forward them to the *verifier*.
(4) Once the *verifier* receives $h_{i,E}$ from β_i, it selects the longest hop-count H and tests whether E is connected to the wormhole link or not. If $H-2 > \gamma$, then E is connected to the wormhole

link, where γ represents the normal alternative route lengths. Selecting an appropriate value of γ is described in Section 3.3.

(5) If no wormhole link exists in the first inspection, S will select $R^2_{S\rightarrow D}$ to be the next *verifier* and $E = R^4_{S\rightarrow D}$. Note the Node 4 is two hops away from the new $verifier$ Node 2.

(6) If the new *verifier* is not the last node before D, then repeat Steps **(1)** to **(5)**. Note, the *neighbors verification* detects the node that connects to the wormhole link. Consequently, in the illustrative example, $a \in \beta_{W_1}$ will perform Step **(1)** to **(5)**. Node a will complete the process of the *neighbor discovery* to obtain $\beta_a \subset \beta_{W_1}$, but $\beta_a \notin \beta_{W_2}$ as described in the *neighbors discovery* section. Next, a specifies E=f and asks all $\beta_a \notin R_{a\rightarrow f}$ to find a route to E that believe to be two hops away from the *verifier*. Hence, $z \in \beta_a$ performs the *neighbors verification* and reports $h_{z,f}$ to a. Once a receives $h_{z,f}$, it determines H=$max(h_{z,f})$ and verifies $H - 2 > \gamma$ to conclude that a and b connected thought the wormhole link.

To further illustrate this process, consider Figure 2. When Node v_0 receives the RREP that includes all immediate nodes IDs to establish $R_{v_0\rightarrow v_{17}}$, it extracts all the IDs and performs the route inspection. The RREP will contain these nodes (v_0, v_3, v_{17}) as the path from Node v_0 to Node v_{17} where attackers do not advertise themselves in the routing packets. When Node v_0 inspects the route path, Node v_0 becomes the *verifier* and sets $E = v_{17}$ (Step **(1)**). Then, Node v_0 performs the process of the *neighbors discovery* to obtain β_{v_0} (Step **(2)**). When all one-hop neighbors of Node v_0 reply to Node v_0, β_{v_0} will contains (v_1, v_2, v_3). Node v_0 will then broadcast (β_{v_0},E) to all one-hop neighbors v_1, v_2 and v_3 and ask them to find a route to v_{17} without passing by Nodes v_0, v_1, v_2 and v_3 (Step **(3)**). These nodes will find routes to E as follows: Node v_1 = (v_5, v_4, v_7, v_{12}, v_{14}, v_{16}), Node v_2 = (v_5, v_8, v_{10}, v_{13}, v_{16}) and Node v_3 = (v_4, v_9, v_{13}, v_{16}). Thereafter, Node v_0 will receive hop lengths (v_1 = 6, v_2 = 5, v_3 = 4) from its immediate nodes. The *verifier* selects the longest hop, which is v_1 = 6, and calculates H−2 > γ (6-2 = 4 > 1, 2, or 3, which are the possible lengths of the alternative routes) (Step **(4)**).

Similarly, in the High Power Transmission and cooperative wormhole, we determine the *verifier* that selects E from the path. Once this step is finished, the *verifier* obtains its one-hop neighbors list by performing *neighbors discovery*. Next, it will send both the one-hop neighbors list and E to all immediate neighbors and ask them to find an alternative route to E that does not go through the established path between the source and destination. Immediate nodes determine the hop length to E and send them to the *verifier* to examine the longest path against the possible existence of the wormhole attack starting from the neighbor of the *verifier* and ending at E.

3.3 Effect of γ

The value of γ represents the alternative route lengths between the intermediate nodes only; we exclude both source and second-hop node to find the maximum possible intermediate nodes. Thus, we find the value of γ ranges between 0-3 when the distance between the source and the second-hop node is equal the communication range. However, when a wormhole exists, the distance between the source and second-hop node will be longer. As a result, the alternative routes will not match the normal possible alternative routes.

Choosing γ is critical and depends on the length of the normal route that is observed from the LGHB in comparison with the wormhole link. Thus, γ needs to be selected carefully to avoid a large number of false positives. If γ is selected as a small value and the wormhole link is long, then the number of false positives will increase significantly, while if γ is large and the wormhole link is small, some short wormhole links may escape the detection. In our detection technique, we select γ in relation to the value of LGHB, which gives a reasonable comparison with the alternative routes.

3.4 Wormhole link Avoidance

It is very difficult to totally isolate the wormhole link nodes in a dynamic network topology because attackers are mobile and move from one location to another quickly. Some legitimate nodes may get affected and disconnected from the network due to attacker isolation, especially when the attackers do not advertise their identity. Therefore, when the attackers do not share their IDs and only operate as connectors between two areas, it is better to perform an avoidance technique.

This phase is intended to nullify the wormhole unobtrusively and ensure nodes are continually connected. When the sender identifies the start and end nodes that connect to the wormhole link, their IDs are reported to the local CH. Then, the CH broadcasts both IDs to all other CHs to search for those nodes in their respective routing tables. Once the CH can identify the wormhole link's start or end node, it will instruct other nodes to reconstruct new routes and rediscover all their one-hop neighbors again to avoid traversing back to the wormhole link. This technique guarantees that nodes will have a safe route to their immediate neighbors and ignore the wormhole link, thereby minimizing damage to the network.

4 EXPERIMENTAL RESULTS

We next describe our experimental setup and present our results from applying the WPD protocol to a simulated CV network.

4.1 Experiment Setup

In this section, we present our experiment setup that was used to demonstrate the effectiveness of the WPD protocol in detecting and mitigating wormhole attacks. For this experiment, we compare a network protected by WPD to one that is not. The simulation comprised a highway model divided into several segments, where each was based on the vehicle communication range with the relation to the length of highway. In each segment, the RSU is placed at the center to represent the CH. In addition, the CH partitions the segment into two zones, as described in Section 3.1. Vehicles are randomly assigned coordinates and variable speeds, where each has predefined attributes such as direction and movement. The randomly-assigned coordinates are bounded by the length and width of the highway.

Table 1 presents our simulation parameters. The number of vehicles is selected to be 194 to ensure the presence of connectivity between nodes. These vehicles placed randomly and bounded with the highway length of 10km and width of 200m in a fixed average range density in each cluster. The speed of vehicles is set to be between 85-100km to ensure vehicles stay connected and do not partition the network. The highway length was set to be 10kph to

ensure it was long enough to examine different wormhole lengths and included enough segments (i.e., clusters) of 1000m length to equal the transmission range. The highway needed at least 10 RSUs (i.e., CH) to cover all segments. The sender node was randomly chosen from the left-most nodes in the network (nodes position < 2500m) and the destination is randomly chosen from any area at minimum 2 zones away from the sender to satisfy the wormhole condition.

Table 1: Simulation parameters

Parameter	Value
Vehicle speed	85-100kph
#Vehicles	194
#RSUs (CHs)	10
Transmission range	1000m
Highway length	10km
Highway width	200m
Cluster length	1000m

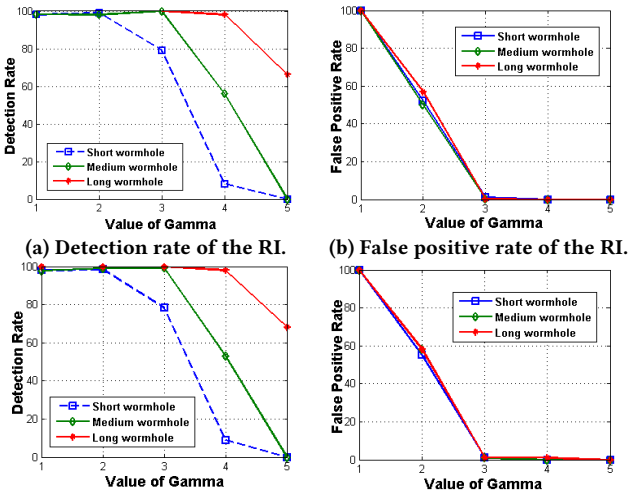
The placement of the wormhole attack depends the length, location, and type of the wormhole link. In our simulations, we have three wormhole link lengths: short, medium and long. For the short wormhole link, the attackers create a link between a source and a destination that covers at most one cluster's zone, while the medium wormhole covers one entire cluster, and the long wormhole covers more than the previous lengths. The type of wormhole (e.g., OBC) is randomly selected following determination of wormhole length. The wormhole attacker is then placed on the highway between the source and destination nodes.

Each simulation runs with a different combination of randomly selected sender-receiver nodes, wormhole type, and wormhole location. We initially performed 380 treatments, but due to invalid configurations, we discarded 38 as they provide no valid wormhole link establishment. The performed experiment was divided into 3 sets based on the wormhole length (i.e., 114 short wormhole, etc.). We conducted two types of experiment: stored object (SO) and random initialization (RI). For the stored object experiment, the CVs simulation was executed and stored all its objects in serializable files that were then used in a second run comprising varying wormhole lengths. For RI, we used the same setup parameters for each SO simulation (e.g., wormhole length, type, location, and number of vehicles), but with randomly-initialized vehicle locations to ensure WPD can identify nodes connected to the wormhole link. In both experimental techniques, we used different values of γ to detect the wormhole link, where the γ varies between 1-5 depending on the wormhole length. Thus, we compare the effectiveness of choosing γ with different wormhole lengths.

4.2 Experimental Results

In this section, we discuss the results from applying WPD to our simulation for detecting the existence of different types of wormhole attack and identifying the nodes that participate in establishing the wormhole link.

Figures 6 presents the detection and false positive rates of both SO and RI simulations of WPD protocol for the three different wormhole lengths and various values of γ. The value of γ plays a major role in detecting which node connects to the wormhole link by indicating the maximum number of intermediate nodes in two hops computation. This occurs by comparing the maximum length of the alternative routes with value of γ to determine from which



(a) Detection rate of the RI. (b) False positive rate of the RI.

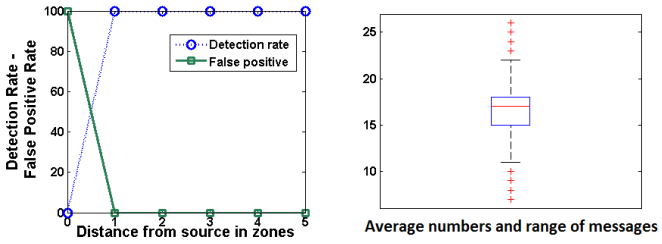
(c) Detection rate of the SO. (d) False positive rate of the SO.

Figure 6: Results of (a and c) are for the Random (R) simulation and (b and d) are for the Stored Object (SO) simulation.

node the wormhole starts. We observed that from Figure 6a and 6c when the wormhole link is short the best value of γ is 2, because all alternative routes should not exceed 2 hops in the worst case scenario between two normal nodes that are within communication range. This also should hold for medium and long wormhole link, but since the alternative routes are significantly greater than γ, we observed that the best value of γ is 3 to avoid a large number of false positives. This indicates that the value of γ results in a discrepancy between alternative routes that enables WPD to detect the wormhole attack.

Figures 6b and 6d demonstrate the false positive rates of the SO and RI simulation. When the wormhole link is short, the value of γ could be very small but will result in very high false positive rates due to the similarity with the alternative routes. For example, in detecting a short wormhole link, if the value of γ = 1, then WPD cannot distinguish between a normal path and a path impacted by a wormhole link. When the value of γ is increased, WPD performs better and ignores the normal routes to identify the suspicious node. For the medium and long wormhole links, the alternative route can show very high variation in comparison with the value of γ.

WPD can detect the existence of the wormhole attack with a high detection rate and 0 false positives when the wormhole link is reasonably long. WPD does not require a large node degree for good performance in detecting the wormhole attack packet because only one node can verify the packet violation as described in Section 3.2.1. The protocol can detect the existence of the wormhole with a 100% detection rate when the source and destination are one zone away, as shown in Figure 7a. WPD also maintains the same detection rate and no false positives as the distance in zones increases between the source and destination nodes. In the simulations, attackers have different communication ranges and can transmit packets for longer distances than normal nodes. Since attackers use their large range transmissions capabilities to transmit packets to a long distance, WPD is able to recognize these packets that come from unknown area with a high detection rate of 100%. Figure 7b shows the number of packets sent by the verifier node during the neighbors verification phase. The minimum number of



(a) Detection and false positive rates of the monitoring packets. **(b) Range of number packets sent by the verifier.**

Figure 7: Simulation results.

packets sent is 7 when neighboring list is small, and the verifier node had some links to the examiner node that obtained from the AODV communication. The number of packets increases as the neighboring list increases and the links between nodes has not been established to be 17 in average and 27 at most. Note the neighboring list depends on the transmission range and node density, so the result of 17 and 27 reflects that the verifier node may joint a highway traffic.

Threats to validity: WPD cannot detect the encapsulation wormhole attack because attackers have the ability to tamper with the information of the packet. For example, if an attacker A tunnels a packet to another attacker B in a different cluster, B will pretend that it received the packet from a secure source and replies with localized cluster information. However, researchers have proposed techniques for enabling receivers of multicast data to verify that the received data originated with the claimed source and was not modified enroute [24].

5 RELATED WORK

In this section, we highlight related work on techniques for detecting wormhole attacks. The wormhole attack was introduced in the wireless networks by Dahill [28], Papadimitratos [22] and Hu *et al.* [11]. As a result, several techniques were proposed to detect, defend, and avoid the wormhole attacks.

5.1 Hardware-based Methods

Several wormhole detection techniques exploit the use of special hardware, such as a directional antenna [9], radio frequency fingerprinting-capable transceivers [17][26], or global positioning system receivers [27][30] to detect the wormhole attack. Hu *et al.* proposed techniques called geographical and temporal leashes to detect the wormhole [9]. Geographical leashes ensure the receiver of the packet is within a specific distance from the sender. Temporal leashes specify an upper boundary to the packet lifetime, restricting the maximum traveling distance of the packets. Similarly, Seyed *et al.* used geographical leashes for avoiding wormhole in VANET and addressed weak points in geographical leashes by using the hop-by-hop efficient authentication protocol (HEAP) to authentication leashes [27]. In both geographical and temporal leashes, nodes need to add authentication data to protect packets against intrusion, thereby increasing overhead in communications and processing. In addition, each node requests a large amount of storage as a result of including the authentication scheme. Hu and Evans presented a technique that uses directional antennas to prevent wormhole attacks [9]. In this approach, every node shares a secret key with other nodes and maintains an updated list of its neighbors that is built using the direction of the signal that is received from its neighbors. To perform this technique, they assumed all antennas on nodes are aligned. However, the requirement of antennas in all nodes may not be feasible and could be difficult in practice, especially in CVs. Consequently, the WPD protocol does not have any hardware requirements.

5.2 Statistical Analysis Methods

Other techniques avoid the use of special hardware and rely on detecting anomalies in round trip time (RTT) [6][16] or via a routing topology [7][17][29]. Most RTT-based techniques can not detect low-latency wormholes, especially if a cut-through technique is used [21]. CapKun *et al.* introduced SECTOR, a set of mechanisms for the secure verification of the time of encounters between nodes in multi-hop wireless networks [6]. Each node estimates the distance to another node by sending a one-bit challenge. Based on response time, the node can detect if the other node is a neighbor or not. This protocol does not require clock synchronization to detect the wormhole, but requires an accurate time measurement with a high precision. The SECUre NeighborhooD (SECUND) [7] and Statistical Wormhole Apprehension using Neighbors (SWAN) [29] techniques each detect wormholes based on anomalies in routing topology. However, SWAN cannot detect wormholes that increases the neighbors list, such as High Power Transmission or Cooperative Wormhole Attack, while SECUND cannot detect the short wormhole attacks. The normal traveling path rule enables WPD to detect short wormholes and attacks that do not necessary increases the neighbors list.

5.3 Neighbor-related Methods

Special guarding nodes with known locations, higher-transmit power and different antenna characteristics, can verify the source of each transmission [15][17]. Issa *et al.* previously introduced LITEWORP, a technique that works with static networks. LITEWORP assumes that there is a guard node within the communication range of any two neighboring nodes along a route. The guard will monitor all traffic from both nodes and check if one does not forward a packet from the other node. Based on its monitoring, the guard node can detect selective forwarding by the wormhole attack. LITEWORP does not require any additional hardware, however, nodes that are chosen to be guards will suffer from overhead due to monitoring and processing of every data received. Lazos *et al.* proposed a cryptography-based solution relying on local broadcast keys and provided a distributed mechanism for establishment in randomly deployed networks [17]. They analytically determined the level of security achieved by their scheme based on spatial statistics theory. However, this solution did not show the establishment of multi-hop pairwise keys or the network scalability issue.

The works closest in scope to ours are Issa *et al.* [15] and Ritesh *et al.* [18]. In Issa *et al.*[15], the protocol used guards to perform the entire detection by monitoring the packets of the network and detect the misbehavior attacks, while our technique used the CH (RSU or guard) as a participant node that can validate the minimum hop count between a source and a destination, which only required the CH interact with a specific nodes during the

detection phase. The protocol in [18] needs centralized topology information in order to detect the wormhole attack. This protocol uses number of independent neighbors to search for forbidden substructures in the connectivity graph that should not be present in a legal conductivity graph. The detection does not guarantee if the number of independent sets does not exist. However, WPD works on a distributed network and does not rely on node density to detect the attack.

6 CONCLUSION

In this paper, we introduced WPD, a protocol for detecting and mitigating wormhole attacks that appear in the AODV protocol for CV networks. WPD is a lightweight protocol that does not require any additional hardware and comprises three phases. The first phase monitors network traffic, where nodes process received packets and determine their validity based on the included packet information. If a suspicious packet is reported to its CH (i.e, RSU), the next phase of the protocol, which is the lower greatest hop bound, will execute. In this phase the RSU determines the minimum possible hop count between the source and destination and informs the destination if a wormhole exists, based on knowledge gained from identifying suspicious nodes, to create a new type of replying packet that makes intermediate nodes include themselves. Following, the inspection phase has each node in the established route discover two-hop neighbors that on the path to identify which nodes connect to the wormhole link. Finally, we demonstrated the validity of our results by a simulation of a CV network in a highway setting.

Future work includes extending WPD to detect the encapsulation wormhole attack [2] and addressing packet authentication during detection phases. Moreover, we intend to examine our protocol in a realistic simulation that uses real highway maps and scenarios (i.e., SUMO with NS3). Finally, we intend to investigate the effectiveness of applying WPD to detect wormhole attacks on the Dynamic Source Routing protocol [14].

7 ACKNOWLEDGMENT

We gratefully acknowledge the assistance from Dr. Dan Steffy in reviewing our mathematical formulae. This work has been supported by the Islamic University in Medina and Oakland University. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of supported by the Islamic University in Medina and Oakland University or other research sponsors.

REFERENCES

[1] 2008. *New Jersey traffic and revenue study.* State of New Jersey Department of Treasury State House.

[2] Mohammed Saeed Al-Kahtani. 2012. Survey on security attacks in Vehicular Ad hoc Networks (VANETs). In *2012 6th International Conference on Signal Processing and Communication Systems*. IEEE, 1–9.

[3] Saif Al-Sultan, Moath M Al-Doori, Ali H Al-Bayatti, and Hussien Zedan. 2014. A comprehensive survey on vehicular Ad Hoc network. *Journal of network and computer applications* 37 (2014), 380–392.

[4] Sami. S. Albouq and E. M. Fredericks. 2017. Lightweight Detection and Isolation of Black Hole Attacks in Connected Vehicles. In *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*. 97–104.

[5] Juhi Biswas, Ajay Gupta, and Dayashankar Singh. 2014. WADP: A wormhole attack detection and prevention technique in MANET using modified AODV routing protocol. In *9th International Conference on Industrial and Information Systems*. IEEE, 1–6.

[6] Srdjan Čapkun, Levente Buttyán, and Jean-Pierre Hubaux. 2003. SECTOR: secure tracking of node encounters in multi-hop wireless networks. In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*. ACM, 21–32.

[7] Thaier Hayajneh, Prashant Krishnamurthy, David Tipper, and Anh Le. 2012. Secure neighborhood creation in wireless ad hoc networks using hop count discrepancies. *Mobile Networks and Applications* 17, 3 (2012), 415–430.

[8] Will Hedgecock, Miklos Maroti, Janos Sallai, Peter Volgyesi, and Akos Ledeczi. 2013. High-accuracy differential tracking of low-cost GPS receivers. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 221–234.

[9] Lingxuan Hu and David Evans. 2004. Using Directional Antennas to Prevent Wormhole Attacks.. In *In Network and Distributed System Security Symposium*.

[10] Yih-Chun Hu, Perrig, and Adrian. 2006. Wormhole attacks in wireless networks. *IEEE Journal on Selected Areas in Communications* 24, 2 (2006), 370–380.

[11] Yih-Chun Hu, Adrian Perrig, and David B Johnson. 2003. Packet leashes: a defense against wormhole attacks in wireless networks. In *INFOCOM. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, Vol. 3. IEEE, 1976–1986.

[12] A. S. T. M. Intl. 2003. Standard specification for telecommunications and information exchange between roadside and vehicle systems-5 GHz band Dedicated Short Range Communications. *ASTM International* (2003).

[13] Shalabh Jain and John S Baras. 2012. Preventing wormhole attacks using physical layer authentication. In *Wireless Communications and Networking Conference*. IEEE, 2712–2717.

[14] David B Johnson and David A Maltz. 1996. Dynamic source routing in ad hoc wireless networks. In *Mobile computing*. Springer, 153–181.

[15] Issa Khalil, Saurabh Bagchi, and Ness B Shroff. 2005. LITEWORP: a lightweight countermeasure for the wormhole attack in multihop wireless networks. In *International Conference on Dependable Systems and Networks. DSN 2005. Proceedings*. IEEE, 612–621.

[16] Dong-uk Kim, Hyo-won Kim, Gisung Kim, and Sehun Kim. 2013. A Counterattack-Detection Scheme in Transmission Time-Based Wormhole Detection Methods. *International Journal of Distributed Sensor Networks* 2013 (2013).

[17] Loukas Lazos, Radha Poovendran, Catherine Meadows, Paul Syverson, and LiWu Chang. 2005. Preventing wormhole attacks on wireless ad hoc networks: a graph theoretic approach. In *Wireless Communications and Networking Conference*, Vol. 2. IEEE, 1193–1199.

[18] Ritesh Maheshwari, Jie Gao, and Samir R Das. 2007. Detecting wormhole attacks in wireless networks using connectivity information. In *26th IEEE International Conference on Computer Communications*. IEEE, 107–115.

[19] Sherif El-Kassas Marianne Azer and Magdy El-Soudani. 2009. Towards Introducing Complex Wormhole Attacks in wireless Ad Hoc Networks "A Full Image of the Wormhole Attacks". *International Journal of Computer Science and Information Security* 1 (2009).

[20] Ali Modirkhazeni, Norafida Ithnin, Mohammed M Kadhum, and Teddy Mantoro. 2012. Mitigation of Wormhole Attack in Wireless Sensor Networks. In *Trustworthy Ubiquitous Computing*. Springer, 109–147.

[21] Lionel M Ni and Philip K McKinley. 1993. A survey of wormhole routing techniques in direct networks. *Computer* 26 (1993), 62–76.

[22] Panos Papadimitratos and Zygmunt J Haas. 2002. Secure routing for mobile ad hoc networks. In *the SCS Commnication Networks and Distributed Systems Modeling and Simulation Conference*. 193–204.

[23] Al-Sakib Khan Pathan. 2010. *Security of self-organizing networks: MANET, WSN, WMN, VANET*. CRC press.

[24] Adrian Perrig, Ran Canetti, Dawn Song, and J Doug Tygar. 2001. Efficient and secure source authentication for multicast. In *Proceedings of the Internet Society Network and Distributed System Security Symposium*. 35–46.

[25] V Karthik Raju and K Vinay Kumar. 2012. A simple and efficient mechanism to detect and avoid wormhole attacks in mobile ad hoc networks. In *2012 International Conference on Computing Sciences*. IEEE, 271–275.

[26] Kasper Bonne Rasmussen and Srdjan Capkun. 2007. Implications of radio fingerprinting on the security of sensor networks. In *Third International Conference on Security and Privacy in Communications Networks and the Workshops*. IEEE, 331–340.

[27] S. M. Safi, A. Movaghar, and M. Mohammadizadeh. 2009. A Novel Approach for Avoiding Wormhole Attacks in VANET. In *Second International Workshop on Computer Science and Engineering*, Vol. 2. 160–165.

[28] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M Belding Royer. 2002. A secure routing protocol for ad hoc networks. In *Proceedings 10th IEEE International Conference on Network Protocols*. IEEE, 78–87.

[29] Sejun Song, Haijie Wu, and Baek-Young Choi. 2012. Statistical wormhole detection for mobile sensor networks. In *Fourth International Conference on Ubiquitous and Future Networks (ICUFN),*. IEEE, 322–327.

[30] Xia Wang and Johnny Wong. 2007. An end-to-end detection of wormhole attack in wireless ad-hoc networks. In *Computer Software and Applications Conference 31st Annual International*, Vol. 1. IEEE, 39–48.