An Effective Scheme to Detect Jamming attacks on the Physical Layer of WSN Using Localization Algorithms (Convex Hull& RSSI)

Mbeng Atemson
College of Computer Science and Technology
Shanghai University of Electrical Powers
Shanghai, China
mbengatemson@yahoo.com

Abstract
Jamming is one of the main attacks on WSN physical layer, detecting and preventing such an attack is a major challenge, there are many known methods to solve jamming attacks related issues in WSN among which are: spread spectrum, frequency hopping and the use of localization algorithms, which has proven to be one of the most effective methods to localize jammers. We proposed the use of Convex Hull algorithms (mainly Divide and Conquer algorithm and the Quick Hullalgorithm) combined with the Receive Signal Strength Indicator algorithm in order to address these attacks. Our solution consists of using RSSI to identify the jammed nodes location and CH to isolate these jammed nodes from the rest of the network.

Keyword: Jamming, Physical Layer, WSN, Convex Hull, Security, Attacks and Detection, Localization, matplotlib-python, RSSI

1. INTRODUCTION

Wireless Sensor Networks are vulnerable against different physical attacks. Attackers can gain full access to sensor nodes, extract and reveal sensitive and sensed data, or launch DoS attacks. There are mainly two types of physical layer attacks Jamming and Tampering. (Jamming attack is a kind of Denial of Service attack, which prevents other nodes from using the channel to communicate by occupying the channel that they are communicating on. Jamming attacks has as main objective to prevent communication between communicating nodes. Jamming attacks can have some serious consequences to our existence, given that wireless sensors are being embedded into every major technology (devices and networks), a large-scale jamming attack on such devices or networks (military, Health, Energy or Power plants, Telecommunication) will be disastrous and might even results to lost in human lives. Some known defenses against these attacks are: Priority messages, monitoring, authorization, redundancy, encryption, Spread-spectrum, lower Duty cycle, region mapping, mode change Tamper-proofing, hiding. The table below shows WSNs' physical layer attacks.

Table 1: Physical layer attacks, definitions and results

Attack	Attack definition	Attack result
Jamming	Transmitting radio signals at the same frequency band(interference)	-Radio interference -Resource exhaustion
Device Tampering	Direct physical access and capture sensor nodes	-Damaging and modifying physically; stopping/changing nodes services; -Taking full control on the captured node; - Taking over the entire WSN; - Software vulnerabilities; - Launching internal attacks;
Path-Based DoS	Sending many packets to the Sink by attacker	-Nodes' battery exhaustion; -Network disruption; -Falsely excluding nodes from local report; -Reducing the WSN's availability;
Node outage	Stopping Functionality	-Stop nodes' services;

	of the WSNs components, such as a sensor node	-Taking over the WSN and preventing from some communications; -Impossibility reading the gathered data; -Launching other attacks;

Jamming attacks on WSNs is one of the main security threats that affect integrity and availability in these networks. It is a popular Denial of service (DOS) attack on physical layer of network. In jamming, adversaries interfere with the communication frequencies (radio frequencies) being used by the nodes of the network. In jamming, an attacker can simultaneously transmit over the WSN refusing the underlying MAC protocol. Jamming can affect the whole n/w if single frequency is used throughout the n/w and it can cause excessive energy consumption at any node if impertinent packets are injected. By getting those packets receiver's nodes will as well consume energy. The following are some countermeasures against jamming on the physical layer:

- Spread Spectrum
- Frequency hopping
- Localization algorithms

Our focus will be on using localization algorithm to detect and prevent jamming attacks on the WSN physical layer by using the convex hall algorithm.

This paper is structured as follows. In section IIwe talk about the preliminaries where we describe different types of existing localization algorithms. In section III we talk about the system model and security requirements; we further introduce some requirements and materials. In section IV we propose a scheme and our algorithm. In section V we talk on the performance analysis of our proposed algorithm while giving advantages over existing methods. We conclude in section VI.

2. PRELIMINARIES

Localization is determining the position of a device or a node, relative or absolute with an appropriate accuracy [1]. Localization is an implicit bargain in wireless sensor networks.Three localization algorithms can be identified; in this section we give a vivid description and some advantages. These algorithms are:

1) Convex Hull[8]

The convex hull, also known as the convex envelope, of a set X is the smallest convex set of which X is a subset. Formally,

Definition: The convex hull H(X) of a set X is the intersection of all convex sets of which X is a subset.

If X is convex, then obviously H(X) = X, since X is a subset of itself. Conversely, if H(X) = X, X is obviously convex.

Theorem: If X is closed and bounded, so is H(X).

Theorem: H(X) is the union of all straight lines joining all pairs of points in X (where the line includes the end-points).

Proof: Let A, B be any two points in X. They are also in any convex set Y containing X. The line joining them must also lie within Y hence it must lie within the intersection of all convex sets containing X, i.e. within H(X). Thus, the union of all such straight lines is a subset of H(X). in other words, the convex hull of a set of points S is the intersection of all half-spaces that contain S. A half space in two dimensions is the set of points on or to one side of a line. This notion generalizes to higher dimensions. A half-space is the set of points on or to one side of a plane and so on.

Note that the convex hull of a set is a closed "solid" region which includes all the points on its interior. Often the term is used more loosely in computational geometry to mean the boundary of this region, since it is the boundary that we compute, and that implies the region.

2) RSSI [7]

the RSSI technique is used as the distance estimation method. The RSSI technique is based on the received signal strength indicator to estimate the distance between neighboring nodes. In free-space, the RSSI value is inversely proportional to the squared distance between the transmitter and the receiver. The radio signals attenuate with the increase of the distance. The propagation of the radio signals may be affected by reflection, diffraction, and scattering. Especially in indoor environments, such effects may impact the measurement accuracy. Therefore, this technique is more suitable for outdoor, rather than indoor applications. This technique has the advantage of requiring no additional hardware since the RSSI feature exists in most wireless devices, and there is no significant impact on the local power consumption.

RSSI is affected from some factors that cause localization errors and reduce accuracy. These errors can be classified into two groups as environmental and device errors. Environmental errors are caused due to wireless communication channel. The causes are usually multi path, shadowing effect, and interference from other RF sources. Device errors are usually caused due to calibration errors, and the important issue here is to keep constant transmit power even under the circumstances of device differences and depleting batteries. Signal samples, even with the same transmit power, show some standard deviations due to atmospheric conditions. Temperature, for example, has a little effect on a signal. However, rain can affect the signal considerably. Especially, in localization based on the received signal strength method, this will cause less accuracy and reliability.

Centroid Localization [4][3]

Centroid Localization Algorithm (CLA) is a coarse-grained algorithm that depends on reference nodes or anchors for location estimation. A free node locates its position using the intersection of the connectivity of the regions as defined by the radio range of each anchor. This is represented in the mathematical expression below:

If there are N jammed nodes (X 1, Y 1), (X2, Y2) ... (X N, Y N), the position of the jammer can be estimated by:

$$(\hat{X}_{jammer}, \hat{Y}_{jammer}) = \left(\frac{\sum_{i=1}^{N} X_i}{N}, \frac{\sum_{i=1}^{N} Y_i}{N}\right) \quad (1)$$

Figure 1: Localization Algorithm Equation (1)

Weighted Centroid Localization [3]

Weighted Centroid Localization adds different contributions to the involved node coordinate information in estimating the location of the target node. We usually call the contribution as weight. One nature metric to be used as weight is the distance between the jammer to the boundary node. By adding the weighing factor into the centroid method, the jammer's position is estimated as:

$$(\hat{X}_{jammer}, \hat{Y}_{jammer}) = \left(\frac{\sum_{i=1}^{N} \omega_i X_i}{\sum_{i=1}^{N} \omega_i}, \frac{\sum_{i=1}^{N} \omega_i Y_i}{\sum_{i=1}^{N} \omega_i}\right) \quad (2)$$

Figure 2: Localization Algorithm Equation (2)

Virtual Force Iterative Localization [4][3]

Virtual Force Iterative Localization (VFIL) [4] tries to improve CL by adjusting the estimation of CL according to the jammed node's distribution. VFIL first estimates the jammer's transmission range, then generates an estimated jammed region in a circle shape (This circle uses the estimation result of CL as the center and covers all jammed nodes while all boundary nodes fall outside of the region.), and after that, it changes the center of the estimated jammed region in the network iteratively in order to cover the most jammed nodes. VFIL assumes that when the estimated jammer's location equals to the true position, the estimated jammed region will overlap with the real jammed region. To move the estimated location to the real jammer's location, VFIL runs multiple times using two virtual force called pull and push. At each iterative step, the jammed nodes that are outside of the estimated jammed region should pull the jammed region toward themselves, which is the pull force, while the unaffected nodes that within the estimated jammed region should push the jammed region away from them, which is the push force.

Let (X0, Y0) be the estimated position of the jammer, (Xi, Yi) be the position of a jammed node, and (Xj, Yj) be j I the location of an affected node. The force F pull and F push as normalized vectors that point to/from the estimated jammer's position:

$$F^{i}_{pull} = \left[\frac{X_i - \hat{X}_0}{\sqrt{(X_i - \hat{X}_0)^2 + (Y_i - \hat{Y}_0)^2}}, \frac{Y_i - \hat{Y}_0}{\sqrt{(X_i - \hat{X}_0)^2 + (Y_i - \hat{Y}_0)^2}}\right],$$

$$F^{j}_{push} = \left[\frac{\hat{X}_0 - X_j}{\sqrt{(\hat{X}_0 - X_j)^2 + (\hat{Y}_0 - Y_j)^2}}, \frac{\hat{Y}_0 - Y_j}{\sqrt{(\hat{X}_0 - X_j)^2 + (\hat{Y}_0 - Y_j)^2}}\right] \quad (3)$$

Figure3: Localization Algorithm Equation (3)

3. System Model and Security Requirements

As early mentioned, our solution for detecting and preventing jamming in WSN is the use of localization algorithms such as RSSI and the convex hull algorithms. The convex hull is a non-mathematical algorithm used in game theory mostly for collision detection for convex polyhedral which are well defined (e.g. GJK and SAT) [], geographic information systems, image processing and pattern recognition. The convex hull is the smallest convex polygon containing all given points or nodes. Computing the convex hull means that non-ambiguous and efficient representation of the required convex shape is constructed. The complexity of the

corresponding algorithms is usually estimated in terms of n, the number of input points, and sometimes also in terms of h, the number of points on the convex hull.

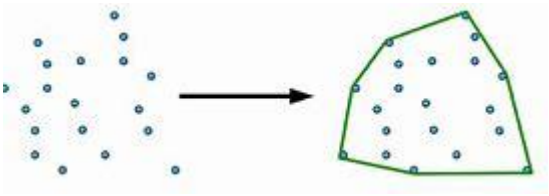


Figure 5: convex hull graphical representation

Two techniques have been employed to solve convex hull: brute force technique (time complexity= O (n^3)) and the divide and conquer technique (time complexity= O (n)). We will focus on the divide and conquer technique which is a much efficient algorithm.

Meanwhile the RSSI is a measurement to show the condition of received power in the anchor nodes and its commonly used in most wireless communication standard. Where The signal strength is proportional to $1/d^2$ which is a function of distance and is generally affected environmental conditions.

$$RSSI(d) = -10n\log_{10}(d) - C$$

n = path loss exponent, C = environment constant.

Theory:

The following steps are for finding the RSSI:

1- Measure RSSI at unknown node from fixed nodes or anchors
2- Estimate distance d from RSSI by using the Log normal shadowing model RSSI = $-10\log_{10}(d)-C$
3- Target is located by trilateration. (wireless triangulation)

Theory:

The following steps are for finding the convex hull of points:

1- Find the point with minimum x-coordinate let's say, min_x and similarly the point with maximum x-coordinate, max_x.

2- Make a line joining these two points, say **L**. This line will divide the whole set into two parts. Take both the parts one by one and proceed further.

3- For a part, find the point P with maximum distance from the line L. P forms a triangle with the points min_x, max_x. The points residing inside this triangle can never be the part of convex hull.

4- The above step divides the problem into two sub-problems (solved recursively). Now the line joining the points P and min_x and the line joining the points P and max_x are new lines and the points residing outside the triangle is the set of points. Repeat point no. 3 till there is no point left with the line. Add the end points of this point to the convex hull.

$$H_{convex}(X) = \{\sum_{i=1}^{k} \alpha_i x_i \mid x_i \in X, \alpha_i \in R, \alpha_i \geq 0, \sum_{i=1}^{k} \alpha_i = 1, k = 1, 2, ...\} \quad (4)$$

Figure 4: Convex hull Algorithm Equation (4)

4. PROPOSED SCHEME

The main idea is to use a fully-range-based localization algorithm (RSSI) along with the convex hull algorithm mainly to detect and furthermore prevent and isolate the jamming areas or regions in the WSN.

The Received Signal Strength Indicator (RSSI) is based on the physical fact of wireless communication that theoretically, the signal strength is inversely proportional to the squared distance between a pair of sensor nodes. A known radio propagation model is used to convert the received signal strength into distance. In RSSI techniques, either empirical or theoretical models are used to translate signal strength into distance [3]

Among the range-based measurement techniques, the RSSI technique is the most common techniques, cheapest and simplest, since its low cost because it does not require additional hardware (e.g. infra-red or ultrasonic). However, one major disadvantage to this RSSI technique is: it'ssusceptible to interference (noise and obstacles, for indoor environment); this disadvantage has made RSSI suitable for use in our experiment since our goal is to detect jamming attacks in the physical layer of WSN. The RSSI algorithm is used to locate the jammer or jamming area due to the signal strength it receives from the nodes, based on the signal strength a distance or location can be identified. After the jamming or interference area

has been identified we can then use the convex hull algorithm to isolate that area from the rest of the network by building a convex around the infected area.

Algorithm Model:

The architecture of our solution is represented in the diagram below, the logic is simple first we run a script to test the signal strength of communicating nodes in WSN depending on the signal quality we can then proceed by calculating the distance and the convex hull.

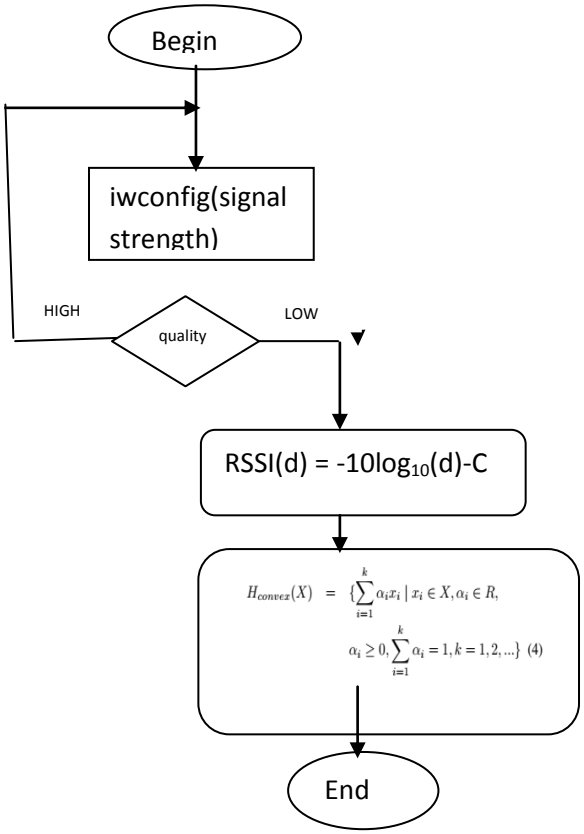


Figure 6: Algorithm representation

```
import subprocess
import time
import argparse

parser = argparse.ArgumentParser(description='Display WLAN signal strength.')
parser.add_argument(dest='interface', nargs='?', default='wlan0',
```

```
                help='wlan interface (default: wlan0)')
args = parser.parse_args()

print '\n---Press CTRL+Z or CTRL+C to stop.---\n'

while True:
cmd = subprocess.Popen('iwconfig %s' % args.interface, shell=True,
stdout=subprocess.PIPE)
   for line in cmd.stdout:
      if 'Link Quality' in line:
         print line.lstrip(' '),
elif 'Not-Associated' in line:
         print 'No signal'
time.sleep(1)
```

Figure 7: RSSI Script to detect nodes strength

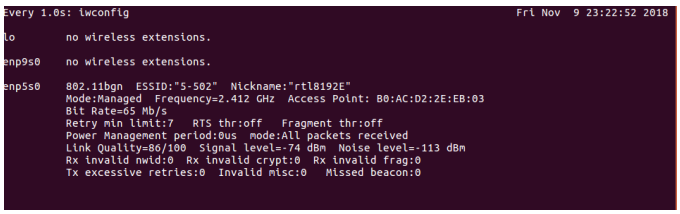


Figure 8: iwconfig command

The above python script when executed gives the wireless devices signal strength.

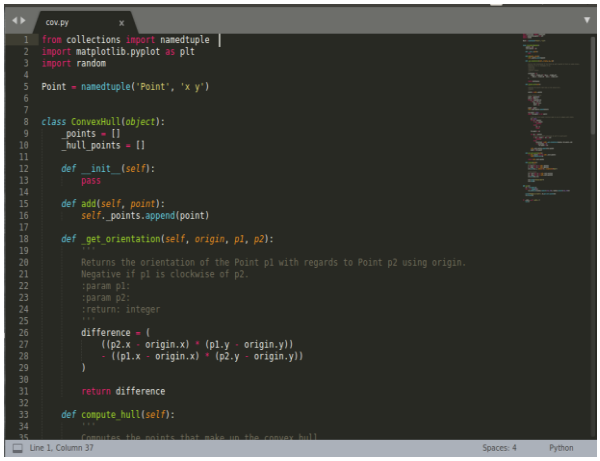
5. PERFORMANCE ANALYSIS

Given the set of points for which we must find the convex hull.

Nodes: X, Y, P,Range=50

Suppose our network comprises hundreds of nodes communicating with each other. We first run python script to calculate the RSSI. The signal strength quality of each communicating node in the WSN is tested continuously to make sure no disruptions or interference has occurred. We should note that the nodes should be transmitting at same frequency at all time. If

during the test, the frequency is higher or lower than its supposed to be we then through our script identify the nodes whose frequency have been tampered or modified. Once we obtain the distances of modified frequency nodes, we can then build a convex around the infected area to isolate those nodes from others and prevent them for further jamming the network. We can use the below python codes to calculate the convex hull of contaminated points on our network.



```
from collections import namedtuple
import matplotlib.pyplot as plt
import random

Point = namedtuple('Point', 'x y')


class ConvexHull(object):
    _points = []
    _hull_points = []

    def __init__(self):
        pass

    def add(self, point):
        self._points.append(point)

    def _get_orientation(self, origin, p1, p2):
        '''
        Returns the orientation of the Point p1 with regards to Point p2 using origin.
        Negative if p1 is clockwise of p2.
        :param p1:
        :param p2:
        :return: integer
        '''
        difference = (
            ((p2.x - origin.x) * (p1.y - origin.y))
            - ((p1.x - origin.x) * (p2.y - origin.y))
        )

        return difference

    def compute_hull(self):
        '''
        Computes the points that make up the convex hull.
```

Figure 9: convex hull sample codes.

From the convex hull results or calculations, we can identify the area or region which has been jammed and the problem can be resolved.

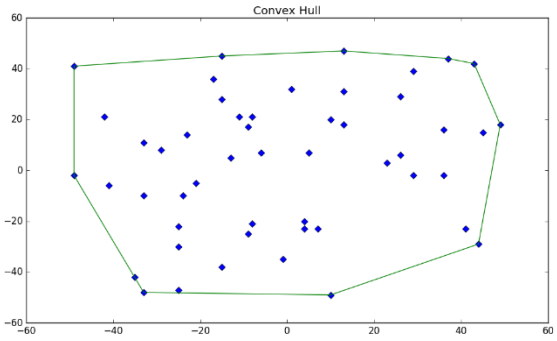


Figure 10: convex hull

('Points on hull:', [Point(x=-49, y=-2), Point(x=-33, y=-48), Point(x=10, y=-49), Point(x=44, y=-29), Point(x=49, y=18), Point(x=43, y=42), Point(x=37, y=44), Point(x=13, y=47), Point(x=-15, y=45), Point(x=-49, y=41), Point(x=-49, y=-2)])

Advantages over other algorithms

Cheap and cost effective

Reliable

Effective, accurate and precise

Reduces computational time due to its time complexity

6. Conclusions and Future Work

In this study of detecting and preventing jamming attacks on physical layer of Wireless Sensor Networks, we have contributed by proposing a method to detect and localizing jamming attacks on WSN devices (which includes using two algorithms RSSI for measuring and calculating communication nodes signal strength or quality and the convex hull for building a protective convex or triangle around the infected nodes). We discussed already existing solutions to this problem. There are many other methods or ways to detect and prevent jamming in WSNs physical layer for example, using spread spectrum (to mitigate signals interference for security and reliability purposes), frequency hopping techniques, honeypots technique (to reduce the effectiveness of jammer as well as to decrease the jamming rate) or using algorithms.Some advantages over existing methods are: it is cheap, reliable, effective and accurate. During our research we discovered that our method is not only cost effective but also can conveniently detect jam nodes and further isolating them. In the future we intend to carry more detailed experiments to test to a greater degree the effectiveness of our proposed method in solving jamming attacks on the physical layer of WSN.

Acknowledgment

I want to thank my supervisor Wen Mi for all the good advice and for setting me on the path during research, my family and friends for helping me reach my set goals and objectives.

References

[1] Neha Thakur and ArunaSankaralingam. “Introduction to jamming attacks and prevention techniques using Honeypots in wireless networks”, IRACST-International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN:2249-9555 vol.3, No.2, April 2013.

[2] Rajani Muraleedharan and Lisa Ann Osadciw. “Jamming attack detection and countermeasures in wireless sensor network using ant system”

[3] HongboLiu,WenyuanXu,YingyingChen,Zhenhua Liu. “Localizing jammers in wireless network”

[4] Tianzhen Cheng, Ping Li, Sencun Zhu. “an algorithm for jammer localization in wireless sensor network”

[5] Aleksandar Donev, salvatoreTorquato, Frank H.S, Robert Connelly. “a linear programming algorithm to test for jamming in hard-sphere packings”, 2004, Journal of computational physics 197 (2004) 139-166.

[6] SahabulAlam and Debashis De. “analysis of security threats in wireless sensor network”, International Journal of Wireless & Mobile Network (IJWMN) vol 6, No. 2, April 2014.

[7] UgurBekcibasi1,MahmutTenruh. “Increasing RSSI Localization Accuracy with Distance Reference Anchor in Wireless Sensor Networks”. Vol. 11, No. 8, 2014.

[8] Constantin P. Niculescu, Lars-Erik Persson. “Convex Functions and Their Applications A Contemporary Approach”. Library of Congress Control Number: 2005932044 ISBN-10: 0-387-24300-3 ISBN-13: 978-0387-24300-9.