Lightweight Detection and Isolation of Black Hole Attacks in Connected Vehicles

Sami S. Albouq
Oakland University
Rochester Hills, MI, USA
ssalbouq@oakland.edu

Erik M. Fredericks
Oakland University
Rochester Hills, MI, USA
fredericks@oakland.edu

***Abstract*—Connected Vehicles (CVs) can be exposed to black hole attacks that deceive legitimate nodes by falsifying an attractive route to a destination node. This occurs when an attacker sends a packet to the source node confirming the existence of a fresh route. In this paper, we propose a Black Hole Detection Protocol (BlackDP) that works on a highway divided into clusters and monitored by Road Side Units (RSUs) to detect both single and cooperative black hole attacks. Every RSU is tasked with performing both detection and isolation of black hole attacks for their respective highway section after authentication violations and suspicious route establishment activities that have been reported by a legitimate node. The design goal of BlackDP is to decouple the detection process from mobile nodes and construct a trusted semi-centric detection process that can collect needed information for lightweight detection and reliable isolation of malicious nodes. We validate BlackDP in a simulated highway environment to demonstrate its effectiveness.**

I. Introduction

Connected vehicles (CVs) are a form of cyber-physical systems that integrate computation, networking, and physical processes to improve safety and the driving experience. CV nodes communicate with each other to share and exchange information (i.e., traffic and road conditions). However, CVs are vulnerable to a wide range of security attacks due to the open nature of wireless channels, the lack of fixed-infrastructure where nodes autonomously form a network, and the hostile environments in which malicious nodes may exist. A black hole attack poses severe threats to routing protocols of CVs (e.g., the Ad hoc On-Demand Distance Vector (AODV)) [3][17][22]. This attack intends to purport and deceive legitimate nodes by having the shortest and most valid route to the destination. This type of attack is considered a denial of service attack that attracts packets to be dropped. In this paper, we present the Black Hole Detection Protocol (BlackDP), a protocol to detect, mitigate, and isolate both the single and cooperative black hole attacks from networks.

It is generally difficult to detect a black hole attack in an ad hoc network because it is infrastructure-less and decentralized [3][16][25]. While existing secure routing protocols [4][17][26] and intrusion detection systems [11][25] have been developed for detecting black hole attacks, these techniques do not necessarily support a detection mechanism for both single and cooperative attacks. Moreover, some methods rely on a set of peers' behavior assessment [8]. For example, measuring trustworthiness where nodes depend on a voting system to judge other nodes behavior. This technique may not work properly when attackers can participate in voting activities, leading to unreliable reports that possibly harm legitimate nodes and disconnect them from the network. As a result, black hole attackers may be better detected using trusted nodes to directly take action.

BlackDP exploits the existence of Road Side Units (RSUs), which are stationary devices located at predefined positions to enhance communication between nodes. In BlackDP, RSUs are responsible for detecting and isolating both single and cooperative black hole attackers after receiving a threat report from a legitimate node. BlackDP works purely based on local connectivity information that is maintained by each RSU to decouple the detection process from mobile nodes and assign the task to each RSU. RSUs are trusted nodes in the network, and can perform critical operations that mobile nodes cannot, such as isolating misbehaving attackers. It is not safe to rely on mobile nodes to report and cause isolation of other nodes in CV as there can be fake reports from attackers that possibly disconnect legitimate nodes from the network. Thus, RSUs can provide reliable detections and isolations since they are trusted and authenticated from an authority (e.g., Department of Motor Vehicles).

BlackDP works on demand from requests by legitimate nodes in the network after an authentication violation or suspicious route establishment activities. RSUs may have less authentication operations when they communicate with each other unlike mobile entities, which require frequent identity changes and authentications due to the privacy issue (i.e., tracking nodes). The process of detection starts after a legitimate node tries to verify a route establishment and senses abnormal route activities, leading to a report sent to an RSU regarding a specific node that claims to have a fresh and valid route to the destination node. The RSU then processes the verification packet based on the received information from the requester and performs the detection. When the RSU finishes the detection process, it informs the requester(s) about the status of the verification. If an attacker is detected, the RSU will perform an isolation process to disconnect the attacker from the network.

We demonstrate the use of BlackDP by applying it to a simulation of CVs that uses the AODV protocol for establishing a route between two non-neighbor nodes, where

a malicious attacker uses its position to deceive legitimate nodes. Experimental results suggest that BlackDP can detect and isolate the black hole attacker with zero false positives and minimal false negatives in most situations. The remainder of this paper is organized as follows. Section II provides background information on CVs, AODV, and black hole attacks. Section III describes the BlackDP approach. Section IV then describes our experimental setup and results. Following, Section V overviews related work. Lastly, Section VI discusses our findings and presents future directions.

II. Background

This section provides relevant background information on CVs, the AODV protocol, and black hole attacks.

A. Connected Vehicles

CVs are a subclass of ad hoc networks that comprise a collection of nodes equipped with wireless communications and networking capabilities [10]. Nodes can communicate directly with neighbors if they are within transmission range. This type of network is infrastructure-less, self-organizing, adaptive, and does not require any centralized administration to form a network [19]. CVs have two main entities: 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. An RSU is a stationary device that is located at a predefined position to enhance communication and provide additional computing power. For instance, an RSU can connect two nodes that are not in the same communication range or perform a payment process for a specific service provider. There are two types of communications in CVs: vehicle to vehicle (V2V) and vehicle to infrastructure (V2I). According to the Dedicated Short Range Communications (DSRC), both entities can provide a transmission range of up to 1000m [12].

B. Ad Hoc On-Demand Distance Vector

AODV is a reactive protocol for the operation of ad hoc networks and has been adapted by CVs to ease message dissemination (i.e., safety and traffic management messages) [9][20]. Each mobile host operates as a specialized router and routes are obtained as needed (i.e., on demand with little or no reliance on periodic advertisements). The routing operations of AODV generally consist of two phases: route discovery and route maintenance. Route discovery is performed through broadcasting a route request (RREQ) packet. This RREQ may rebroadcast several times via intermediate nodes until it reaches a node that has a route to the destination or the destination itself, which then generates a route reply (RREP). Both RREQ and RREP have important fields such as hop count and sequence number (SN). The hop count indicates the number of intermediate nodes between the source and destination, and the sequence number determines the freshness of the route [3], where route freshness indicates the reliability that valid route exists. Typically, a SN is generated from the destination, and a source node may include a SN in a RREQ demanding a route with a certain level of freshness. The RREP then will be transmitted back to the originator of the RREQ in order to inform the route. Route maintenance is performed with two additional messages: $Hello$ and error route (RRER). Each node broadcasts $Hello$ messages periodically to inform neighbors about its connectivity. The recipient of a $Hello$ message proves that there is an active route towards the originator. However, if a link to the next hop cannot be detected during a period of time out, a RRER packet will be broadcast to inform nodes about a broken link.

C. Black Hole Attack Specifications

A black hole attack is a routing protocol threat executed either by an individual (single) or multiple (cooperative) compromised nodes. In this attack, the dangerous nodes deceive the network by pretending to have the best (i.e., freshest) route between source and destination nodes that are trying to establish a connection. The following sections describe how single and cooperative attacks are performed.

1) Single Black Hole Attack: In this attack, a malicious node receives a RREQ from a legitimate node and quickly claims to have a route to the destination by sending a RREP with a very high SN to convince the sender that it has the freshest route to the destination [3]. For example, in Figure 1(a), suppose Node 1 sent a RREQ for Node 5 with a SN = 0. Neighboring nodes, such as Node 3, that have a route to the destination would receive the request and send RREP packet that contains SN = 20 to reflect the freshness of the route. In contrast, when the attacker B_1 receives the RREQ through Node 2, it would send a RREP packet that contains a very high SN = 120 to ensure it is the highest number of all the RREPs. Note the attacker does not have a route to the destination and does not know the generated SN from the destination, so it tries to set its SN to the highest possible to guarantee its RREP is selected. Based on the information obtained from the RREPs, the source node (Node 1) is certain that the attacker's route is the most convenient one, therefore, it will start broadcasting messages through that route. The attacker then starts its attack, dropping any received messages.

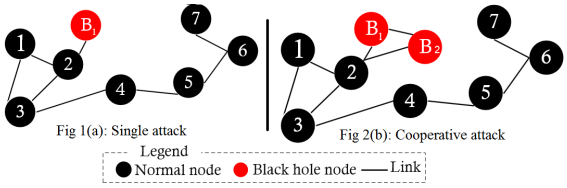

Fig. 1: Black hole attacks

2) Cooperative Black Hole Attack: Following the same strategy as the single attack, two or more malicious nodes cooperate together to initiate the attack (see Figure 1(b)) [22]. Based on an agreement between the attackers, the first attacker (Node B_1) receives the RREQ and replies to the source node with the highest SN, informing the source node that it has the freshest route through the cooperative attacker (Node B_2). Hence, when the source node wants to verify the connection,

it sends another request through a different route to Node B_2 to verify Node B_1's route status. Based on the agreement between the two attackers, B_2 will approve B_1's message to fool the source and assure that it is using a safe route.

III. Approach

This section introduces BlackDP technique. First, the assumptions for the network and nodes are described. Then, a description is provided of how BlackDP works to automatically detect a black hole attack, including a mechanism that isolates the black hole attacker(s).

A. Assumptions and Network Model

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

Network: Node communications are bidirectional, meaning Node A can hear Node B and Node B can hear Node A. As a result, the communication range ($\tilde{r}$) is identical for all nodes in the CV networks. The AODV protocol works correctly if nodes have an identical transmission range that allows them to send and receive RREQ/RREP packets. If the transmission range is not identical, a node may send a RREQ and possibly cannot receive a RREP [7].

Nodes: We assume there are two types of nodes in a CV networks: vehicles and RSUs. Vehicles are mobile nodes with various speeds, included intelligent features, and can determine their positions on the roads. RSUs are stationary devices that connect to each other via high speed links to form sequential static clusters. Every RSU can perform additional operations, such as packet verification. Nodes can provide a transmission range of up to 1000m [12].

Attack Model: We assume malicious nodes can establish their attacks at any arbitrary location in the network and choose the type of black hole attack. We also assume that attackers do not have a route to the destination and try to send a RREP as fast as it can. Moreover, there may be multiple black hole attackers in the network. The attackers respond to any route request while they are in the highway.

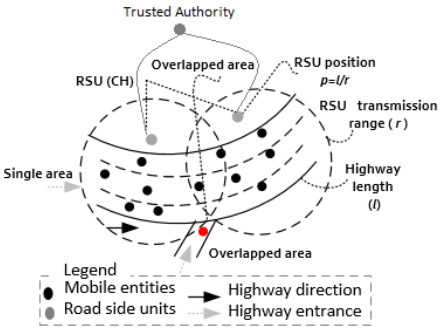

Fig. 2: Highway network model.

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 constructed of several static clusters with RSUs designated as cluster heads (CH) stationed centrally in each cluster, as shown in Figure 2. Each CH is positioned centrally within each cluster, where each cluster is the same size. For example, if we have a highway of length l, then the least number of CHs required to cover the entire highway is $p = \frac{l}{\tilde{r}}$. All CHs are deployed sequentially over the highway to form segments. Each CH covers a segment of the road and maintains a list of members joining and leaving the 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 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 a join reply (JREP) packet that contains information such as the cluster head identity to be included in the packets to allow other nodes know where the packets come from. 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 vehicle's identity, speed, position and direction, which helps an appropriate CH to identify the new vehicle 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 joins another cluster. The node must then send two types of packets: leaving cluster and JREQ. When the node sends a leaving cluster packet, the CH moves the node from its routing table to its history members table. A vehicle must register in a CH in order to have services such as identity renewal and traffic information.

Security and Privacy: CV networks require privacy and security to hide a driver's identity and authenticate nodes and messages. To address this issue in our model, we assume that a Trusted Authority (TA) exists and acts as a root of trust in the network (e.g., Department of Motor Vehicles). The TA is responsible for generating public/private cryptographic keys (K^+/K^-), certificates (CR), and Temporary Pseudonymous Identifications (id) as defined by IEEE Std 1609.2 [2]. A certificate generally contains a public key and node's identity that can be confirmed or validated by the available TA public key K^+_{TA}. TA also is responsible for distributing generated information securely to nodes. Since our network model consists of several regions covered by RSUs, TA can leverage fog computing (a cloud computing extension to the edge computing) to hierarchically deploy nodes close to the RSUs to generate and revoke certificates quickly and securely [5]. For example, a subset of RSUs can link to a TA node that can renew vehicle certificates periodically for several regions to avoid being tracked. These certificates are then used to authenticate node identity during secure neighbor discovery and source/destination authentications. Secure neighbors discovery is outside the scope of this paper, but it is worth addressing to clarify the difference to pure AODV

authentication. The secure neighbor discovery authentication is mainly concerned about immediate node verification by validating their positions, speeds and identities, while the AODV protocol is concerned about source and destination authentications as it works on demand over a subset of intermediate authenticated nodes. Hence, we assume nodes can perform secure neighbor discovery by mutual authentication when two nodes are within the transmission range of each other, while source and destination authentication will be addressed by BlackDP during the route discovery.

Notations: Consider a CVs network, as shown in Figure 3, consisting of clusters represented by C =$\{c_1, c_2,...,c_y\}$ and TA =$\{ta_1, ta_2,...,ta_z\}$ as trusted authorities. Every $ta_z \in TA$ has $x \subset C$ where z is the identity of TA. Each $c_y \in C$ has vehicles denoted by the set V = $\{ v_1, v_2,..., v_i \}$. We denote $v_i^{c_y}$ as a vehicle i that belongs to C_y. When a node issues a RREQ and receives a RREP, the node that claims it has a route to the destination with a high sequence number is denoted as v_B. Note that i, y, and B represent the identity of a CH or vehicle. We refer to a packet detection as d^{req}.

B. Black Hole Detection

In this section, we describe the two phases of BlackDP: black hole node identification and isolation.

1) Black Hole Node Identification Phase: This phase of the protocol comprises two steps: source and destination verification and suspicious node examination.

Source and Destination Verification: In this step, the originator node verifies the route information after authenticating the destination. There will be two scenarios after sending the route discovery request: an intermediate node (possibly an attacker or a normal node) sends a RREP claiming to have a route the destination or the destination itself sends a RREP. In both scenarios, nodes need to authenticate themselves to the originator node and provide route information for verification. When the destination sends the RREP, the originator node can directly verify the identity and route status, while if the intermediate node sends the RREP, the originator node needs to send a $Hello$ packet to the destination to verify the route information along with identity.

If the destination v_d sends the RREP, it needs to provide the originator node v_i with a secure RREP that contains its certificate, which includes the public key K_d^+, id, and expiration time of the certificate to allow Node v_i to verify its identity. However, this is not sufficient because attackers can tamper with the content of the RREP or impersonate another node identity when the RREP is being transmitted back to Node v_i. To solve this issue, Node v_d needs to digitally sign the RREP by using one-way hash function (e.g., SHA-256) to generate a hash code for the message and then encrypt the hash code with its private key K_d^-. Node v_d then attaches the encrypted hash value with the RREP to ensure the integrity of the packet and confirm its identity. In this paper, we will use the term secure packet to indicate the previous message authentication process. When Node v_i receives the RREP, it uses the authority public key K_{TA}^+ to decrypt the certificate and extract K_d^+. Next, Node v_i calculates a hash value for the received RREP, decrypts the signature using K_d^+, and compares the calculated hash value to the decrypted hash value. If the two hash values match, Node v_i is assured that the message must have been signed by Node v_d, thereby the SN and hop count are valid.

When the attacker or intermediate node responds to the RREQ, it needs to send a secure RREP similar to the destination scenario for authentication purpose. This step only confirms the authenticity of the intermediate node that claims to have a route to the destination, but does not verify the node behavior. The intermediate node could be a cooperative attacker that tries to launch a black hole attack and refers to another node as a terminal or destination. Thus, if the originator node does not authenticate the destination and verify the route, it will send packets to the attacker. However, to overcome this issue, Node v_i needs to send a secure $Hello$ packet to Node v_d through the intermediate node (v_n or v_B) to verify the route existence. If Node v_n receives the $Hello$ packet, it will act legitimately and forward the packet to Node v_d. Next, Node v_d will send a secure $Hello$ packet back to Node v_i through Node v_n. Node v_i then will verify the received $Hello$ packet to ensure the authenticity of Node v_d and the route information. However, if an attacker exists and receives the $Hello$ packet, the attacker cannot forward the packet to Node v_d because it does not have a route to it. Therefore, Node v_i needs to confirm the misbehaving by sending another RREQ after a timeout period. Typically, the attacker intuition is to capture network traffic and send a RREP containing the same or higher SN.

When Node v_i receives the RREP, it will send a secure $Hello$ packet again. If there is no response from the destination, then Node v_i can consider Node v_B to be suspicious. As a result, Node v_i creates a d^{req} that includes selective information from the suspicious RREP, such as the sender's identification v_B, and its CH's identification $v_B^{c_y}$. This information aggregated in a packet d^{req}=$\langle v_i, v_i^{c_y}, v_B, v_B^{c_y} \rangle$ and sent to its CH for detection. Note Node v_B may act legitimately by ignoring sending a RREP when it receives the second RREQ to avoid being trapped. In this case, BlackDP can only prevent the black hole establishment and cannot detect the attacker.

Suspicious Node Examination: When CH receives the d^{req}, it records d^{req} in a $verification\ table$ that stores several fields including *ids* for the originators, and suspicious nodes, as well as the *ids* for all affected CHs. This information can assist CHs to identify cluster membership. Furthermore, it helps in reducing the number of redundant detection requests for the same suspicious node. This may happen when the highway is congested and many nodes wish to verify the same suspect node against the existence of the black hole attack. The complete entry information of the table will be as follows: t_i=$\langle v_i, v_i^{c_y}, v_B, v_B^{c_y} \rangle$. Note that c_y depends on the location of the nodes in the clusters. The location of the suspicious node is verified and recorded within the *verification table*. If the suspicious node (Node $v_B^{c_y}$) is in the same cluster as the originator (Node v_i), then verification begins

immediately. Otherwise, the CH forwards d^{req} to the CH where the suspicious node resides. If d^{req} is forwarded, the receiving CH searches for Node $v_B^{c_y}$ in its routing table, rather than immediately storing the node in its *verification table*, to reduce storage overhead.

Detection is performed once the CH finds the suspicious node in its routing table after generating a disposable identity that is used to fool the attacker. This will make attacker feel safe during launching attacks and think the CH is a normal node. Thereafter, CH creates a $RREQ_1$ that includes a fake destination identity and sends it to the suspect node, which will reply immediately to ensure that it is the fastest node to respond. The CH receives the RREP from the suspect node and confirms the attack, where the confirmation involves violation of the AODV protocol. Specifically, a node must not send a RREP if it does not have a higher SN than the received RREP in the RREQ [21]. Thus, the CH creates a new $RREQ_2$ packet for the same fake destination with a higher destination sequence number than in $RREP_1$, with an additional inquiry about the next hop. $RREQ_2$ is then sent to the suspect node, which responds with a new $RREP_2$ that contains a higher sequence number than that in the $RREQ_1$ and additionally may add information about a teammate (i.e., in the case of a cooperative attack) to the packet. Next, the CH performs the same detection process on the cooperative attacker and may take action, depending on the response from the attacker. When the attackers are detected by the CH, a response packet will be sent to the originator node through its respective CH reporting the result of the detection. Meanwhile, the current CH that performs the detection applies the *black hole node isolation* and notifies adjacent clusters.

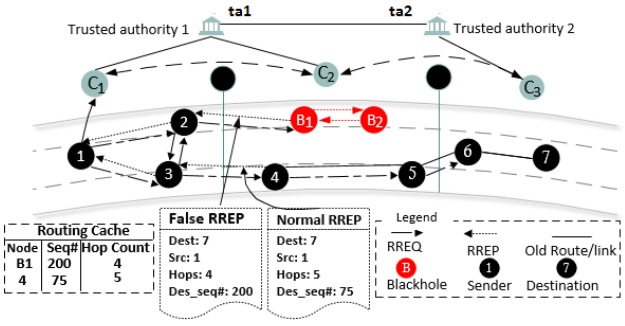

Fig. 3: Cooperative black hole attack detection process.

2) Black Hole Node Isolation Phase: When the attacker is detected, the CH sends a certificate revocation request to the trusted authority reporting misbehaving activities. The trusted authority processes the revocation request and informs other trusted authority nodes to pause attacker renewal certificates and sends a revocation notice to the surrounding CHs. The notice include the latest id (temporary pseudonyms identification), serial number, and expiration time of the attackers certificate. This revocation notice needs to be stored until the revoked certificate would have expired normally. The attacker is officially not certified, but preserves a valid certificate that can be used to falsify routes. Therefore, the attacker may still use its certificate to mislead other nodes and purport them as if it is certified. To overcome this issue, the CH needs to report the existing and newly-joined vehicles about the recent revoked certificate information to be aware of the attackers. In addition, every CH needs to store the revoked certificate information and then remove them once they expired to avoid reporting expired information and reduce storage overhead. We next illustrate our technique with a motivating example.

3) Illustrative Example: Consider the scenario in Figure 3 that depicts a highway divided into three clusters, each represented by a cluster head (C_1, C_2, C_3) and each cluster contains members (vehicles). For example, $\{v_1, v_2, v_3\} \in C_1$ and $\{v_4, v_{B_1}, v_{B_2}, v_5\} \in C_2$. Assume there are two trusted authorities nodes TA =$\{ta_1, ta_2\}$, and every $ta_z \in TA$ is responsible for a set of CHs such as $\{C_1, C_2\} \in ta_1$ and $\{C_3\} \in ta_2$. Let Node v_1 wants to communicate with Node v_7 which is not in transmission range. Therefore, it needs to broadcast a RREQ to its immediate neighbors, requesting a route to Node v_7.

Node v_2 and Node v_3 receive the RREQ and they check their routing table for a route to Node v_7 and they will rebroadcast the RREQ to all their immediate neighbors upon finding none. Once the RREQ is received by Node v_{B_1}, it will immediately send a secure or insecure RREP with a very high SN = 200 (c.f., Figure 3) through the same path that the RREQ came from as follow: $v_{B_1} \rightarrow \{v_1\} : RREP = \{RREP, CR, d_{sign}(RREP, K_B^-)\}$. Note the SN value depends on the attacker behavior, and it could be over or less 200, but we chose 200 to exemplify the reaction of the attacker based on the black hole definition [3]. The RREP contains the original RREP information, the certificate of the node, and the secure code that generates from the one-way hash function d_{sign}. Meanwhile, assume Node v_4 had already communicated with Node v_7 before the RREQ was sent from Node v_1, so Node v_4 will send a RREP that has a normal SN = 75 (normal SN that received from the destination) to Node v_1 as follows: $v_4 \rightarrow \{v_1\} : RREP = \{RREP, CR, d_{sign}(RREP, K_4^-)\}$. Hence, Node v_1 will store both RREP packets in its routing cache.

When Node v_1 checks its routing cache table, it will find that the RREP belonging to Node v_{B_1} is the freshest route. Therefore, Node v_1 sends a $Hello$ packet to authenticate and verify the destination as follow: $v_1 \rightarrow \{v_7\} : Hello = \{Hello, CR, d_{sign}(RREP, K_1^-)\}$. When Node v_{B_1} receives the $Hello$ packet, it cannot reply because it is not the destination. Consequently, Node v_1 waits for a time out and if no reply comes, it will redo the route discovery with the authentication process. If the attacker sends a RREP, Node v_1 directly sends a detection request d^{req} =$\langle v_1, v_1^{c_1}, v_{B_1}, v_{B_1}^{c_2} \rangle$ to Node c_1 to perform the detection. Another possible scenario, Node v_{B_1} may reply with a fake $Hello$ packet claiming that itself or the teammate attacker is the destination. In this scenario, Node v_1 sends the detection request without performing the second route discovery because of the anonymity response. When Node c_1 receives the d^{req} request, it records it in its $verfication\ table$ and checks the position field of the d^{req}. Node v_{B_1} is found to not be in the same cluster as Node v_1, so the d^{req} is forwarded to c_2 to complete the detection. Node c_2

receives the d^{req} and checks its routing table to find that Node v_{B_1} is registered in its routing table. Node c_2 then records the d^{req} in its $verification\ table$ and performs the detection.

First, Node c_2 will create a disposable identity = 50 and a fake $RREQ_1$ = ⟨Dest:10, Src:50, Hop:0, Dest_seq#: 0⟩, whose destination does not exist. Then, RREQ is sent to v_{B1}, which, by black hole attack definition, will reply as fast as possible to this request to guarantee that $RREP_1$ = ⟨Dest:10, Src:50, Hop:4, Dest_seq#:250⟩ reaches the requestor before others. Upon receiving the $RREP_1$, Node c_2 creates another $RREQ_2$ = ⟨Dest:10, Src:50, Hop:0, Dest_seq#:251, Next_Hop:⟩ and sends it again to v_{B_1}, thereby receiving $RREP_2$ ⟨Dest:10, Src:50, Hop:4, Dest_seq#:300, Next_Hop:v_{B_2}⟩ that contains a higher sequence number than the previous $RREP_2$. Node c_2 then reports misbehaving activities to Node c_1 and Node ta_1. Node c_1 will notify its members to avoid any route through Node B_1, while Node ta_1 processes the revocation certificate and officially reports that to Node ta_2 to pause renewing the attacker certificate and inform CHs about the revoked certificate to include it in their black-list. CHs needs to inform newly joined vehicles about the revoked certificates to avoid receiving multiple detection requests about anonymity responses, which may cause a denial of service attack. Since Node v_{B_1} claims to have a route to the destination through Node v_{B_2}, Node c_2 needs to verify that by sending a $RREQ$ includes this claim to v_{B_2}. If Node v_{B_2} supports the claim of Node v_{B_1}, then Node v_{B_2} is considered as a cooperative attacker, and Node c_2 isolates it from the network.

C. Overhead Analysis

In this section, we present the RSUs overhead due to computation and storage incurred by our protocol.

1) Computation Overhead: The computation overhead through RSUs during the single black hole attack detection has two cases. In the best case, when the attacker exists in the same cluster of the node that requested the detection, the number of packets sent to perform the detection are at least six: one from the legitimate node to the CH, four for detection packets (i.e., $RREQ, RREP$), and one for informing the node that requested the detection. However, in the worst case, there are two scenarios. First, if the attacker left to another cluster after sending the first RREP of the detection, the CH is required to send a packet to the next CH for completing detection and reporting the detection results, leading to two extra packets. Second, if the attacker is not in the same cluster as the legitimate node but still exists in its same cluster during the detection process, the number of packets required to perform the detection are eight and increases by one if the attacker left the cluster. Similarly, the cooperative black hole attack has the same number of detection packets for the first attacker, plus an additional two packets for detecting each cooperative node. From the previous analysis, only received packets from mobile nodes need to be authenticated and verified by the RUSs, while packets exchange between RSUs do not as they use their internal communication. Therefore, the overhead computation associate with the number of detection request from mobile nodes.

2) Storage Overhead: The storage overhead depends on two factors: nodes that requested the detection and nodes being verified. Most of the storage overhead of BlackDP takes a place in the $verification\ table$. The $verification\ table$ length relies on the number of suspect nodes, while the node that requested the detection is limited to those within cluster range. Therefore, the hashed list could be, in the worst case scenario, equal to the number of nodes in the cluster.

D. Limitation

Using RSUs as trusted nodes to perform the detection and isolation in BlackDP may incur computation overhead that possibly poses performance issues. BlackDP requires RSUs to authenticate nodes that report suspicious activities to ensure packets come from secure nodes. The authentication processing time may create a bottleneck when the density of the cluster is very high and numerous vehicles have to communicate with the RSU in range to validate a route. However, RSUs can leverage fog computing to overcome such issues by expanding the computation resources and forward heavy computation to nearby fog nodes [5].

IV. Experimental Results

We next describe our experiment set up and present experimental results from applying the BlackDP protocol to a simulated CV network.

A. Experiment Setup

In this section, we describe our simulation to demonstrate the effectiveness of BlackDP in detecting black hole attacks. For the purpose of our testing, we chose to implement our experiment on a controlled-access highway. This highway is divided into many clusters based on the signal range of the RSU. Each cluster is supervised by an RSU that monitors entering and exiting vehicles and oversees the flow of traffic. RSUs are arranged in the middle of every cluster and the vehicles are randomly distributed within the clusters. Every node in the network uses the AODV protocol for routing establishment and the traditional Elliptic Curve Digital Signature Algorithm for authentication as defined in IEEE 1690.2 [2].

TABLE I: Simulation parameters

Parameter	Value
Vehicle speed	50-90km
#Vehicles	100
#RSUs (CHs)	10
Transmission range	1000m
Highway length	10km
Highway width	200m
Cluster length	1000m

Table I shows our simulation parameters. The number of vehicles is selected to be 100 to ensure the disconnectivity between some nodes (i.e., attacker and destination) and allows a malicious node to purport neighbors of having a route to a destination node. The highway is selected to be a length for $10km$ and a width of $200m$ and be long enough with 10

segments to give attackers flexibility to flee from their clusters. Every vehicle has its own unique coordinates, communication range of 1000m [12], direction, and speed between 50-90km. A source car is placed at the beginning of the highway, while the attacker is placed in the same cluster or in the neighboring clusters, but not in the same as the destination to ensure that the attacker does not have a route to the destination. Note the destination may not exist in the clusters depending of the experiment scenario. We specified a set of clusters (e.g., cluster 8-10) to randomly renew attackers' certificates and allow them to perform malicious attacks along with normal behavior to measure the performance of the detection.

Each simulation runs with a different combination of sender-receiver nodes and black hole attacker locations. We repeated the simulation 150 times with different experimental treatments of either a single or cooperative black hole. The placement of the attacker(s) depends on the type of the black hole. If the black hole is single, then the attacker can be placed anywhere in the network, but not in the communication range of the destination. In the cooperative attack, both attackers need to be within communication range of each other to cooperate and falsely deceive the legitimate nodes.

B. Experimental Result

In this section, we discuss the result of applying BlackDP to our simulation for detecting the existence of a black hole attack.

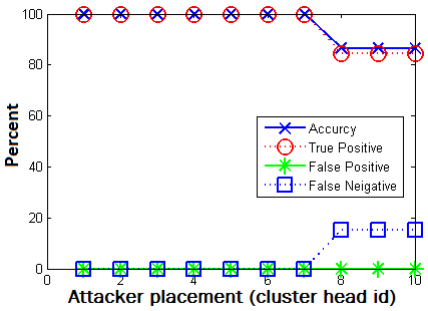

Fig. 4: Single and cooperative black hole attacks

Figure 4 shows that BlackDP can detect the existence of the black hole attack with a high detection accuracy rate of 100% and 0% false positives and false negatives when the attacker exists between cluster 1 and cluster 7, because the attacker(s) confirmed the attack and maliciously responded to all detection packets. However, when the attacker(s) is placed between cluster 8-10, the performance of the detection accuracy and true positive rates drop due to several reasons: the attacker acted legitimately during the detection phase (e.g., authentication phase or an attacker examination through an RSU), the attacker fled from the network, specifically cluster 10, without responding to the RSU detection packets, or due to certificate renewal where the attacker takes advantage of changing its identity during the detection process. These reasons could prevent BlackDP from detecting and isolating the attacker, but they could not prevent BlackDP from impeding black hole attacker from launching their attack. Figure 4 also shows the false negative rates increased between cluster 7 and 10 due to the failure of detecting and isolating the attackers, while they may still exist in the network.

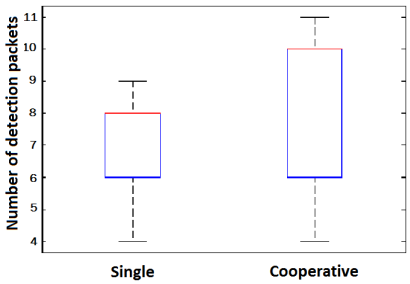

Fig. 5: Number detection packets

Figure 5 demonstrates number of detection packets needed by BlackDP through RSU (CH) to detect the single and cooperative black hole attacks. When there is no black hole attacker in the network, the total number of detection packets varies between four to six for both attackers. If a black hole attacker exists in the network, the total number of detection packets required is six to nine packets. Six packets are needed if the attacker is in the same cluster of the legitimate node. Eight packets are needed when the attacker exists in the same cluster of the originator node and responds to the first detection packets, but left to the next cluster (causing the CH forwarding the process to the next CH to complete the detection). Nine packets are needed the previous scenario, however the attacker is not in the same cluster as the originator node. For the cooperative black hole attack, there are eight to eleven packets sent to detect the attacker. This is similar to the single black hole detection approach, but with an additional two packets for detecting the cooperative attacker.

V. Related Work

In this section, we highlight related work on techniques for detecting black hole attacks using sequence number based detection [13][15][26], cryptography [18][23], and opinion and trust based approaches [6][8][24].

A. Sequence Number Based Methods

Jaiswal *et al.* [13] suggested collecting all RREPs at the source node and then comparing the first RREP SN with the rest. If the first RREP's SN is high compared to the rest, then whoever's node sent this RREP is surely an attacker and its entry is removed from the route request table. Jhaveri *et al.* assigned an intermediate node to calculate the maximum possible value of a SN that any RREP can have in the current state PEAK (the maximum possible value of the sequence number) after every time interval [15]. If the SN is higher than the PEAK, its node is marked as malicious in the routing table. Tan *et al.* defined three thresholds for different environments (small, medium, and large) based on the minimum and maximum SN [26]. If the SN is greater than the threshold, then the node is considered a malicious one and its packet is discarded. Previous approaches relies on SN, and they assumed that there are always multiple RREPs for a specific RREQ. This assumption is not valid in CV highway

networks. There might be a situation where the attacker is the connector of two networks in a highway and responds with a RREP. In this case, non of the previous techniques can detect the attack. In addition previous approaches do not detect both types of black hole attack. Yet, BlackDP detects and isolates both attackers after examining their behavior directly through trusted nodes (RSUs).

B. Cryptography-based Methods

Several works rely on cryptography for preventing black hole attacks. Gajera *et al.* introduced IKM, a combination of ID-based key management and threshold cryptography, to authenticate all routing messages [18]. Malicious nodes are unaware of the network security configuration and they can't enter and interrupt the routing process. Sachan *et al.* schema introduced the based message authentication (HMAC) to authenticate the non-mutable fields of the routing message, such as the SN and IP address, to provide fast message verification [23]. However, authenticating both route discovery and maintenance add more complexity and overhead to the network as every node needs to secure every packet sent and routed. Techniques that assume source and destination share a symmetric key before the route establishment indicates the network is small and centrally managed as if a new node joins the network, it needs to know the secret key before sending a RREQ. This assumption is not valid in CV network as nodes arbitrary join/leave the network. However, BlackDP requires only the destination, source and intermediate node (claims to have a route to the destination) authenticate their identity after the route discovery phase. In addition, BlackDP does not rely on a symmetric key for route establishment due to the frequent node identity changes, which require nodes to change their information to hide their identities.

C. Opinion- and Trust-based Methods

Dangore *et al.* used opinion-based methods and judge nodes based on the number of delivered packets. The more packets a node delivers, the higher recommendation they will receive [8]. Sen suggested that every node maintains a routing information table to store neighbors routing activates that can be used to check which nodes do not forward packets and are marked as an attacker [24]. Kaur *et al.* proposed a trust-based detection algorithm for elimination of black hole nodes that gives nodes trust values based on their activities, then it calculates all trust values for all paths and chooses the path with the highest trust value [14]. Yet, opinion and trust-based methods may not work properly in CV networks due to high speeds and frequent network leaving/joining, resulting in inaccurate trust and opinion rates. However, BlackDP is constructed to be semi-centric and does not require additional information to examine attackers.

VI. Conclusion and Future Works

In this paper, we presented BlackDP, a semi-centric protocol that decouples the detection mechanism from mobile nodes and assigns it to stationary devices (i.e., RSUs) for detecting both single and cooperative black hole attacks after a report from a legitimate node about abnormal activities. BlackDP is a protocol that comprises two phases. The first phase is black hole node identification, where a CH (i.e., RSU) receives a request from a legitimate node reporting a suspicious route activities from a node that claims to have the freshest route to a destination node with a very high SN. The CH then performs verification based on the suspicious node's location to prove the existence of a single or cooperative black hole attack. Next, the attacker(s) are isolated from the network. The CH will report the attacks to its members (i.e., vehicles) and neighbors (i.e., CHs) to add attacker(s) to their blacklist and avoid communications with the attacker(s). We demonstrated the validity of our results by a simulation of a CV network with varying types of black hole attacks.

Some open issues remain to be explored. The storage and computation overhead at each cluster head needs to be reduced. Moreover, the proposed detection protocol does not yet account for an urban topology network. Lastly, we will investigate our approach in advanced simulation frameworks, such as NS3 and SUMO.

Acknowledgment

We gratefully acknowledge the assistance from Jasser Al Jasser in discussing the general solution of this paper. 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 and Oakland University or other research sponsors.

References

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

[2] Ieee standard for wireless access in vehicular environments–security services for applications and management messages. *IEEE Std 1609.2-2016 (Revision of IEEE Std 1609.2-2013)*, pages 1–240, March 2016.

[3] Mohammad Al-Shurman, Seong-Moo Yoo, and Seungjin Park. Black hole attack in mobile ad hoc networks. In *Proceedings of the 42nd annual Southeast regional conference*, pages 96–97. ACM, 2004.

[4] Raghad Baiad, Hadi Otrok, Sami Muhaidat, and Jamal Bentahar. Co-operative cross layer detection for blackhole attack in vanet-olsr. In *International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 863–868. IEEE, 2014.

[5] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[6] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the confidant protocol. In *3rd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, pages 226–236. ACM, 2002.

[7] M. E. M. Campista, P. M. Esposito, I. M. Moraes, L. H. M. k. Costa, O. C. M. b. Duarte, D. G. Passos, C. V. N. De Albuquerque, D. C. M. Saade, and M. G. Rubinstein. Routing metrics and protocols for wireless mesh networks. *IEEE Network*, 22(1):6–12, Jan 2008.

[8] Monika Y Dangore and Santosh S Sambare. Detecting and overcoming blackhole attack in aodv protocol. In *Cloud & Ubiquitous Computing & Emerging Technologies*, pages 77–82. IEEE, 2013.

[9] Jerome Haerri, Fethi Filali, and Christian Bonnet. Performance comparison of aodv and olsr in vanets urban environments under realistic mobility patterns. In *Proceedings of 5th IFIP Mediterranean Ad-Hoc Networking Workshop*, volume 1045, 2006.

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

[11] Yi-an Huang and Wenke Lee. A cooperative intrusion detection system for ad hoc networks. In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 135–147. ACM, 2003.

[12] A. S. T. M. Intl. Standard specification for telecommunications and information exchange between roadside and vehicle systems-5 ghz band dedicated short range communications. *ASTM International*, 2003.

[13] Pooja Jaiswal and Rakesh Kumar. Prevention of black hole attack in manet. *IRACST–International Journal of Computer Networks and Wireless Communications*, pages 2250–3501, 2012.

[14] Tanupreet Singh Jasmeen Kaur. Trust based discovery and disposal of blackhole attack in mobile ad hoc networks. *HCTL Open International Journal of Technology Innovations and Research*, 16, 2015.

[15] Rutvij H Jhaveri, Sankita J Patel, and Devesh C Jinwala. A novel approach for grayhole and blackhole attacks in mobile ad hoc networks. In *Second International Conference on Advanced Computing & Communication Technologies*, pages 556–560. IEEE, 2012.

[16] Zdravko Karakehayov. Using reward to detect team black-hole attacks in wireless sensor networks. In *Workshop on Real-World Wireless Sensor Networks*, pages 20–21, 2005.

[17] Songbai Lu and Longxuan Li. Saodv: a manet routing protocol that can withstand black hole attack. In *Computational Intelligence and Security.*, volume 2, pages 421–425. IEEE, 2009.

[18] Sowmya K. S Mayuri Gajera. Prevention of black hole attack in secure routing protocol. *International Journal of Science and Research*, 2013.

[19] Hassnaa Moustafa and Yan Zhang. *Vehicular networks: techniques, standards, and applications*. Auerbach publications, 2009.

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

[21] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*. IEEE, 1999.

[22] Sanjay Ramaswamy, Huirong Fu, Manohar Sreekantaradhya, John Dixon, and Kendall E Nygard. Prevention of cooperative black hole attack in wireless ad hoc networks. In *International conference on wireless networks*, volume 2003, pages 570–575, 2003.

[23] Preeti Sachan and Khilar. Securing aodv routing protocol in manet based on cryptographic authentication mechanism. *International Journal of Network Security & Its Applications*, volume 3, 2011.

[24] Jaydip Sen. Detection of cooperative black hole attack in wireless ad hoc networks. *International Journal of Simulation,Systems,Science and Technology*, volume 12, 2013.

[25] Ming-Yang Su. Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems. *Computer Communications*, volume 34:107–117, 2011.

[26] Seryvuth Tan and Keecheon Kim. Secure route discovery for preventing black hole attacks on aodv-based manets. In *High Performance Computing and Communications*, pages 1159–1164. IEEE, 2013.