Wireless Personal Communications (2020) 113:977–993
https://doi.org/10.1007/s11277-020-07263-9

The Design of a Defense Mechanism to Mitigate Sinkhole Attack in Software Defined Wireless Sensor Cognitive Radio Networks

Lanka Chris Sejaphala[1] · **Mthulisi Velempini**[1]

Published online: 13 April 2020

Abstract
The sink hole attack is a typical wireless sensor network attack. The sink hole node advertises itself as the best route to the sink node or the base station. The nodes in the communication zone of the sink hole node then redirects their observed data to the sink hole node upon receiving the broadcasted supposedly the best route. The sink node receives the packets and then drops them. It can also modify them before relaying them to the sink node or base station. This study proposes a scheme designed to address the effects of the sink hole attack called the Hop Count-Based Detection Scheme for Sinkhole Attack. The performance of the scheme is evaluated using Matlab. The scheme is compared to Ibrahim's algorithm which is the best algorithm in literature. The simulation results show that HCODESSA achieves good results in comparison to Ibrahim's. The results show that the sinkhole attack has adverse effect on the performance of the network and that its severity depends on the number of nodes it can mislead. We further evaluated the two schemes using statistical techniques which proved that HCODESSA performs better.

Keywords Internet of things · Routing · Sensor network · Sinkhole attack · Software-defined wireless sensor cognitive radio networks · Statistical techniques

1 Introduction

Software-defined wireless sensor cognitive radio network (SDWSCRN) is an emerging technology which is the integration of the software-defined networking (SDN) [1] and wireless sensor cognitive radio networks (WSCRN). It integrates the capabilities of SDN in the WSCRN such as programmability and automation. Software-defined wireless sensor cognitive radio network enhances the simplicity and flexibility of WSN management [2]. Software-defined wireless sensor cognitive radio network also incorporates internet

✉ Mthulisi Velempini
mvelempini@gmail.com

Lanka Chris Sejaphala
lcsejaphala@gmail.com

[1] Computer Science, University of Limpopo, Mankweng, South Africa

of things (IoT) which enables sensor actuator to be embedded onto physical objects to exchange observed data. It has the same structure as the SDN and it consists of the application layer, the control layer/plane which controls the network and the flow of data [3]. In an ever changing environment where it is difficult to manage the WSCRN after deployment and the reconfiguration of nodes is not possible hence the need for SDN capabilities [4]. The sensor nodes in the software-defined wireless sensor cognitive radio network reports local observed data to the sink node which is also known as the base station in an infrastructure based environment. Nodes can also send their reports to the web server in an ad hoc environment. In a case where the nodes require reconfigurations, network administrators use the software defined concept to reconfigure the nodes [5]. The concept of SDN in WSCRN simplifies network management [6]. However, WSCRN is prone to sinkhole attacks which is one of the major attacks in WSN in general. The compromised node advertises to its neighbours fake routing updates as the best route and shortest route to the base station, which attracts network traffic [7]. The neighbouring nodes will then direct their data to the sinkhole nodes which either drops packets or modifies packets before forwarding them. The attack can be used to launch other attacks such as selective forwarding, wormhole, Acknowledge spoofing attack. The sinkhole attack prevents the sink node or base station from receiving observed data or correct data.

Software-Defined Wireless Sensor Cognitive Radio Networks are susceptible to sinkhole attack. There is a need for the mitigation of the sinkhole attack in Software-Defined Wireless Sensor Cognitive Radio Network. The attack is capable of altering and dropping data packets which can lead to incorrect updates and decision making. In military, the effects of the sink hole attack are diverse in the context of intelligence, surveillance, and reconnaissance in combat in the battlefield and situational awareness [8]. Observation data is used to detect, identify, and classify threats based on count, number, and type whether it is an armed vehicle or men on foot, and the type and quantity of weapons [9]. Incorrect updates based on such observed data due to sinkhole attack can lead to causalities in the battlefield or at the camp. With the advancement of technology, SDWSN provides significant advantages in advancing humanity.

The introduction of IoT enables vehicle drivers to have pre-defined information about the conditions of the roads [10]. The sensor nodes transmit observed data to the gateway which will relay the data to the controlling computer or to the cloud application, where the drivers will be able to access the data in real-time and make timely decisions [11]. Such data should not be modified because it can course chaos on the roads in a typical smart transportation set up. This study propose to develop a Hop Count-Based Detection Scheme for Sinkhole Attack (HCODESSA) to mitigate the effects of the destructive sinkhole attack in the SDWSCRNs.

The scheme uses hop counts of the nodes as the bases of detecting the sinkhole nodes in the network. It isolated the identified sinkhole nodes from the network. The performance of the proposed scheme was evaluated in terms of probability of false positive, probability of false negative, and probability of detection in contrast to Ibrahim's Hop Count-Based Detection Algorithm. Moreover, the study evaluated the effects of the packet dropping sinkhole attack with regard to packets loss ratio and packet delivery ratio. On average, from the simulation results; HCODESSA outperformed Ibrahim's detection algorithm. Furthermore, statistical tests on the average probability of detection and probability of false positive of the two algorithms were conducted and the tests show that there is significance difference in terms of the two average probabilities and it was concluded that HCODESSA indeed outperformed Ibrahim's Hop Count-Based Detection Algorithm.

2 Literature Review

The literature review discusses some of the proposed algorithms for detecting and mitigating the sinkhole attack in the SDWSCRN and their challenges; furthermore, it highlights spectrum sensing concepts of the network and other security algorithms in wireless networks such as Vehicle Ad hoc Network (VANET) and Mobile Ad hoc Network (MANET). The two discussed networks are related to our study in a sense that they are wireless and mobile networks.

Choi et al. [12] proposed a method that uses link quality indicator to detect sinkhole attack in a network. Sinkhole attack advertises the highest link quality to its neighbors. The method requires extra nodes called detector nodes, which are used in detecting the attack. The challenge of the scheme is that detection of sinkhole only occurs when the detector node is between sinkhole node and source node, and sinkhole node and base station. Sheela et al. [13] proposed a non-cryptographic scheme which used mobile agents to defend against sinkhole attack. The mobile agents used dummy data to detect data modification, considering the data alteration capability of sinkhole attack. The algorithm is not flexible to allow scalability, which creates high network overheads because of the communication mode of the mobile agents and sensor nodes which is active mode. Sreelaja et al. [14], proposed a model for detecting the sinkhole attack which uses Ant Colony Optimization to identify an intruder in the WSN. The algorithm does not generate false positive according to the authors' results. The authors used Swarm intelligence techniques to detect the sinkhole attack [15]. Zhang et al. [16] proposed a redundancy mechanism to detect the sinkhole attack in a network. The authors took advantage of the three stages of path establishment, namely the Route Request, Route Reply and Route Establishment. In [17] authors proposed a light weight intrusion detection system equipped with an advanced intelligent system. For detection, they used an ensemble soft computing technique. Optimal relay selection technique is proposed in [18]. The algorithm implements the amplification and forwarding techniques to achieve efficient relay selection of the spectrum. Through dynamic management of relay selection and spectrum allocation, the algorithm is able to increase network throughput and maximizes the signal to noise ratio (SNR). Vehicle Ad Hoc Network (VANET) is one of the new emerging network technologies; which also requires security. Vijayakumar et al. [19] proposed a dual authentication and key management technique to allow secure communication between vehicles in VANET and to prevent unauthorized vehicle to enter into the VANET. The algorithm takes advantage of the Batch Level Signature (BLS) for key generation and smart card which the drivers are presented upon registering their vehicles in the VANET. Energy consumption is one of the attributes to consider in a wireless environment [20]. Muthurajkumar et al. [21] proposed a secure energy efficient routing algorithm which make use of intelligent agents for effective decision making. The algorithm does not only improve secure routing and efficient energy consumption. It also reduces routing delay in MANET.

In [22], authors proposed a message digest anomaly-based detection scheme to detect sinkhole nodes. The scheme uses an instance of cryptographic mechanism (the message digest approach) to assess the invulnerability of data, its integrity and authenticity. A novel detection algorithm for the detection of sinkhole attacks in WSNs is proposed [23]. In the approach, sink node checks the data communication route and keeps the current route of the nodes in memory. If it notices inconsistencies in a data packet, it checks the route of the nodes against the memory route and keeps similar nodes in memory and deletes the malicious node, then notifies other nodes not to forward their observed data to the detected

malicious node anymore. In the recent years (2018 and 2019), the research interest in routing related attacks has been increasing [24]. For example, Internet of things as an integral component of the 5th generation (5G) communications is vulnerable to routing attacks. It needs to be protected from routing attacks [25]. Sinkhole and selective forwarding are the most common and destructive attacks in infrastructure based networks. With the development of the SDN technology, it is essential for a network to be able to spread quickly new code to every node in the network and in a secured manner. Two-hop Neighborhood Information joint Double Broadcast Radius (TNI-DBR) algorithm was proposed to disseminate the security codes in a Software Defined Wireless sensor network in a fast, secure, and energy-efficient way in [26]. In [27] Liu et al., proposed a Probe Route Based Defense Sinkhole Attacks (PRDSA) which counters sinkhole attack and guarantees security for Internet Of Things. The algorithm consists of a routing mechanism which combines far-sink reverse routing, equal-hop routing, and minimum hop routing. The algorithm has minimum impact on the network lifetime.

3 Methodology

The study relies on computational simulation and mathematical formulae which leads to accurate and precise results, because this is a computational study. MatLab was used to code the algorithms, run simulations and to generate data which is then presented in a graphical format. MatLab is a multi-paradigm numerical computing environment and fourth-generation programming language [28].

3.1 Metrics

To test the effectiveness and robustness of the proposed scheme, some performance metrics were selected which, according to literature, are the best fit metrics for this kind of study. Moreover, to evaluate the effect of the sinkhole attack, some metrics were selected which are believed to be the suitable metrics.

The performance of the two algorithms was evaluated using the three performance metrics from literature, which are Probability of detection, Probability of false negative and Probability of false positive [29]. Furthermore, the effect of the attack was evaluated using packet delivery ratio (PDR) and packet loss ratio (PLR).

3.2 Network Setup and Simulations

Table 1 presents simulation parameters and their corresponding values, giving a summary of parameters which were considered in carrying out the experiment of the study. To enable the cognitive radio capability, the study used IEEE 802.11b standard with extension to support Cognitive radio networks.

The number of channels was set to 6, from literature 6 channels are suitable and efficient for a network of between 10 and 150 nodes [30]; hence, that is why 6 channels were chosen. Energy detection was used as the type of incumbent user detection type because it is less complex and it does not require the knowledge of primary user signal in detecting the presence and absence of the primary user [31].

Table 1 The simulation parameters of the study

Parameters	Values
Antenna	OmniAntenna
Environment	Cognitive radio wireless Environment
MAC protocol	IEEE 802.11b with extension to support cognitive radio networks
Number of data channels	6
Number of common control channels	1
Channel date rate	11 mbps
Number of sensor nodes	20, 40, 60, 80, 100
Percentage of sinkhole nodes in a network	15%, 25%, 35%
Propagation model	TwoRayGroundl
Network size	500×500 m^2
Fusion time	0.5
Primary user detection type	Energy detection
Mobility	Random waypoint model
Spectrum sensing	Cooperative spectrum sensing (CSS)
Fusion rule	MAJORITY-based
Sensor nodes	Cognitive radio sensor nodes

3.3 Proposed Algorithm Detection Method

Figure 1 displays an instance of two sinkhole nodes (node 3 and 5) in the network. Hop counts in red are the corresponding hop counts of the attacking nodes (3 and 5). Furthermore, in green are the hop counts of the legitimate sensor nodes. At this instance, the change in position values for all nodes are zeros since the detection have not yet taken place. The hop counts are sent to the base station using hop by hop transmission.

After the base station had received all the hop counts from neighbor nodes, it then updates the nodes hop count database with new hop counts. The base station then sorts the hop counts in an ascending order. As it sorts the hop counts sequentially, the change in positions is also affected. If a node moves one step to the left or to the right its change in position value will be incremented by 1. As such, a node with the smallest hop count value will have a greater change in position value if it was to be moved to the far end in Fig. 2. As such, it will also affect change in position of nodes which were to its left before its move, as show in Fig. 2.

Figure 2 represents a state of the base station dataset after node 3 had been moved to the far left because of its small hop count value.

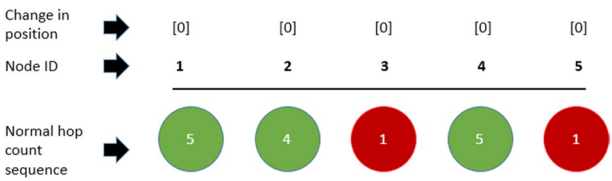


Fig. 1 Normal hop count sequence

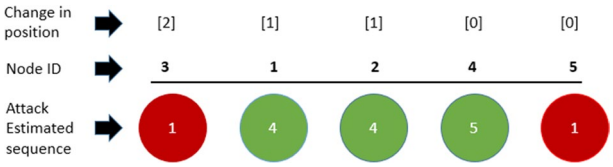


Fig. 2 sink node dataset after the first instance of sort took place

The movement affected node 1 and 2′s change in position values and the values are incremented by 1 as they moved to the right by one unit. The change in position of node 3 is now 2 because it was moved two units to the left. At this stage node 5 is not yet moved to the leftmost side because the base station sorts the hop counts sequentially.

If a node shifts j units, that j value becomes its change position value. That is, if a node shifts 4 units for that instance 4 is its change in position. And any node that is non-malicious but is forced to change its position, its current change in position value will be the initial value plus 1. From Fig. 3, node number 3 shifted 2 units to the left and its changed in position value changed from zero to 2. Every node before node 3 will shift one place to the right in order to make space for it; thus their current change in position values are $0+1$ which is 1. Node 3 will also change position in a case where there are other sinkhole nodes in the network. It will shift one unit to the right for each instance and its current change in position will be initial change in position value plus one which will be 3 $(2+1)$, shown in Fig. 3.

Now that node 5 was at the most far right, it affects every node in the dataset and hence, causes them to shift one position to the right to make a position available for node 5. For each initial change in position of all affected nodes, one (1) will be added to the initial change in position, therefore the current change in position will be given by initial change in position $+1$.

The change in position value for node g in the ith attack estimated sequence is calculated as follows:

$$p_j(g) = \left| C\left(\widehat{h}_j, g\right) - C\left(\overline{h}_j, g\right) \right|, \quad 1 \le g \le N \tag{1}$$

where $C(*, g)$ signifies the change in position of node s in the attack estimated sequence *. Then the average change in position value of node s in the *ith* attack estimated sequence is computed as follows:

$$P(g) = \sum_{j=0}^{n} \frac{p_j(g)}{n} \tag{2}$$

where $P(g)$ is the sample mean of the change in position value. With regard to the above definition, the first theorem come into existence.

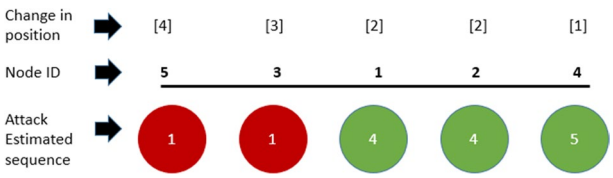


Fig. 3 Sink node dataset after the second instance of sorting

Theorem 1 *A node d is a sinkhole node if its average change in position P(g) is more than a bound K given by:*

$$K = \frac{i}{N-1}(\theta + i) \tag{3}$$

Proof Suppose the nodes 1 up to i are sinkhole nodes and all the others are legitimate, the change in position of the sth node in any sample is a random variable denoted by $P(s)$. In a case of a single sinkhole node, it is noticed that a sinkhole node s with change in position $p(s)$ may make at most $p(s)$ number of nodes change in position by 1.

In computing the average change in position value, the sink node will first compute the change in position value for each of the observed attack estimated sequences. In the example, there are two attack estimated sequences, so $p_1(3)$ will denote the first attack sequence and $p_2(3)$ will denote the second attack estimated sequence. For first instance $p_1(3)$ is found to be 2 and $p_2(3)$ is found to be 1, both are calculated as follows:

Using equation Eq. 2, we get the average change in position of node 3 to be 3/2 which is similar to 1.5. The equation is as follows $P(3) = \sum_{j=1}^{2} \frac{2+1}{2} = 1{,}5$,

Notice that node 3′s final change in position value is not 2 but 3 which is the effect of sinkhole node 5. Node 3 only affected two nodes 1 and 2, respectively. This can be further explained as a sinkhole node's final change in position can be affected by other sinkhole nodes, while the number of non-malicious nodes it affected is equal to the first change in position of the affected sinkhole nodes. Node 3 affected two non-malicious nodes which is equal to its first change in position $p_1(3) = 2$. Then sinkhole node 5 changed position to the most left entry of the attack estimated sequence and decreased node 3′s change in position to 1, which is node 3′s final change in position $p_2(3) = 1$. However, it is visible that from the Figs. 2 and 3 that node 1 and 2′s change in position are both 2, due to the two sinkhole nodes 3 and 5. This points out that node 3′s effect on node 1 and 2′s change in positions remains the same. Therefore, given the final change in position of any sinkhole node s, it is highly possible that the actual change in position of $p'(s)$ for node s is $p_i(s) + i - 1$, which happens when all other sinkhole nodes affect s's change in position in the same direction which is opposite to node s's initial shift direction. Moreover, the extreme number of affected nodes due to a change in position of node s is given by $p_i(s) + i - 1$. Furthermore, the probability that a node's change in position is changed by s is upper bounded by:

$$\frac{p(s) + i - 1}{N-1} \tag{4}$$

Considering the fact that a non-malicious node cannot change its position value on its own, therefore given $p_i(s)$, the expected change in position of any non-malicious node g due to node s denoted as $p^s(g)$ can be expressed as:

$$E(p^s(g)|p(s)) \leq \frac{p(s) + i - 1}{N-1} \forall (i+1) \leq g \leq N$$

$$E(E(p^s(g)|p(s))) \leq E\left(\frac{p(s) + i - 1}{N-1}\right)$$

$$\frac{E(p^s(g))E(p(s))}{E(p(s))} \leq \frac{E(p(s)) + i - 1}{N - 1}$$

$$\approx E(p^s(g)) \leq \frac{E(p(s)) + i - 1}{N - 1} \forall (i + 1) \leq g \leq N \tag{5}$$

Therefore, the expected change in position of any legitimate node g caused by sinkhole nodes is upper bounded by:

$$E(p(s)) \leq \sum_{s=1}^{i} E(p^s(g)) \leq \frac{\sum_{s=1}^{i} E(p(s)) + i - 1}{N - 1}$$

$$\leq \frac{\theta + i^2}{N - 1} \quad \forall (i + 1) \leq g \leq N \tag{6}$$

where θ is the mean average change in position of all sinkhole nodes; computed as follows:

$$\theta = E\left[\frac{\sum_{s=1}^{i} p(s)}{i}\right] = \sum_{s=1}^{i} \frac{E(p(s))}{i} \tag{7}$$

In Eq. 7, the first inequality holds since the extreme total change in position of a non-malicious node caused by sinkhole nodes happens when all sinkhole nodes make g to move to the same direction. As the sample size becomes sufficiently large, it guarantees that a sample mean becomes a great estimator of expectation stated in Eq. 7. When the expectations are replaced with sample mean, the equations become:

$$P(g) \leq \frac{i}{N - 1}(\theta + i) \quad \forall (i + 1) \leq g \leq N \tag{8}$$

Let K denotes the upper bound on the right–hand side of Eq. 8. Then the average change in position of every non-malicious node g is upper bounded by K. Moreover, if a node g has an average change in position $P(g) > K$, it must be a sinkhole node.

Theorem 1 yields an adequate condition for node s being a sinkhole node if $P(s) \geq K$ implies s is a sinkhole node, it again yields an imperative condition for node g being a legitimate node that is, if g is a legitimate node then $P(g) \leq K$ holds.

If the attack estimate sequence is $\{n_1, n_2, \ldots, n_i\}$ then the other nodes $\{n_{i+1}, n_{i+2}, \ldots, n_N\}$ are legitimate nodes. As such, the sample average and sample variance of change in position of legitimate nodes can be computed as follows:

$$\hat{\mu} = \sum_{j=i+1}^{N} \frac{P(g_j)}{N - i} \tag{9}$$

$$\hat{\sigma}^2 = \sum_{j=i+1}^{N} \frac{\left(P(g_j) - \hat{\mu}\right)^2}{(N - i) - 1} \tag{10}$$

The nodes whose change in positions are greater than the second threshold are sinkhole nodes. That is if $P(t) > \hat{\mu} + 3\hat{\sigma}$ then node t must be a sinkhole node s.

We assumed that sinkhole nodes always report a hop count of 1 to their neighborhood and that all other legitimate node reports a hop count of more than 1. After the nodes have transmitted their neighborhood information to the base station, the base station then sorts the hop counts of the nodes in ascending order. This means nodes with significantly lower hop counts are placed at the head of the queue.

Using a change in position the base station computes the change in position of each node as it sorts the hop counts. Figure 4 illustrates the flow chart of HCOBASSA algorithm in details.

3.4 Statistical Analysis

The student's t test statistics was used to check the significant difference between the means of the average probabilities of detection and probabilities of false positive. As working with independent or unpaired samples the significant difference of variance using the F test statistic was first checked in order to choose the appropriate T test (assuming equal variance or assuming unequal variance). The hypothesis was stated as follows:

H0 There is no significant difference in the averages of the probabilities of both schemes.

H1 There is significant difference in the averages of the probabilities of both schemes.

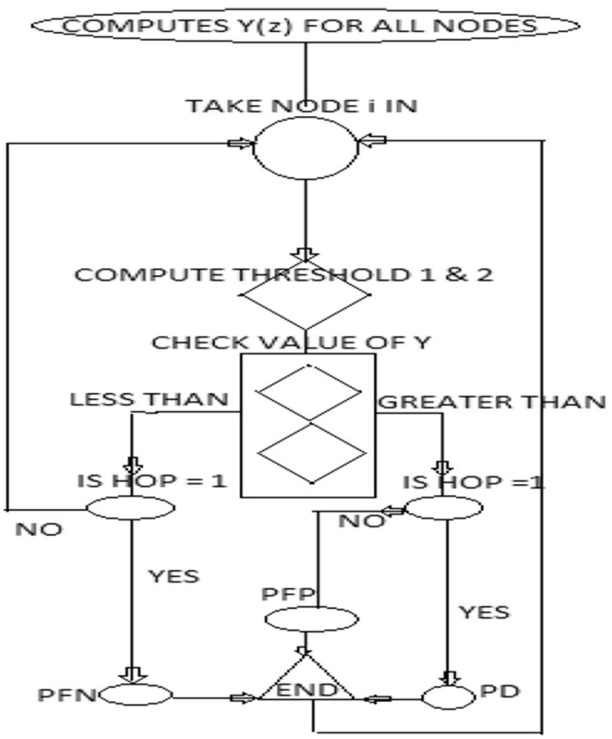


Fig. 4 detection flow-chat of HCODESSA

H0 is the null hypothesis and H1 is the alternative hypothesis. The null hypothesis states that there is no significant difference in terms of the averages of probability of detection and averages of probability of false positive of the two algorithms, meaning that the performance of the two algorithms are simillar. Whereas, the alternative hypothesis states that there is significant difference in the probabilities which means.

The tests were carried out at a 99% level of significant (α), which means that if we calculate a confidence interval from 100 simulation samples, about 99% of them would contain the true mean of the population. Furthermore, confirming that there is no difference, therefore the performance of the two algorithms is similar.

4 Results and Analysis

This section presents the comparative performance results of the HCODESSA and Ibrahim's algorithm. Figure 5 depicts the comparison of the average probability of detection of the two algorithms.

On average, HCODESSA managed to achieve 100% probability of detection in all the network sizes, whereas the Ibrahim's scheme achieved an average of 70% probability of detection. In this scenario, there were 15% attacking nodes in the network. The HCODESSA outperformed the Ibrahim's by almost 30%. With 25% of the attacking node, HCODESSA's performance dropped to 91% and Ibrahim's probability remained constant. It detected almost the same number of attacking nodes in the varying network sizes.

Though the performance of HCODESSA dropped, it managed to maintain the desirable percentage and its detection rate remained above 80%. It is noticeable that percentages of the probability of detection with 25% and 35% attacking nodes are the same; in other words, HCODESSA achieved a constant probability whilst Ibrahim's could not reach 80% average probability. However, its performance increased marginally to 76% when the percentage of the attacking nodes was increased to 35%.

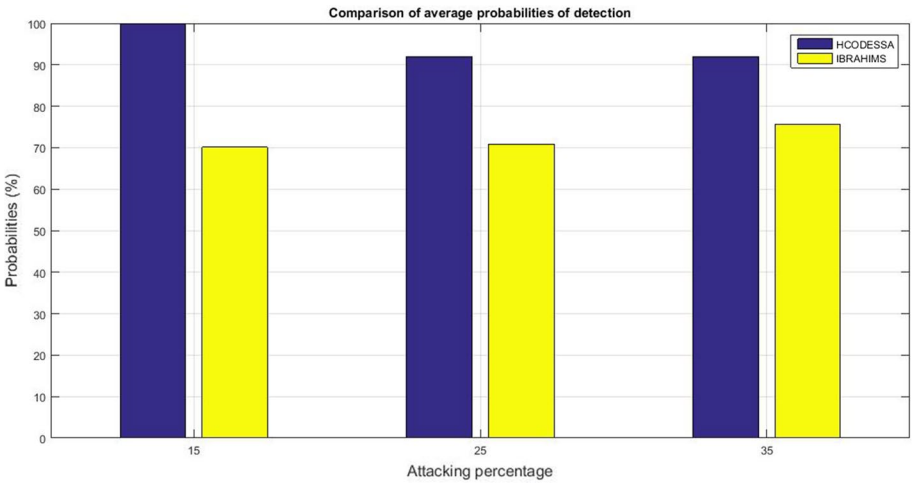


Fig. 5 Comparison of average probability of detection for both HCODESSA and BRAHIM's Hop Count-Based Detection Algorithm

Figure 6 presents the comparative results of the two schemes with regard to probability of false negative. The probability of false negative is the inverse of probability of detection. Higher probability of detection means less probability of false negative and vice versa.

HCODESSA achieved 0% probability of false negative whereas Ibrahim's algorithm achieved 30% when 15% attacking nodes were considered. This means HCODESSA managed to detect all the sinkhole nodes and Ibrahim's only managed to detect 70% of the sinkhole nodes. Although HCODESSA's performance degraded when malicious nodes were increased to 25% and achieved 9% probability of false negative, the results are acceptable in comparison to Ibrahim's. The performance of the Ibrahim's improved and less probability of false negative was achieved in comparison to the results of the scenario with 15% sinkhole nodes.

The performance of Ibrahim's algorithm with respect to probability of false negative has showed a great improvement of about 6% with 35% attacking nodes. Although its results are not that favorable, the improvement is welcome. HCODESSA maintained the same performance for the scenarios with 25% and 35% of the attacking nodes in the network. This is a good performance as the probability of false negative was maintained before 10%.

The probability of false positive was also generated to evaluate the two schemes. The probability checks the percentage of legitimate nodes that are regarded as sinkhole nodes by both algorithms. It is desirable that the probability of false positive be as low as possible, while keeping legitimate sensor nodes in the network and detecting and isolating sinkhole nodes.

Figure 7 shows the comparative results of the average probability of HCODESSA and Ibrahim's schemes. Both algorithms have the same trend of incurring a higher percentage with 25% of sinkhole than on the other percentage of the attacking nodes; 15 and 35. This means that with 25% of the sinkhole nodes in the network, both algorithms regarded a higher number of legitimate sensor nodes as sinkhole nodes than in the case where there are 15% and 35% sinkhole nodes. However, even though this is the case, HCODESSA's probability of false positive is 4.9% which is good compared to 43% of the Ibrahim's algorithms.

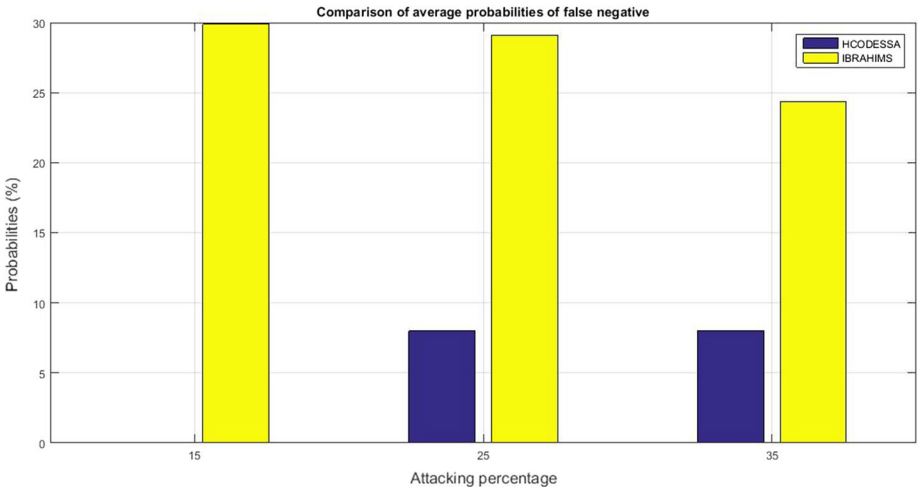


Fig. 6 Comparison of average probability of false negative for both HCODESSA and IBRAHIM's Hop Count-Based Detection Algorithm

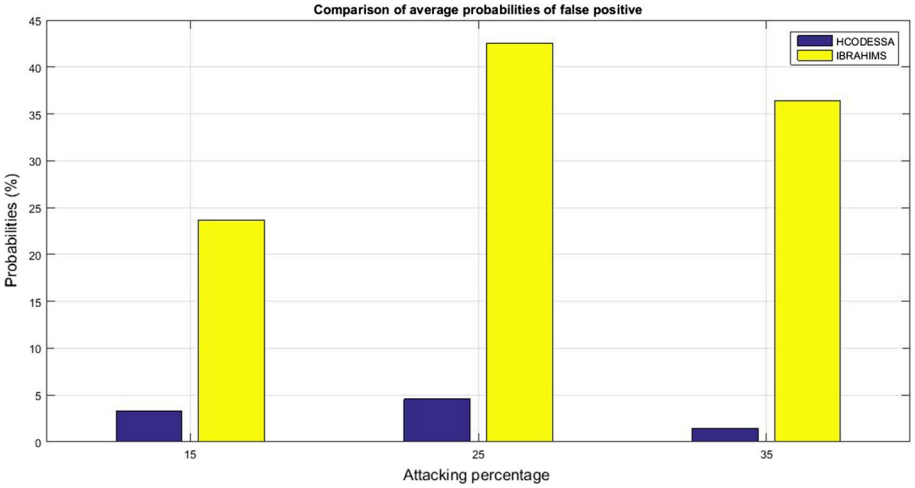


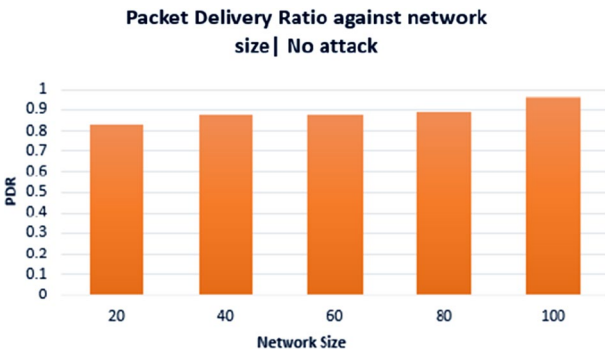
Fig. 7 Comparison of average probability of false positive for both HCODESSA and IBRAHIM's Hop Count-Based Detection Algorithm

HCODESSA achieved a probability of false positive which is less than 5% while Ibrahim's was above 20% in the scenario with 15% malicious nodes. The HCODESSA therefore outperformed the Ibrahim by more than 15%. Although HCODESSA incurred a greater percentage in the scenario with 25% malicious nodes, the results are comparatively good. The Ibrahim's in this case, it achieved 43% while HCODESSA incurred 4.9%. the performance of both algorithms improved in the scenario with 35% malicious nodes.

4.1 The Effect of the Sinkhole Attack

In this Section, we present the simulation results of the effects of the sinkhole attack. Packet loss ratio (PLR) and Packets delivery ratio (PDR) metrics were used to evaluate the two schemes. The results depict the effect of the sinkhole attack on the performance of the network. Figure 8 presents the performance results of the network when there is no attack. The results show that, as the network size increases, the packet delivery ratio also increases. The lowest ratio of packet dropped value is 0.83 (83%) which is almost the same as the maximum ratio value of the packet delivery ratio when

Fig. 8 PDR against network size in the absences of the attack



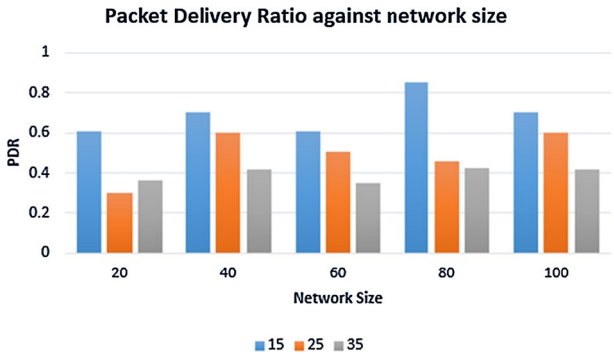


Fig. 9 PDR against network size in the presence of the attack

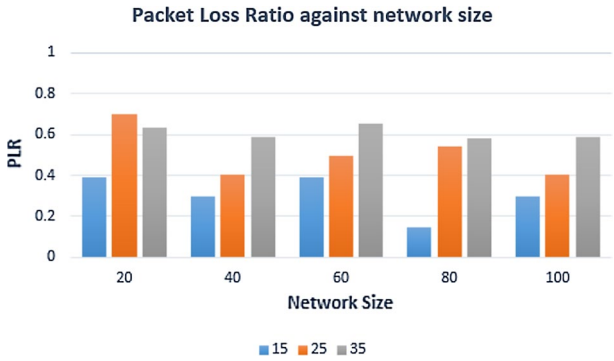


Fig. 10 PLR against network size in the presence of the attack

there is a sinkhole attack in Fig. 5. This shows that the sinkhole attack has a major effect on the packet delivery ratio in the network.

Figure 9 presents the results of PDR in the presence of the sinkhole attack. The results show that the network dropped a significant number of packets compared to Fig. 8. For each network size, the first blue bar denotes a scenario with 15% attacking nodes and the next two bars represent the scenarios with 25% and 35% attacking nodes. However, the position of the sink hole nodes has effect on the performance of the network. When the sink hole node is closer to the base station, it can sink all the network traffic. However, if it's further way at the edge of the network it can drop packets of the nodes in its neighborhood.

The greater impact of the sinkhole attack can be seen in Fig. 10. The results show that a large number of packets were dropped or lost when there were 25% and 35% of malicious nodes in the network. The greatest loss ratio recorded was 0.7 which is equivalent to 70%. This means 70% of the packets were dropped by the sinkhole attacking nodes before they were delivered to the base station.

The severity of the sinkhole attack therefore depends on the number of nodes which it can mislead. This will give a sink hole attack node an opportunity to drop as many packets as possible. In a network scenario with 80 nodes and 15% sinkhole nodes, the effect of the sinkhole attack is moderate in comparison to 25% and 35% scenarios because the sinkhole nodes were not located in a zone where they could mislead a good number of nodes.

4.2 Statistical Tests for Probability of Detection

Figure 11 presents the test statistic for the two algorithms with regard to their probabilities of detection. Figure 11a presents the F-test statistic for checking whether the variances of the two algorithms are equal or not. The calculated value of F test statistic (2.3898) is less than the F critical value (99). This means the null hypothesis is not rejected. We therefore conclude that we are 99% confident that there is not significant difference in the variance of the probability of detection of the two algorithms.

Figure 11b proves that HCODESSA outperformed Ibrahim's algorithm. Assuming equal variance from the F test in Fig. 11a, the t test statistic value of 7.06196 is greater than the t critical two-tailed value of 4.6040. We therefore reject the null hypothesis which states that there is not significant difference in the means of the probability of detection of the two algorithm. We therefore conclude that we are 99% confident that there is significant difference in the probability of detection of the two algorithms.

4.3 Statistical Tests for Probability of False Positive

Figure 12 presents the test statistic of the probabilities of false positive of the two algorithms. Figure 12a presents the F-test statistic for checking whether the variances of the two algorithms are equal or not. The calculated value of F test statistic of 37.21267 is less than the F critical value of 99. The f test statistic is not in the rejection range. We then conclude that statistically, there is no significant difference in the variances of the probabilities of false positive of the two algorithms.

Testing the difference in the Means of the probability of false positive for the two algorithms, it is clear that the t test statistic value of 5.5338 is greater than the t critical two-tailed value of 4.60409. The t test statistic falls in the rejection range. We therefore conclude that at 99% confidence interval there is significant difference in the Means of the probability of false positive of the two algorithms and reject the null hypothesis.

(a) F-test: Two-Sample for Variances (probability of detection)

	HCODESSA	IBRAHIM
Mean	94.66137566	72.21058201
Variance	21.37574536	8.94457378
Observations	3	3
df	2	
F	2.389800329	
P(F<=f) one-tail	0.295002626	
F Critical one-tail	99	

(b) T-test: Two-Sample Assuming Equal Variances (probability of detection)

	HCODESSA	IBRAHIM
Mean	94.66137566	72.21058201
Variance	21.37574536	8.94457378
Observations	3	3
Pooled Variance	15.16015957	
Hypothesized Mean Difference	0	
df	4	
t Stat	7.061963059	
P(T<=t) one-tail	0.001060437	
t Critical one-tail	3.746947388	
P(T<=t) two-tail	0.002120874	
t Critical two-tail	4.604094871	

Fig. 11 Test statistics for probability of detection

(a) F-test: Two-Sample for Variances (Probability of false positive)

	IBRAHIM	HCODESSA
Mean	34.19974584	3.119880675
Variance	92.15450189	2.476428085
Observations	3	3
df	2	
F	37.21267031	
P(F<=f) one-tail	0.026169331	
F Critical one-tail	99	

(b) T-test: Two-Sample Assuming Equal Variances (Probability of false positive)

	IBRAHIM	HCODESSA
Mean	34.19974584	3.119880675
Variance	92.15450189	2.476428085
Observations	3	3
Pooled Variance	47.31546499	
Hypothesized Mean Difference	0	
df	4	
t Stat	5.533796691	
P(T<=t) one-tail	0.002605782	
t Critical one-tail	3.746947388	
P(T<=t) two-tail	0.005211565	
t Critical two-tail	4.604094871	

Fig. 12 Test statistic for probability of false positive

5 Conclusion

The performance of the HCODESSA and Ibrahim's algorithms were evaluated in three different network setups. In each network setup, three scenarios of the number of sink hole nodes were also considered. The number of the sink hole nodes ranged from 15 to 35%. The performance of the schemes were evaluated based on the following metrics: probability of detection, probability of false positive and probability of false negative. The results show that HCODESSA outperformed the Ibrahim's algorithm. Furthermore, the effect of the sink hole attack was evaluated using the packet delivery ratio and packet loss ratio. The results show that the sinkhole attack has adverse effect on the performance of the network. The severity of the sink hole attack also depends on the location of sink hole node and the number of nodes it can mislead.

The results also show that the proposed HCODESSA outperformed the Ibrahim's algorithms in all the performance metrics and scenarios which were considered. In addition, the statistical tests on the average performance of the two algorithms regarding the probability of detection and probability of false positive were conducted. The statistical results show that the proposed HCODESSA outperformed Ibrahim's algorithm.

Funding This work is based on the research supported in part by the National Research Foundation of South Africa (Grant Numbers: 114155).

References

1. Foundation, O. N. (2016). *Software-defined networking (SDN) definition*. https://www.opennetworking.org/sdn-resources/sdn-definition.
2. Bera, S., Misra, S., Roy, S., & Obaidat, M. (2016). Soft-WSN: Software-defined WSN management system for IoT applications. *IEEE System Journal, 16*, 2074–2081.
3. Nachikethas, A., Krishnamachari, J., & Krishnamachari, B. (2014). Software defined networking paradigm in wireless networks: A survey. *ACM Computing Surveys, 47*, 1–11.
4. Delisle, J. (August 2014) *Wireless networks advance software backbone*. Retrieved February 14, 2017, from www.mwrf.com/system/wireless-networks-adavance-software-backbone.
5. Morabito, G. (2015). *Summer school on signal processing*. Retrieved February 14, 2017, from https://sdn-wise.dieei.unict.it/sps2015.pptx.
6. Huang, R., Chu, X., Zhang, J., & Hu, V. (2015). Energy-efficient monitoring in software-defined wireless sensor networks using reinforcement learning: A prototype. *International Journal of Distributed Sensor Networks, 11*, 360428.
7. Kibirige, G., & Sanga, C. (2015). A survey on detection of sinkhole attack in wireless sensor network. *The International Journal of Computer Science and Information Security, 13*(5), 48–52.
8. Winkler, M., Street, M., Tuchs, K., & Wrona, K. (2012). Wireless sensor networks for military purposes. *Springer Series on Chemical Sensors and Biosensors, 13*, 365–394.
9. Prabhu, B., Pradeep, M., & Gajendran, E. (2017). Military applications of wireless sensor network system. *A Multidisciplinary Journal of Scientific Research & Education, 2*, 12.
10. Taiconi, D., Miorandi, D., Carreras, L., Chiti, F., & Fanracci, R. (2010). Using wireless sensor networks to support intelligence transport system. *Ad Hoc Networks, 8*(5), 462–473.
11. SmartWax, A. B. (2016). *Smart IoT technologies for adaptive trafffic management using wireless mesh sensor network*. Retrieved February 20, 2017, from https://advantech-bb.com/Smart-IoT-technologies-for-adaptive-trafffi-management-using-wireless-mesh-sensor-network/.
12. Choi, G., Cho, J., Kim, H., Hong, S. (2009). A sinkhole attack detection mechanism for LQI based mesh routing in WSN. *ICOIN* (pp. 1–5).
13. Sheela, D., Kumar, N., & Mahadevan, G. (2011). A non-cryptographic method of sinkhole attack detection in wireless sensor networks. *Recent Trends in Information Technology (ICRTIT)*. https://doi.org/10.1109/ICRTIT.2011.5972397.
14. Sreelajaa, N., & Pai, G. (2014). Swarm intelligence based approach for sinkhole attack detection in wireless sensor networks. *Elsevier Applied Soft Computing, 19*, 68–79.
15. Keerthana, G., & Padmavathi, G. (2015). A study on sinkhole attack detection using swarm intelligence techniques for wireless sensor networks. *International Journal of Computer Science and Information Technology & Security, 5*(5), 376–380.
16. Zhang, F., Zhai, L.-D., Yang, J.-C., & Cui, X. (2014). Sinkhole attack detection based on redundancy mechanism in wireless sensor networks. *Elsevier Procedia Computer Science, 31*, 711–720.
17. Sindhu, S., Geetha, S., & Kannan, A. (2012). Decision tree based light weight intrusion detection using a wrapper approach. *Expert Systems with Applications, 39*(1), 129–141.
18. Ruby, D., Vijayalakshmi, M., & Kannan, A. (2019). Intelligent relay selection and spectrum sharing techniques for cognitive radio networks. *Cluster Computing, 22*(5), 10537–10548.
19. Vijayakumar, P., Azees, M., Kannan, A., & Deborah, L. (2016). Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks. *IEEE Transactions Intelligent Transportation Systems, 17*(4), 1015–1028.
20. Muthurajkumar, S., Senthilnayaki, B., Venkatalakshmi, K., & Kannan, A. (2019). Intrusion detection system using fuzzy rough set feature selection and modified KNN classifier. *International Arab Journal of Information Technology, 16*(4), 746–753.
21. Muthurajkumar, S., Ganapathy, S., Vijayalakshmi, M., & Kannan, A. (2017). An intelligent secured and energy efficient routing algorithm for MANETs. *Wireless Personal Communications, 96*(2), 1753–1769.
22. Sharmila, S., & Umamaheswari, G. (2011). Detection of sinkhole attack in WSN using message digest algorithms. *Process Automation, Control and Computing (PACC)*. https://doi.org/10.1109/PACC.2011.5978973.
23. Bahekmat, M. Y. M., Yazdi, A., & Sadegi, S. (2012). A novel algorithm for detecting sinkhole attacks in WSNs. *International Journal of Computer Theory and Engineering, 4*(3), 418–420.
24. Santos, A., Cervantes, C., Nogueira, M., & Kantarci, B. (2019). Clustering and reliability-driven mitigation of routing attacks in massive IoT systems. *Journal of Internet Services and Applications, 10*, 18.

25. Teng, H., Liu, Y., Liu, A. N., Cai, Z., Wang, T., & Liu, X. (2019). A novel code data dissemination scheme for Internet of Things through mobile vehicle of smart cities. *Future Generetion Computer Systems, 94*, 351–367.
26. Sun, X., Liu, W., Wang, T., Deng, Q., Liu, A., Xiong, N., et al. (2019). Two-hop neighborhood information joint double broadcast radius for effective code dissemination in WSNs. *IEEE Access, 7*, 88547–88569.
27. Liu, Y., Ma, M., Liu, X., Xiong, N., Liu, A., & Zhu, Y. (2018). Design and analysis of probing route to defense sink-hole attacks for internet of things security. *IEEE Transactions on Network Science and Engineering*. https://doi.org/10.1109/TNSE.2018.2881152.
28. Ngai, E., Liu, J., & Lyu, M. (2007). An efficient intruder detection algorithm against sinkhole attacks in wireless sensor networks. *Computer Communications, 30*, 2353–2364.
29. Salmon, S. (August 2016) *False positive vs. false negative*. https://blogs.ams.org/mathgradblog/2016/08/06/false-positive-vs-false-negative/.
30. Poole, I. (June 2016) *Signal to noise ratio, SNR*. Retrieved October 16, 2017, from www.radio-electronics.com/info/rf-technology-design/rf-noise-sensitivity/receiver-signal-to-noise-ratio.php.
31. Kaviarasu, A., & Devapriya, S. (2014). SNR based adaptive spectrum sensing in cognitive radio networks. *International Journal of engineering Reserach Trends (IJERT), 3*(3), 1438–1442.

Lanka Chris Sejaphala is an MSc Computer Science student with research interest in Software-Defined Wireless Sensor Cognitive Radio Networks, Internet of Things, and sinkhole attack.

Mthulisi Velempini is an IEEE member and active researcher in Medium Access Control protocols, Routing protocols and Security in Computing. He is an emerging researcher in wireless access network technologies.