SWANS: SDN-based Wormhole Analysis using the Neighbor Similarity for a Mobile ad hoc network (MANET)

Faheed A.F. Alenezi
School of Computing and Engineering
University of Missouri-Kansas City
Kansas City, USA
faacnb@mail.umkc.edu

Sejun Song
School of Computing and Engineering
University of Missouri-Kansas City
Kansas City, USA
sjsong@umkc.edu

Baek-Young Choi
School of Computing and Engineering
University of Missouri-Kansas City
Kansas City, USA
choiby@umkc.edu

***Abstract*—In managing a large-scale mobile ad hoc network (MANET), real-time root cause analysis of system events is difficult, as it requires vast measurement and processing resources. Software-defined management and control for a MANET promise to enhance their performance, scalability, and flexibility. However, a wormhole attack is one of the most challenging yet detrimental security issues in a software-defined MANET. As most of the existing security countermeasures are designed mainly for wired SDNs using network topologies, a software-defined MANET cannot directly use them.**

This paper proposes an SDN-based Wormhole Analysis using the Neighbor Similarity (SWANS) approach as a novel wormhole countermeasure in a Software-defined MANET. As SWANS analyses the similarity of neighbor counts at a centralized SDN controller, it apprehends wormholes not only without requiring any particular location information but also without causing significant communication and coordination overhead. SWANS also countermeasures various false-positive and false-negative scenarios generated by the Link Layer Discovery Protocol (LLDP) vulnerability. We performed extensive studies via both analysis and simulations. Our simulation results show that SWANS can efficiently detect various intelligent wormhole attacks, keeping low false-positive and false-negative rates.

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

As a MANET uses an open-architecture based broadcast medium to support mission-critical applications in challenging environments, it is susceptible to various security attacks. First, network security management in a mobile ad hoc network (MANET) is limited due to scalability. As the number of devices increases, the network membership control becomes highly dynamic, radio access resources are managed across multiple aggregated carriers, and network problem detection, isolation, and root cause analysis become increasingly expensive. Second, due to increased complexity, Software-defined Networking (SDN) increases the attack surface, including forwarding switches, controllers (logically a single logical point of failure), the connection between the switch and the controller, and various applications. SDN needs to verify that the flow commands come from authorized sources. However, as the Link Layer Discovery Protocol (LLDP) vulnerability can amplify attacks on the controller, SDN creates additional security challenges, including unauthorized access to data and control planes of networks (i.e., wormhole attacks) and data leakage by timing analysis.

978-3-903176-32-4

A wormhole attack [6] is a particularly challenging security problem because it can silently deploy the attack without compromising other security means as long as the communication channel is known. Wormhole attackers attract data packets into one point and relay them to distant locations by tunneling through the attacker's implicit direct communication links. The attackers can also perform various malicious data and control traffic manipulations by selectively dropping, flooding, recording, or modifying packets without revealing their identity. However, most of the existing countermeasures against wormhole attacks are for static SDNs using network topology information using sophisticated devices such as directional antennas and GPS. However, harnessing those devices on the resource-limited MANET nodes is not practical. It is critical yet challenging to design wormhole attack detection and protection methods in a software-defined MANET because they depend upon the network topology, distance, direction, and location among the pivotal neighbors [3], [5], [11].

This paper proposes a novel wormhole attack analysis method in a Software-defined MANET, namely an SDN-based Wormhole Analysis using the Neighbor Similarity (SWANS) approach. SWANS detects wormhole attacks using an online outlier detection algorithm at a centralized SDN controller identifying any neighbor counts abnormality (the lack of similarity). We also utilize node mobility itself to determine the anomaly caused by wormholes. For example, when a node moves inside a wormhole attack area, the node would experience the rapid change of its neighbors' characteristics due to the virtual tunnel created by wormhole attackers. As the neighbor discovery protocol (i.e., LLDP in SDN) is one of the essential functionalities in a software-defined MANET, SWANS does not require any additional communication overhead. SWANS also countermeasures various false-positive and false-negative attack scenarios generated by the intelligent

wormhole attacker in assessing LLDP vulnerability (attacker models defined in Figure 1). We performed extensive studies via both analysis and simulations. Our simulation results show that SWANS can detect various intelligent wormhole attacks efficiently with low false-positive and false-negative rates. SWANS is the first wormhole countermeasure in a software-defined MANET that does not require global topology information or special hardware to the best of our knowledge.

The remainder of this paper is organized as follows. Section II describes the related work. We present the wormhole attacker types, models, and countermeasure designs in Section III. We discuss the proposed SWANS algorithm and the implementation details in Section IV. Section V provides the experimental setup, assumptions, and detecting results for each attacker model, and Section VI concludes the paper.

II. Related Work

The existing wormhole detection methods mostly use GPS, directional antennas, and timing devices. [4] proposed geographic and temporal leashes to detect the wormhole attack. The geographic leash is used to ensure that the packet's receiver is within the sender's range, which requires each node to know its location with GPS. [7] uses fuzzy logic according to residue energy, the distance between nodes, and hop count. [8] introduced a trust-based approach to circumvent wormhole attackers. It detects the trusted path by using the round trip time (RTT) threshold, packet drop ratio(PDR) threshold, and energy consumption threshold rate. [9] detects changes of neighbors' status for a node and the length of paths. Any node that has an unexpected change is considered malicious. [10] uses a coordinator node to mitigate wormhole attacks (an election -based central approach). The coordinator manages all nodes by using a special coordinator message. [13] proposed a statistical approach using neighbors to detect the wormhole attack in mobile WSNs. They identify the wormhole attack by counting the recent number of neighbors of each node and compare it with the previous count.

SWANS differs from the existing work, as it countermeasures wormholes without requiring any special devices for timing and location coordination. It also does not cause any significant communication and coordination overhead.

III. Wormhole Attacker Models and Analysis Methods

A. Wormhole Attacker Types and Models

As described in Table I, a wormhole attacker, W_x, consists of more than one end node, which can be three different types, including Full_Stealthy, Partial_Stealthy, and No_Stealthy types. W_x attempts to attract more network traffic by compromising neighbors. Unlike the distributed algorithm, a centralized SDN controller can efficiently detect W_x by checking the number of neighbor nodes' abnormality, $nNum$, from the neighbor table. However, W_x can manipulate their neighbor counts intelligently using the LLDP vulnerabilities in SDN. Hence, we identify and tackle the following intelligent attacker models:

TABLE I
Notations

Notation	Explanation
S_x	Wireless & mobile node x
H_x	Wireless & mobile host x
W_x	Wormhole attacker x, which consists of more than one end nodes (W_1 & W_2). W_x can be 3 different modes (Full_Stealthy, Partial_Stealthy, and No_Stealthy)
X_{com}	Node X's communication range area with radius r
F	Entire network field area
N	Total number of the nodes in the field
$nNum_a$	The number of neighbors of node a
NSI	The Neighbor Similarity Index (NSI) is the distance of two K means clusters
ACI	The Augmented Concentration Index (ACI) is the value of the similar cluster nodes in the surrounding neighbors
T_sh	Threshold class for both NSI and ACI

- W_x can perceive and manipulate (eavesdrop, spoof, alter, etc.) LLDP control messages within its range, X_{com}. W_x can observe the packet transmission time and frequency data to perform traffic analysis and infer target objects' locations.
- W_x can adjust $nNum_a$ by manipulating its reply to LLDP requests to hide its existence. It can shrink its communication range smaller than the existing node's range (RR in Figure 1). It also can send LLDP request to only one of the two endpoints instead of both (RO in Figure 1).
- W_x can randomly spoof LLDP Packet-In messages on behalf of other nodes (RS in Figure 1) to hide its location and victimize other nodes. W_x can inject fake neighbors by randomly choosing both source and destination node IDs and sending LLDP response.
- W_x cannot spoof neighbor nodes for the out of range nodes (random locations only).

B. Wormhole Attacker Scenarios

An SDN controller uses a centralized discovery protocol (LLDP) to identify neighbor nodes for each node. W_x can replay all the LLDP request messages within the range to its neighbors via a wormhole tunnel. As illustrated in Fig. 2, while node S_1 is in the communication range of a wormhole node W_1, S_1 receives all LLDP requests from the nodes within S_1's communication range X_{com} as well as from the nodes around W_1 and W_2 as the wormhole node W_1 relays the incoming LLDP request messages at W_2 through the virtual tunnel. Hence, when there is W_x, the SDN controller will receive more Packet-In notifications (neighbors) from the wormhole affected nodes for an LLDP request. When there is a packet forwarding message, the SDN controller's routing protocol identifies a route (forwarding neighbor nodes on the path) from source to destination using the shortest hop-count algorithm according to neighbor status in the neighbor table. For example, when H_1 sends data packets to H_5, it takes the shortest route from S_1, S_2, S_n, S_4, and S_5 before any wormhole attack. However, after a wormhole attack, as routes via W_x become the shortest route (e.g., from S_1 and S_5 via

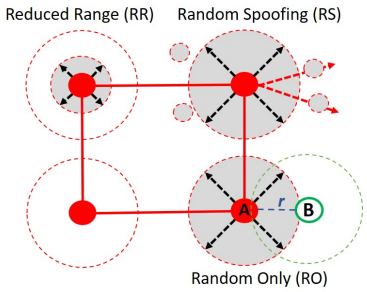


Fig. 1. Wormhole attacker models

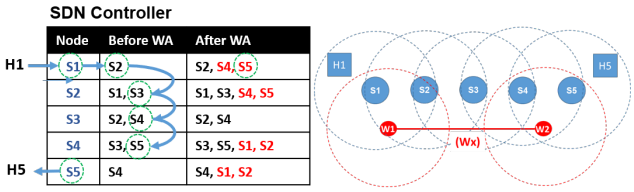


Fig. 2. Impact of wormhole attacker on neighbors and routes

W_x) for many forwarding cases, a wormhole attacker W_x can receive and manipulate both control and data packets.

C. Wormhole Attacker Analysis Methods

When there is W_x, $nNum$ of the nodes within the wormhole range increases beyond the range of statistical fluctuation. SWANS uses a K-means clustering algorithm to determine the Neighbor Similarity Index (NSI), which classifies the area of high neighbor count nodes from normal neighbor count nodes. However, NSI alone may not identify the wormhole attackers due to potential noises. Some nodes may temporarily have a high number of neighbors by chance. Also, a spoofing attack can intentionally increase the number of neighbors of a target node. Hence, in addition to NSI, SWANS proposes an Augmented Concentration Index (ACI) algorithm, which locates the wormhole attacker's core by checking the degree (or density) of surrounding nodes. As illustrated in Figure 3, for a high neighbor cluster (cluster 1), ACI augments the highly concentrated area by incentivizing the nodes surrounded by the same cluster nodes and penalizing the nodes surrounded by the different cluster nodes. Using k-means clustering with the ACI, SWANS can denoise the less concentrated areas.

As nodes move around in the network, the number of neighbors on a node changes over time. The analysis approach is to discover if any neighbor-counts exhibit abnormal increments than the nodes, out of the wormhole range. There are several approaches to outlier detection. SWANS conducts a general framework for estimating the underlying distribution of the neighborhood counts. The problem of finding outliers can be solved efficiently if the data distribution is calculated accurately. There are several model estimation techniques proposed in the literature including wavelets [1], histograms [2], and kernel density estimators [12]. For mobile nodes, we use a scheme to quantify the difference between the previous data

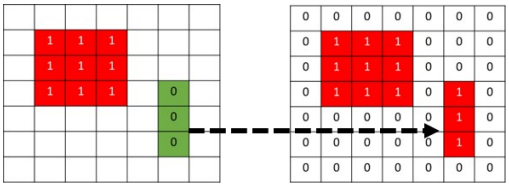


Fig. 3. ACI Analysis Algorithm

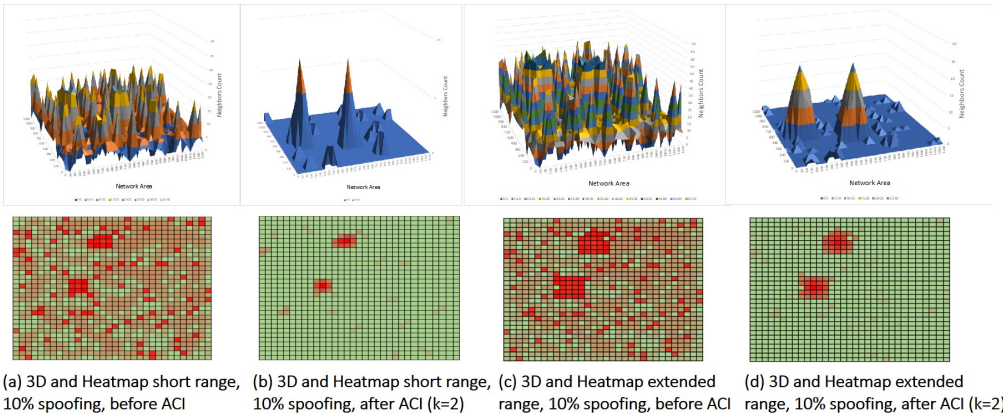
set of neighborhood counts (i.e., training set) and the new or recent data set (i.e., test set).

IV. SWANS: Algorithm and Implementation

The SWANS is a centralized approach to detect wormhole attackers in SDN. First, it calls a Check_Wormhole () using the neighbor table (n-table) and T_sh parameters. T_sh is a class with four different threshold values, including neighbor-distance, neighbor-balance, annotated-density-distance, and annotated-density-balance. The annotated-density-balance is a threshold of the NSI. A node S_x would have $(X_{com}/F) * N$ neighbors on average. We assume the node's transmission ranges of both W_x are the same as X_{com}, and the wormhole tunnel replays the beacon messages. The number of neighbors increases approximately from two times (when a node's X_{com} almost overlaps with one of the wormhole nodes' X_{com}) to three times (when a node gets into the range of a wormhole attacker). When there is no W_x, the NSI is bounded by the average number of neighbors. However, it increases approximately three times when there is a wormhole attacker. Hence the difference of the wormhole affected and not affected clusters (NSI) roughly increases by two times of $(X_{com}/F)*N+\alpha$. Therefore, it indicates a potential wormhole attacker in the network. Besides, the neighbor-balance is used to verify the wormhole attacker. The number of W_x affected nodes is far less than other nodes in the network. Typically, neighbor-balance should be less than $(2 * X_{com}/F) * N + \alpha$. However, due to various intelligent spoofing by W_x, it may be less than the value. Check_Wormhole () runs the k-means clustering algorithm with k=2 to find an initial NSI. However, Check_Wormhole () runs k-means clustering with k=3, if it cannot satisfy the neighbor-balance. When the NSI's k-means clustering tests (k = 2 and 3) fail (e.g., RS in Figure 1), SWANS also checks ACI parameter. As illustrated in Figure 3, there are two clustered areas of cluster 1. We annotate cluster 1 with different weights $+cluster1 * 2and - cluster0$. As the same cluster nodes usually surround the wormhole attacker range, they have higher density values (ACI). When a k-means clustering with k=2 or 3 is applied, it separates the less concentrated cluster from the more concentrated cluster.

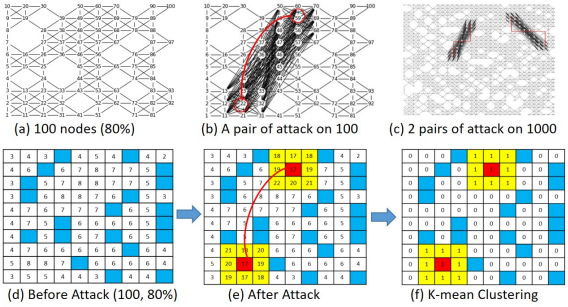
V. Evaluations

We have evaluated the SWANS with various wormhole attack scenarios, as illustrated in Figure 1. We used a discrete event simulation implemented by using Python. We apply both 100 and 1000 nodes on a grid network. We also set a couple of discrete radius as a communication range (short and extended



(a) 3D and Heatmap short range, 10% spoofing, before ACI (b) 3D and Heatmap short range, 10% spoofing, after ACI (k=2) (c) 3D and Heatmap extended range, 10% spoofing, before ACI (d) 3D and Heatmap extended range, 10% spoofing, after ACI (k=2)

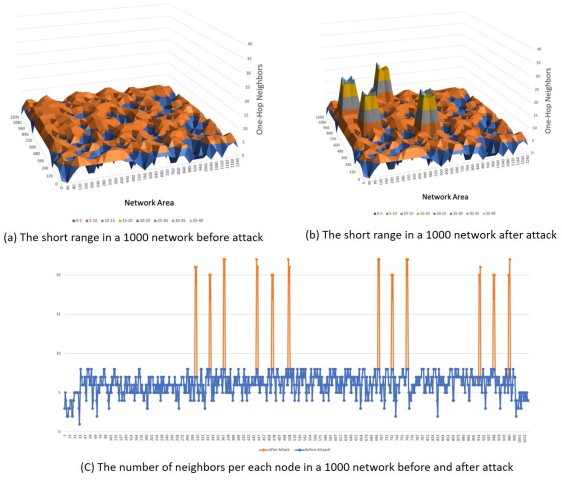
Fig. 4. Augmented Concentration Index (ACI) Results against 10% Spoofing



(a) 100 nodes (80%) (b) A pair of attack on 100 (c) 2 pairs of attack on 1000

(d) Before Attack (100, 80%) (e) After Attack (f) K-mean Clustering

Fig. 5. Neighbor similarity maps and distance



(a) The short range in a 1000 network before attack (b) The short range in a 1000 network after attack

(C) The number of neighbors per each node in a 1000 network before and after attack

Fig. 6. Short range on 1000 network with two pairs of attackers

range) and randomly allocate network nodes by varying the network density from 10% to 80%. For example, in Figure 5 (a), 80 initial nodes are randomly allocated in the 100 node network with 80% configuration. We randomly arranged a pair of wormhole attackers on the network. Figure 5 (f) presents k-means clustering results when k is 2. NSI after a wormhole attack becomes 13.63 as cluster 0's centroid is 5.26, and cluster 1's centroid is 18.89. Also, the number of nodes in cluster 1 is only 22.5% (unbalanced), and NSI is abnormally high (4 to 9 is the normal NSI range). The neighboring count-based clustering results reveal the clusters of nodes with an abnormally high number of neighbors after a wormhole attack. It also shows the potential area of the attacker.

Figure 6 presents the 1000 node network with a short communication range with a couple of wormhole attackers. Before any wormhole attack, the neighbor counts are similar over the entire system. However, after a couple of concurrent wormhole attacks, the NSI is very high, 14.18 (cluster 0's centroid is 6.01, and cluster 1's centroid is 20.19 using a k means clustering (k is 2)), and cluster1 has the only 4.5% of the nodes. The results show the potential four wormhole attack areas.

Figure 4 presents the results after 10% of RS wormhole attacks in both short and extended range networks. After the spoofing attack, the NSI alone cannot identify any wormhole attacker. For example, NSI becomes high, 12.35 (cluster 0 is 6.05, and cluster 1 is 18.4) in a short-range and 32.44 (cluster 0 is 16.6, and cluster 1 is 49.04) in an extended range. However, as the cluster nodes are not balanced (cluster 1 has 100 nodes in a short-range and 129 nodes in an extended range), we augment cluster 1 by incentivizing and penalizing each node (i.e., ACI) and apply k-means clustering. We detect the wormhole attacker areas as shown in Figure 4 (b) and (d).

VI. Conclusions

An SDN-based Wormhole Analysis using the Neighbor Similarity (SWANS) approach is presented as a novel wormhole countermeasure in a Software-defined MANET. As SWANS analyses the similarity of neighbor counts at a centralized SDN controller, it apprehends wormholes not only without requiring any particular location information but also without causing significant communication and coordination overhead. SWANS algorithm also countermeasures various intelligent wormhole attacker model scenarios applying NSI and ACI values. The extensive experimental results show that SWANS can detect wormhole attacks efficiently against very sophisticated attack scenarios.

REFERENCES

[1] Anna C. Gilbert, Yannis Kotidis, S. Muthukrishnan, and Martin Strauss. Surfing Wavelets on Streams: One-Pass Summaries for Approximate Aggregate Queries. In *VLSB*, 2001.

[2] Sudipto Guha and Nick Koudas. Approximating a Data Stream for Querying and Estimation: Algorithms and Performance Evaluation. In *ICDE*, pages 567–576, 2002.

[3] Lingxuan Hu and David Evans. Using directional antennas to prevent wormhole attacks. In *ACM Mobicom*, 2004.

[4] Y. . Hu, A. Perrig, and D. B. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, volume 3, pages 1976–1986 vol.3, 2003.

[5] Y.-C. Hu, A. Perrig, and D.B. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *IEEE INFOCOM*, 2003.

[6] Yih-Chun Hu, A. Perrig, and D.B. Johnson. Wormhole attacks in wireless networks. *IEEE Transactions on Selected Areas in Communications*, 24(2):370 – 380, 2006.

[7] Shahram Jamali and Reza Fotohi. Dawa: Defending against wormhole attack in manets by using fuzzy logic and artificial immune system. *the Journal of Supercomputing*, 73(12):5173–5196, 2017.

[8] Ashok Panwar, Bhavana Panwar, D Srinivasa Rao, and G Sriram. A trust based approach for avoidance of wormhole attack in manet. *International Journal of Computer Science and Mobile Computing*, 9:47–57, 2020.

[9] Manish Patel, Akshai Aggarwal, and Nirbhay K Chaubey. Detection of wormhole attacks in mobility-based wireless sensor networks. *International Journal of Communication Networks and Distributed Systems*, 21(2):147–156, 2018.

[10] R Arun Prakash, WR Salem Jeyaseelan, and T Jayasankar. Detection, prevention and mitigation of wormhole attack in wireless adhoc network by coordinator. *Appl. Math. Inf. Sci*, 12:233–237, 2018.

[11] Naveen Sastry, Umesh Shankar, and David Wagner. Secure verification of location claims. In *ACM Workshop on Wireless security*, WiSe '03, pages 1–10, New York, NY, USA, 2003.

[12] D. Scott. *Multivariate Density Estimation: Theory, Practice and Visualization*. Wiley & Sons, 1992.

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