A DISTRIBUTED SCHEME TO DETECT WORMHOLE ATTACKS IN MOBILE WIRELESS SENSOR NETWORKS

by
OYA ŞİMŞEK

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
February 2011

A DISTRIBUTED SCHEME TO DETECT WORMHOLE ATTACKS IN MOBILE WIRELESS SENSOR NETWORKS

APPROVED BY

Assoc. Prof. Dr. Albert Levi ....

(Thesis Supervisor)

Assoc. Prof. Dr. Erkay Savaş ....

Asst. Prof. Dr. Hüsnü Yenigün ....

Assoc. Prof. Dr. Yücel Saygın ....

Assoc. Prof. Dr. Özgür Erçetin ....

DATE OF APPROVAL ....

A DISTRIBUTED SCHEME TO DETECT WORMHOLE ATTACKS IN MOBILE WIRELESS SENSOR NETWORKS

Oya Şimşek
Computer Science and Engineering, MS Thesis, 2011
Thesis Supervisor: Assoc. Prof. Albert Levi

Keywords: Wormhole Attack, Security, Mobile Wireless Sensor Networks

Abstract

Wireless sensor networks are composed of sensor nodes which are small, battery-powered devices having limited resources. Sensor nodes collect data from environment, and transmit them via their radio communication medium towards a base station. Although majority of wireless sensor applications use static sensor nodes, sensor node can be mobile either by itself, or due to environmental factors such as wind, water, or deployment of sensor nodes on moving objects.

It is not easy to control sensor nodes once they are deployed in a hostile environment. Due to mostly being unattended, sensor nodes become open to physical attacks such as wormhole attack, which is our focus in this thesis. In wormhole attack, an attacker tunnels messages received in one part of the network over a low-latency wormhole link and replays them in a different part of the network. By doing so, the attacker makes two distant nodes believe that they are in the communication range of each other. The low-latency tunnel attracts network traffic on the wormhole link which can empower the attacker to perform traffic analysis, denial of service attacks; collect data to compromise cryptographic material; or just selectively drop data packets through controlling these routes using the wormhole link.

In this thesis, we propose a distributed wormhole detection scheme for mobile wireless sensor networks in which mobility of sensor nodes is utilized to estimate two network features (i.e. network node density, standard deviation in network node density) through using neighboring information in a local manner. Wormhole attack is detected via observing anomalies in the neighbor nodes' behaviors based on the

estimated network features and the neighboring information. We analyze the performance of proposed scheme via simulations using different system parameters. The results show that our scheme achieves a detection rate up to 100% with very small false positive rate (at most 1.5%) if the system parameters are chosen accordingly. Moreover, our solution requires neither additional hardware nor tight clock synchronization which are both costly for sensor networks.

MOBİL KABLOSUZ DUYARGA AĞLARINDA SOLUCAN DELİĞİ SALDIRILARINI TESPİT ETMEK İÇİN DAĞITIK BİR ŞEMA

Oya Şimşek
Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans Tezi, 2011
Tez Danışmanı: Doç. Dr. Albert Levi

Anahtar Kelimeler: Solucan Deliği Saldırısı, Güvenlik, Mobil Kablosuz Duyarga Ağları

Özet

Kablosuz duyarga ağları küçük, pille çalışan, sınırlı kaynaklara sahip aygıtlardan oluşur. Duyarga düğümleri çevreden veri toplar ve bu verileri radyo iletişim ortamı üzerinden baz istasyonuna iletirler. Kablosuz duyarga ağı uygulamalarının çoğunluğu statik duyarga düğümlerini kullansa da duyarga düğümleri kendiliğinden, ya da rüzgar, hava gibi çevresel etkenlerden, ya da duyarga düğümlerinin hareketli nesneler üzerine konuşlandırılmasından dolayı mobil olabilir.

Duyarga düğümleri saldırılara açık bir ortamda konuşlandırıldıklarında güvenliklerini sağlamak kolay değildir. Genelde gözetimsiz olduğundan dolayı, duyarga düğümleri bu tezin odağını oluşturan solucan deliği saldırısı gibi fiziksel saldırılara açık hale gelirler. Solucan deliği saldırısında, saldırgan ağın bir bölgesinde alınan mesajları düşük gecikmeli solucan deliği bağlantısı üzerinden gönderir ve bu mesajları ağın başka bir bölgesinden tekrar yayınlar. Böyle yaparak, saldırgan birbirine uzak iki düğümü birbirlerinin iletişim alanında olduklarına inandırır. Düşük gecikmeli tünel, ağ trafiğini solucan deliği bağlantısı üzerine çeker. Saldırgan, solucan deliği bağlantısını kullanan bu rotaları kontrol ederek trafik analizi ve servis reddi saldırılarını gerçekleştirebilir; şifrelemeyle ilgili bilgileri çıkarmak için veri toplayabilir; ya da veri paketlerini seçerek düşürebilir.

Bu tezde, mobil duyarga ağlarında solucan deliği saldırısını tespit etmek için dağıtık bir şema önerdik. Bu şemada lokal komşuluk bilgilerinini kullanarak iki farklı ağ özelliğinin (ağ düğüm yoğunluğu, ağ düğüm yoğunluğunun standart sapması) hesaplanmasında duyarga düğümlerinin mobilitesinden yararlanıldı. Solucan deliği saldırısı, hesaplanan ağ özellikleri ve komşuluk bilgileri baz alınarak, komşu düğümlerin davranışlarındaki anormalliklerin gözlemlenmesi yoluyla tespit edilir. Önerilen şemanın performansını simülasyonlarla analiz ettik. Sonuçlar, sistem parametreleri uygun bir şekilde seçildiğinde şemamızın %100'e varan bir doğru tespit oranına eriştiğini gösterdi. Bununla birlikte, hatalı tespit oranı %1.5 gibi çok düşük bir düzeyde kaldı. Üstelik, çözümümüz duyarga düğümleri için pahalı sayılabilecek bir ek donanıma ya da katı bir zaman senkronizasyonuna ihtiyaç duymaz.

To my family

Acknowledgements

First of all, I would like to thank my thesis advisor, Albert Levi, for all his support throughout my university experience including guiding and motivating me in all of my works.

I also thank Hüsnü Yenigün, Erkay Savaş, Yücel Saygın, and Özgür Erçetin for devoting their time amongst their high volume schedule and joining my jury.

I thank all classmates at FENS 2001 Lab.

I thank my dearest Ahmet Hakan Göral for his mental support during my thesis.

I thank my family for supporting me in every aspects of my life and growing me up to this day.

I specially thank to my sister Emel Şimşek for being there when I need her to be and supporting me no matter what happens.

I also thank Scientific and Technological Research Council of Turkey (TÜBİTAK) for funding me by BİDEB scholarship.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

As a result of significant advances in hardware manufacturing and wireless communication technology along with efficient software algorithms, wireless sensor networks [1] emerged as a promising network infrastructure for various applications such as environmental monitoring, medical care, industry and agriculture, military surveillance, target detection and tracking. Wireless sensor networks composed of many battery-powered, small, and resource constraint devices called sensor nodes. Sensor nodes are capable of sensing environment, processing data, and communicating with other sensor nodes in the network using short-range radio. Wireless sensor networks can be deployed randomly which can be viewed as advantage if we consider the deployment in inaccessible terrains or disaster relief operations. However, in such random deployments, sensor network protocols and algorithms need to be self-organized. Although majority of wireless sensor applications use static sensor nodes, sensor nodes can be mobile either due to improvements in technology, or environmental causes such as wind, water, or deployment of sensor nodes on moving objects. ZebraNet [21] is an example of mobile wireless sensor network application which is a habitat monitoring system. In ZebraNet, sensors are attached to zebras and collect information about their migration and behavior pattern. Some other applications are detailed in [22].

Wireless sensor networks are vulnerable to various malicious attacks. Due to the open nature of wireless communication channels, an attacker can easily eavesdrop the communication between sensor nodes which can lead to message tampering, or identity spoofing. In order to prevent such attacks, strong security algorithms should be implemented. These strong security algorithms require more resources such as computational power, or tamper-proof hardware. However, sensor nodes have limited resources for the sake of being low-cost devices, and a wireless network is composed of hundreds maybe thousands of sensor nodes. Hence, implementing such strong security algorithms seems infeasible without increasing the cost of sensor nodes, or without making a trade-off between security and performance. Another problem is that it is not easy to control sensor nodes once they are deployed in hostile environments such as military fields. Due to being mostly unattended, sensor nodes become open to physical attacks such as identity spoofing, node capture and compromise which may lead to various attacks including wormhole attack, Sybil attack, denial of service attacks. These malicious attacks, which are generally categorized as mote class / laptop class attacks, insider / outsider attacks, passive / active attacks, are well described in the literature [4].

Wormhole attack is an example of passive, outsider, laptop class attacks, where there are two or more malicious colluding nodes. An attacker tunnels messages received in one part of the network over a wormhole link and replays them in a different part of the network. Due to the low-latency tunneling over wormhole link, the attacker makes two distant nodes believe they are in the communication range of each other, and the network topology can be distorted as a result of these fake neighboring connections. Also, sensor nodes which are close to transceivers of the wormhole deplete their battery earlier as a result of heavy packet forwarding. Such an attack is a serious threat especially on routing protocols. The low-latency tunnel attracts network traffic on the wormhole link which can empower the attacker to perform traffic analysis, denial of service attacks; collect data to compromise cryptographic material; or just selectively drop data packets through controlling these routes using the wormhole link.

Several techniques have been proposed to detect wormhole attacks in wireless sensor networks which mostly focus on static networks. These solutions, some of which will be detailed later, are mainly based on detecting the maximum distance any message can travel, or the maximum time of travel of any message, discovering one-hop neighbors in a secure way, or monitoring the data traffic of neighbor nodes. Most of the proposed techniques require specialized hardware such as a GPS receiver or antennas, highly accurate time or location measurements, tight clock synchronization, or specialized trusted nodes, which seems infeasible for large scale wireless sensor networks because of its resource limitations and economic costs. Moreover, mobility of sensor nodes is not considered in these solutions.

1.1. Contribution of the Thesis

In this thesis, we propose a distributed wormhole detection scheme for mobile wireless sensor networks which is composed of two phases: (i) stabilization phase, and (ii) detection phase. In stabilization phase, two network features (i.e. network node density, standard deviation in network node density) are estimated via using local neighbor information along with preset parameters which are detailed in Section 3. Detection phase starts once stabilization phase ends. In this phase, the wormhole attack is detected via observing anomalies based on the estimated network features along with the neighboring information. Our scheme utilizes the mobility of the sensor nodes to estimate two above-mentioned network features in a local manner. Without a wormhole attack being performed, the difference between the number of neighbors of a node and its estimated network density does not exceed the standard deviation of its network density. However, under wormhole attack, this difference can be higher due to fake neighboring connections, especially when a node is close to the wormhole ends.

Our scheme achieves a detection rate up to 100% and very small false positive rate (at most 1.5%) when the parameters are chosen accordingly. Moreover, our solution requires neither additional hardware nor tight clock synchronization both of which are costly for sensor networks in terms of power consumption and economic costs.

1.2. Organization of the Thesis

The rest of the thesis is as follows. Section 2 gives general background information on wormhole attacks in wireless sensor networks and presents previous solutions in the literature. In Section 3, details of the proposed scheme are explained. Section 4 presents performance details including system assumptions and threat model, performance metrics, and simulation results. Finally, Section 5 concludes the thesis.

2. BACKGROUND ON WORMHOLE DETECTION IN WIRELESS SENSOR NETWORKS

In this section, background information about wormhole attacks and proposed solutions are presented. Section 2.1 explains the wormhole attacks as well as their effects on the network while Section 2.2 details the proposed solutions for wormhole attack detection.

2.1. Wormhole Attacks

Wormhole attack is an example of passive, outsider, laptop class attacks, where there are two or more malicious colluding nodes. An attacker tunnels messages received in one part of the network over a wormhole link (i.e. out-of band hidden channels such as a wired link, high power transmissions, packet encapsulation.) and replays them in a different part of the network. Figure 2.1 shows a typical wormhole attack scenario where node *X* and node *Y* are captured by an attacker and a wormhole is created via wired link. Each packet received at node *X* is sent to node *Y* over the wired link, and replayed in that part of the network. Due to the low-latency tunneling over wormhole link, nodes *a*, *b*, and *c* which are in the communication range of *X* believe that node *e* and *d* are their neighbors which is not the real case. Similarly, each packet received at node *Y* is sent to *X* over the wormhole link and replayed at that part of the network. By doing so, node *d* and *e* believe that they are neighbors with node *a, b,* and *c* which is not the real case. Network topology can be distorted as a result of fake neighboring connections introduced by the wormhole link.

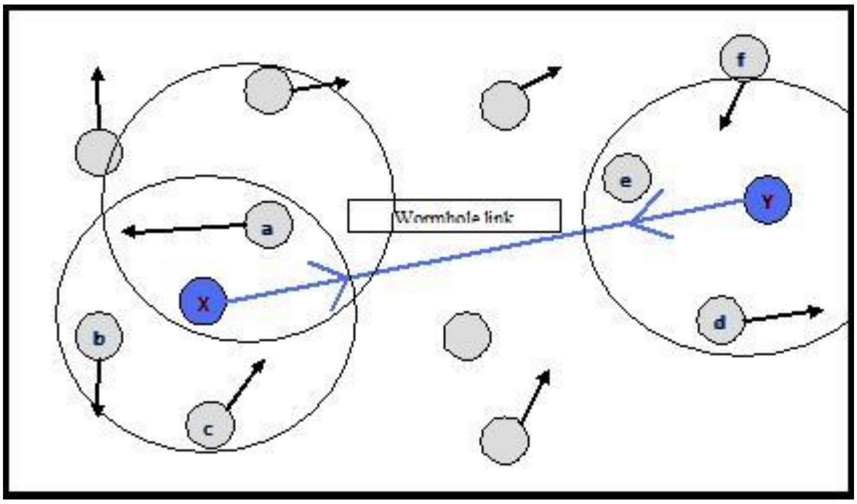

Figure 2.1: Wormhole attack scenario

Such an attack is a serious threat especially on routing protocols. The low-latency tunnel attracts network traffic on the wormhole link which can empower the attacker to perform traffic analysis, denial of service attacks; collect data to compromise cryptographic material; or just selectively drop data packets through controlling these routes using the wormhole link. In [3], simulations show that more than 50% of the data packets are attracted to fake neighboring connections and get discarded when there are more than two wormholes in the network. Moreover, an attacker can perform this attack without compromising any legitimate nodes, or knowing any cryptographic materials since the attacker neither creates new packets nor alters existing packets. Hence, wormhole attack cannot be prevented using only cryptographic measures.

2.2. Literature on Wormhole Detection

In [2], the concept of *packet leashes* are proposed to defend against wormhole attacks. The idea is to restrict the maximum transmission distance that a packet can travel through using either location information or tight time synchronization. *Temporal leash* guarantees that each packet has an upper bound on its life time. Hence, maximum travelling distance of the packet is also restricted. Each node appends a timestamp to each sent packet, and the network is assumed to be tightly synchronized. *Geographical leash* guarantees that the recipient of each packet is within a certain distance from the sender. Each node is assumed to know its exact location, and it appends this information along with sending time to each sent packet. The recipient nodes use both location and time information to verify whether a packet is sent over a wormhole link. Geographical leash requires loosely synchronized clocks. Both approaches need either location information and loosely synchronized clocks, or only tightly synchronized clocks. However, neither sensor node localization, nor network synchronization is not easy to achieve in wireless sensor networks.

In [3], a cooperative scheme is proposed to prevent wormhole attacks in wireless ad hoc networks where each node in the network is assumed to be equipped with directional antennas [12], [13]. A directional antenna can transmit/receive signals most effectively in a particular direction (or more directions as in Omnidirectional antennas). Therefore, each node can obtain the direction of incoming packets though using specific sectors of its directional antenna. Since a node knows from which direction it gets a packet, it can derive the relative orientation of the sender node with respect to its own location. In the scenario where there is no wormhole, when a node sends a packet in a given direction, its neighbors should get that packet from the opposite direction. If there is a wormhole in the network, the above rule may be broken by fake neighbors due to the location of the wormhole. Hence, the wormhole can be detected. However, wormhole may be located such that it does not break the above mentioned rule. To overcome this problem, two algorithms are presented [3] in which a node cooperates with its neighbors during detection period. Although the proposed approach is efficient in terms of energy consumption, the requirement of directional antennas is not practical in large scale wireless sensor networks.

SECTOR [5] is another proposed scheme for detection of wormhole attacks in wireless networks via enabling each node to securely discover its one-hop neighbors. To do so, the real physical distance between two nodes is calculated using an authenticated *distance bounding* protocol. Each node first sends a one-bit challenge request to the other node which will respond with a one-bit response instantly. After receiving the one-bit response, each node locally calculates the difference between sending the challenge and receiving the response, and estimates the distance to the other node. Hence, each node can determine whether the calculated distance is within the maximum possible communication range. Accurate measurement of local timing is an essential part of this method which is possible with current technology. However, special medium access control protocols are required as well as a specialized hardware for an instant challenge request-response mechanism.

In [6], two mechanisms are proposed to detect wormholes in wireless sensor networks. Neighbor number test (NNT) and all distances test (ADT) are both based on hypothesis testing and the results are probabilistic. NNT which is based on the distribution of neighboring-node-number detects the increase in the number of neighbors of the sensor nodes in order to detect bogus neighbors introduced by the wormhole. ADT detects the decrease of the lengths of the shortest paths between all pair of sensor nodes in order to detect shortcut links introduced by the wormhole. In both approaches, the sensor nodes send their neighbor lists to the base station and the base station runs the algorithm on the network graph which is reconstructed from the received neighborhood information. In other words, this is a centralized solution where the base station is assumed to have no resource limitations such as memory or computational power. However, this is not applicable in some wireless sensor network applications where the base station has limited resources.

In [7], a centralized solution, Multi Dimensional Scaling – Visualization of Wormhole (MDS-VOW), is presented in which wormhole is detected via visualizing the distortions due to the existence of wormhole link using computed maps. In this approach, each sensor node estimates the distance to its neighbors and sends this information to a central controller which reconstructs the layout of the sensors using a multi-dimensional scaling algorithm. When there is a wormhole in the network, it creates distortions in the layout which leads the way to detecting and locating the wormhole. However, a central controller without computation and memory limitations is required in this technique. Also, each sensor node needs to estimate the distance to its neighbors which implies the requirement for either a localization algorithm or a GPS receiver to get location estimate.

In [8] and [9], a wormhole detection mechanism is proposed for wireless sensor networks performing under multi-path routing which is based on statistical analysis of multi-path (SAM). In most of the multi-path routing protocols, the wormhole link attracts the network traffic due to its low latency transmission, and thus, certain routes are chosen more frequently than others. Therefore, it is possible to detect wormhole attack and identify the malicious nodes via analyzing the difference between two of most frequently used links among all obtained routes. However, the success of the method depends on the availability of enough routing information. Neither specialized hardware nor any changes to existing systems is required in this solution. Despite the fact that this is an efficient and accurate solution under multi-path routing protocols, it cannot perform well under uni-path routing protocols.

SeRLoc [10] is proposed as a localization scheme which is robust under wormhole attack via using location information. However, unlike the geographical leash approach [2], this approach requires only a small number of the nodes to be equipped with GPS receivers which are called *guards*. The guards broadcast their locations in their first-hop neighbors in an authentic way as well as protected against replay. Guards are also assumed to have larger radio range than other nodes (R), and they are placed $2 \times R$ far from each other. Therefore, each node can hear from only one guard, the distance to that guard cannot exceed R, and a node cannot receive same message twice from the same guard. Otherwise, it is probable that a wormhole attack is being performed in the network.

LiteWorp [11] is proposed to detect wormhole attacks in static networks. Each node is required to know its one-hop and two-hop neighbors once the network is deployed. Some of the nodes are chosen as *guards* which monitor neighboring nodes' data transmission. This approach does not require any additional hardware, and efficient in static wireless networks. However, it cannot perform well in mobile wireless sensor networks with this setup. In [14], MobiWorp is introduced for wormhole detection in mobile ad hoc networks. The basics of this protocol are similar to LiteWorp [11] with addition of a central certification authority (CA) for global tracking of node positions via verifying the truth of any location. In other words, MobiWorp enables nodes to securely discover their one-hop and two-hop neighbors. However, all nodes are assumed to be aware of their current and destination locations, and thus, either GPS or location discovery algorithms based on beacon nodes [15], [16], [17], [18] are required. Moreover, the network is assumed to be loosely synchronized, and the CA is not limited in terms of memory and computational power.

Most of these proposed solutions focus on static networks, and thus, mobility is not considered. Also, they either require additional hardware (e.g. directional antennas in [3], GPS in [2], [7], and [14], a specialized hardware for one-bit challenge request-response [5] protocol), or a central controller [6], and [7] which is assumed to have unlimited resources, or special nodes such as guards in [10], or tight network synchronization [3] which is hard to achieve in sensor networks due to resource limitations. We propose a distributed solution without requiring additional hardware or tight time synchronization or an unlimited central controller, or special nodes. Our solution is simply based on statistical metrics explaining network which are estimated via utilizing mobility of the sensor nodes.

3. THE PROPOSED DISTRIBUTED SCHEME FOR WORMHOLE ATTACK DETECTION IN MOBILE WIRELESS SENSOR NETWORKS

In this section, we propose a distributed wormhole detection protocol for mobile wireless sensor networks which detects anomaly in the network via taking the advantage of mobility based on the neighboring information. Our scheme uses the statistical metrics which are calculated locally using the neighboring information. Depending on the choice or system parameters, our scheme achieves a detection rate up to 100% and a very small false positive rate (at most 1.5%).

The rest of this section is as follows. The network assumptions and threat model is explained in Section 3.1. Our detection scheme is detailed in Section 3.2.

The notations which are used in this section are specified in Table 3.1.

Table 3.1: List of notations used in Section 3

A	Size of the network area (m^2)
N	Number of nodes in the network
R	Communication range (m)
$\vartheta_{\min}$	Minimum speed allowed (m/s)
$\vartheta_{\max}$	Maximum speed allowed (m/s)
i	Identity of a node
d_i^r	Local network density of node i at round r
σ_i^r	Standard deviation in d_i^r of node i at round r
ψ_i	The number of neighbors of node i
N_i	Set of neighbors of node i
T_{round}	Round threshold
T_{alarm}	Alarm threshold
T_{revoc}	The minimum number of nodes required to revoke a node
α	Weight for previous values of d_i^r and σ_i^r
$(1-\alpha)$	Weight for new values of d_i^r and σ_i^r
S	Number of rounds in stabilization phase
LocalSuspectsList$_i$	The list of locally suspected nodes that node i witnessed but has not broadcasted to the network as globally suspected yet.
GlobalSuspectsList$_i$	The list of globally suspected nodes that node i has which is more or less same for all nodes.

3.1. Network Assumptions and Threat Model

The network is assumed to be composed of mobile nodes which moves based on random way point model. In this mobility model, each node chooses a random destination and moves towards it with a speed uniformly distributed in $[\vartheta_{\min}, \vartheta_{\max}]$. Each node stops for a preset duration when it reaches the destination. Moreover, the network is homogeneous which implies that all sensor nodes in the network have same communication range as well as the same physical properties. The sensor nodes are deployed randomly using uniform distribution in the sensing area. None of the nodes know their location information, or have GPS. The deployment area is much larger than the communication range of the nodes. More importantly, a node can obtain the neighbor count information of its neighbors as well as its own neighboring information via a secure neighbor discovery protocol in terms of cryptographic measures such as authenticity, integrity, and confidentiality. Secure neighbor discovery is out of the scope of the thesis. There are proposed solutions for neighbor discovery, [23], [24], [25], [26], addressing node mobility as well as energy efficiency in the literature. We assume that appropriate cryptographic algorithms and key infrastructures considering resource limitations in sensor network are used. Necessary link level security requirements (i.e. confidentiality, authentication, and integrity) are assumed to be fulfilled by the lower layers. Hence, the attacker cannot alter existing data packets and messages or fabricate new ones.

Due to its nature and being an outsider attack, a wormhole attack can be performed without compromising cryptographic materials such as encryption key. It is sufficient for an attacker to capture two legitimate nodes and create a low-latency tunnel between them. In our proposal, we assume that the wormhole link is bidirectional. In other word, both ends of wormhole link overhear the packets; tunnel these packets to other node via this low-latency tunnel so that the receiving node can replay these packets at that end of the wormhole. The attacker may drop the packets selectively in a random way. However, by doing so, the wormhole link becomes less attractive and this is not a desired situation for the attacker. Thus, we assume that the attacker does not drop any packets.

3.2. The Proposed Approach

In this section, the details of the proposed scheme are explained along with the motivation behind the approach. Section 3.2.1 gives the motivation behind this approach. The general overview of the proposed scheme is explained in Section 3.2.2. In Section 3.2.3 the steps and details of the stabilization phase are explained. Finally, in Section 3.2.4, detection phase is detailed.

3.2.1. Motivation

There are several approaches for wormhole detection in wireless sensor networks some of which are detailed in Section 2. However, majority of these proposals focus on static networks, and thus, mobility is not considered. Also, most of these approaches require additional hardware (e.g. directional antennas in [3], GPS in [2], [7], and [14], a specialized hardware for one-bit challenge request-response [5] protocol), or a central controller [6], and [7] which is unlimited in resources, or special nodes such as guards in [10], or tight network synchronization [3]. Moreover, the limitations of sensor nodes and base stations are not considered in all solutions. Our aim in this study is to develop a distributed wormhole detection protocol for mobile sensor networks without requiring any additional hardware via utilizing mobility of the sensor nodes in the network.

3.2.2. Overview of the Protocol

We propose a distributed wormhole detection scheme based on the statistical information derived from neighboring information. Our scheme aims to utilize the mobility feature of the sensor nodes to examine the environment and network properties, and derive new features which help understanding the network better. It includes two main phases: (i) stabilization, and (ii) detection phases.

Stabilization phase is for sensor nodes to collect information from the network using neighboring information to estimate the node density of the network locally, d_i^r for node i at r^{th} round, and to compute the standard deviation of the change in the estimated node density, σ_i^r. This phase runs once right after the uniform random deployment of the sensor nodes. We assume that there is no wormhole attack being performed during the stabilization phase.

In detection phase, based on the pre-computed statistical values, the detection mechanism is activated to check for anomalies in the network, and detected nodes are revoked from the network.

Workflow of these phases is shown in Figure 3.2.2.1.

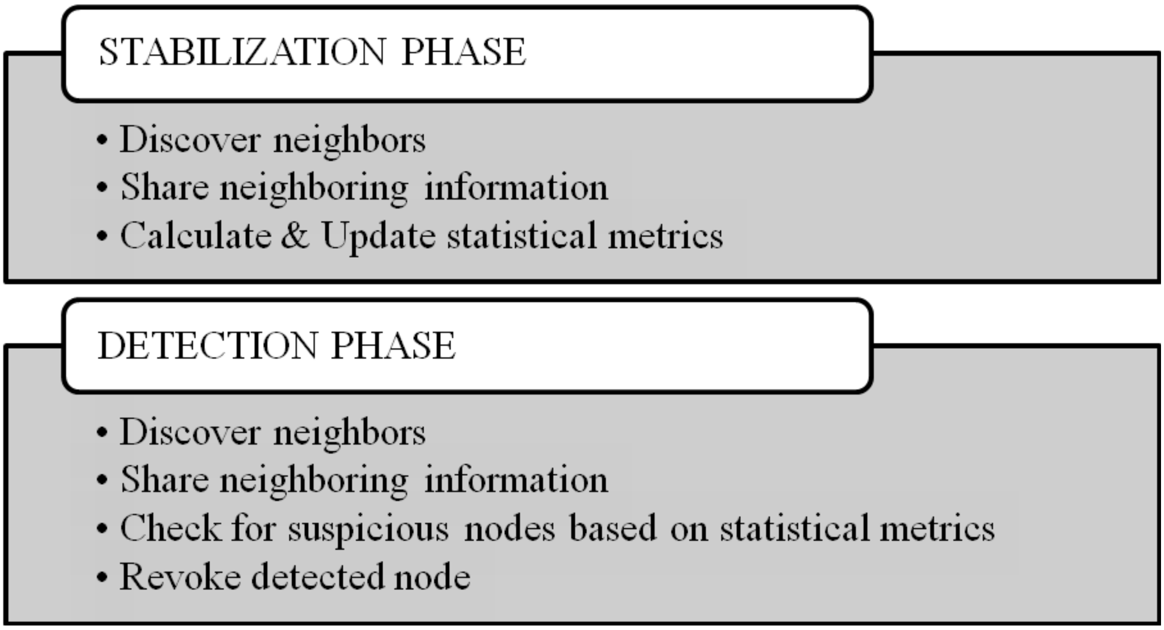
STABILIZATION PHASE
• Discover neighbors
• Share neighboring information
• Calculate & Update statistical metrics
DETECTION PHASE
• Discover neighbors
• Share neighboring information
• Check for suspicious nodes based on statistical metrics
• Revoke detected node

Figure 3.1: Workflow of the proposed scheme

3.2.3. STABILIZATION PHASE

Stabilization phase starts right after the uniform random deployment of N sensor nodes, and runs S rounds. In a round, each node discovers their neighbors securely, broadcasts its neighbor count, and locally computes statistical features of the network (i.e. d_i^r and σ_i^r) after receiving all neighbor counts of its neighbors.

3.2.3.1. Discover Neighbors

As mentioned in Section 3.1, neighbor discovery is not in the scope of the thesis. We assume that a secure neighbor discovery algorithm is used. There are proposed solutions, [23], [24], [25], [26], to discover one-hop neighbors in a secure way considering mobility of the nodes besides energy efficiency.

3.2.3.2. Share Neighboring Information

When a node learns its neighbors, it broadcasts an information packet including its own identity, i, and the number of its neighbors, ψ_i. This information is critical in the estimation of the network features (d_i^r and σ_i^r).

3.2.3.3. Calculate & Update Statistical Metrics

After all nodes share the number of their neighbors, each node i has the following information: its own neighbors, N_i, the number of its own neighbor number, ψ_i, and neighbor count information of its neighbors, $\psi_j \forall j \in N_i$. Then, node i computes the network density, d_i^r, and standard deviation in d_i^r, σ_i^r, in a local way using equations:

$$d_i^0 = 0 \tag{1}$$

$$d_i^r = \frac{\psi_i + \sum_{j \in N_i} \psi_j}{\psi_i + 1} \times (1-\alpha) + d_i^{r-1} \times \alpha \tag{2}$$

$$\sigma_i^0 = 0 \tag{3}$$

$$\sigma_i^r = \sqrt{\frac{1}{\psi_i + 1}\left[(\sum_{j \in N_i} (\psi_j - d_i^{r-1})^2) + (\psi_i - d_i^{r-1})^2 \right]} \times (1-\alpha) + \sigma_i^{r-1} \times \alpha \tag{4}$$

We use exponential averaging, which we are inspired by its usage in TCP round trip time estimation, to give more importance to the latest data retrieved from neighbors without losing the previous calculated values. α and $(1-\alpha)$ are the weights which are used to estimate standard deviation and local network density of a node. As shown in Eq.1 and Eq.3, initial values for both node density and standard deviation are set to 0. At each round, each node estimates a candidate density value which is calculated by averaging the neighbor counts received from neighbors along with its own neighbor count. After that, the node updates its density via using the exponential average of the previous value and the new estimated value. The procedure is same for the calculation of standard deviation in the node density. The only difference here is that it uses basic standard deviation calculation via utilizing the neighbor count information received from neighbors.

In the stabilization phase, apart from neighbor discovery messages, the only message overhead in the network is caused due to sharing neighboring information explained in Section 3.2.3.2.

3.2.4. DETECTION PHASE

In detection phase, pre-computed network features (i.e. d_i^r and σ_i^r) along with round threshold, T_{round}, alarm threshold, T_{alarm}, and the number of nodes to revoke a node, T_{revoc}, are used to detect the anomaly created by the wormhole link. Detection phase runs as long as the lifetime of the sensor node. A round in detection phase is composed of neighbor discovery, sharing the number of neighbors, testing detection criteria along with broadcasting specific messages when necessary, and finally revocation of detected nodes.

3.2.4.1. Discover Neighbors

Discovering neighbors is challenging in mobile wireless sensor networks. There are proposed solutions in [23], [24], [25], and [26] some of which focus on energy-efficiency, or neighbor list management, or mobility. As mentioned in Section 3.1 while explaining our assumptions, we assume that nodes are capable of defining their neighbors.

3.2.4.2. Share Neighboring Information

Sharing the neighborhood information is a crucial part of detection phase. Each node requires its neighbors sending their neighbor counts to detect a suspicious behavior. Each node broadcasts its identity along with the number of its neighbors as explained above, in Section 3.2.3.2.

3.2.4.3. Check for Suspicious Nodes based on Statistical Metrics

After obtaining the neighborhood information, each node i has the network density, d_i^S, and standard deviation in d_i^S, σ_i^S, and the neighboring information $\psi_j \forall j \in N_i$. Node *i* detects possible anomaly using the check in Figure 3.2 which is the pseudo-code for local detection. It first checks whether the number of its own neighbors exceeds its locally-estimated density more than its locally-estimated standard deviation. If the difference exceeds the locally-estimated standard deviation, i accuses its neighbors and adds them in its list for tracking suspicious nodes. Otherwise, node i checks its neighbors one by one with the same method to detect a suspicious behavior and updates its list accordingly. If the alarm counter for a locally suspected node *j* exceeds the alarm threshold, then node *i* broadcasts a message deeming *j* is a globally suspected node. If any node in the list of locally suspected nodes does not show an anomaly during the round threshold, then node *i* deletes that node from its list.

```
if $((\psi_i - d_i^r) \succ \sigma_i^r)$
    $\forall j \in N_i$
    if $j \in$ *LocalSuspectsList* of $i$
        update information for $j$
        if alarm number for $j$ exceeds $T_{alarm}$
            broadcast $[$*Global Suspect Message, i, j*$]$ where $j$ is the global suspect
        else add $j$ to *LocalSuspectList* of $i$
        $\forall k \in$ *LocalSuspectsList* of $i$
            remove $k$ if no alarm increased for $k$ for $T_{round}$
else
    $\forall j \in N_i$
     if $((\psi_j - d_i^r) \succ \sigma_i^r)$
         if $j \in$ *LocalSuspectsList* of $i$
             update information for $j$
             if alarm number for $j$ exceeds $T_{alarm}$
                 broadcast $[$*Global Suspect Message, i, j*$]$ $j$ is the global suspect
         else add $j$ to *LocalSuspectsList* of $i$
         $\forall k \in$ *LocalSuspectsList* of $i$
             remove $k$ if no alarm increased for $k$ for $T_{round}$
```

Figure 3.2: Pseudo-code of local detection

When a node i receives a *Global Suspect Message* saying node j is a potential malicious node, it runs the following check in Figure 3.3 which is the pseudo-code for global detection. To revoke node *j*, the number of nodes deeming node *j* as suspected must exceed the revocation threshold which is basically a preset percentage of the total number of nodes.

```
if $j \in GlobalSuspectsList$ of $i$
    update the number of nodes add $j$ as global suspect
    if the number of nodes adding $j$ as global suspect exceeds $T_{revoc}$
        broadcast $[Revoke\ Message,\ i,\ j]$ where $j$ is the one to be revoked
        remove the entry for $j$ from $GlobalSuspectsList$
        add $j$ to $RevokedList$
else add $j$ to $GlobalSuspectsList$ of $i$
```

Figure 3.3: Pseudo-code of global detection

3.2.4.4. Revoke Detected Node

A globally suspected node can be revoked from network through node self-destruction mechanisms proposed in [27] and [28]. When a node i receives a *Revoke Message* saying node j is a malicious node, it sends a message to the base station for revocation of *j* and updates its *RevokedList* accordingly.

4. PERFORMANCE EVALUATIONS

We analyzed the performance of our scheme via simulations. Section 4.1 contains detailed explanation of system parameters. Simulation setup is given in Section 4.2. Section 4.3 shows the simulation results including performance metrics.

4.1. System Parameters & Performance Metrics

System Parameters:

- Round threshold, T_{round}, is the maximum number of rounds in which a node *a* needs to witness an anomaly about a node *b* to keep node *b* in its local suspected nodes list.
- Alarm threshold, T_{alarm}, is the minimum number of alarm to broadcast a node as globally *suspected*.
- Revocation threshold, T_{revoc}, is the number of nodes required to revoke a node.
- α and $(1-\alpha)$ are the weights used for estimating the network features defined in the proposed scheme. We simulated different values of α varying between $[0..1]$ interval. The results show that the more optimal and stable value for α is 0.5. Therefore, we choose α as 0.5 in our simulations.

Performance Metrics:

Detection rate and false positive rate are our main metrics while evaluating the success of the simulations. Detection rate is the ratio of the number of simulation runs where the wormhole is detected successfully, call D#, over total number of simulation runs, call S#. It is computed as follows:

$$\text{Detection rate} = \frac{\text{D\#}}{\text{S\#}} \tag{5}$$

False positive rate per simulation run is computed as the ratio of falsely detected nodes, call F#, over total node number, *N*. False positive rate is the average of this ratio of all simulation runs. It is computed as follows:

$$\text{False positive rate} = \frac{\sum_{1}^{\text{S\#}} (\frac{\text{F\#}}{\text{N}})}{\text{S\#}} \tag{6}$$

4.2. Simulation Setup

Simulation code is written using C# language in Windows 32-bit operating system. We perform 20 simulations for each parameter value; the results presented in the graphs are average of 20 simulations. In our simulations, $N = 200$ nodes are distributed over a field of $A = 100m \times 100m$. We use random way point mobility model in which each node chooses a random destination; moves towards it with a uniformly distributed random speed in the range of $[5m/s, 15m/s]$; and stops for a preset duration when it reaches the destination. Nodes have a communication range of $15m$. Alarm threshold, T_{alarm}, varies between $[10...90]$ with 5 units increments. We simulated three values ($T_{revoc} = 0.05 \times N$, $T_{revoc} = 0.10 \times N$, and $T_{revoc} = 0.15 \times N$) for the percentage of nodes that are required to revoke a node. We assume that some of the nodes in the network, which is selected as 5%

of all nodes, are static all the time. Also, we assume that the wormhole attack is not performed right after the deployment of the sensor nodes during stabilization phase. The proposed scheme is composed of two phases: (i) stabilization, (ii) detection. Stabilization phase runs once and lasts $S = 1000$ rounds. Detection phase runs during the lifetime of a sensor node due to the possibility of wormhole attack being performed at any time. However, we limit this value to 2000 rounds in our simulations. In each round, a node discovers its neighbors, shares its own neighbor count with its neighboring nodes, and runs the wormhole detection algorithm locally. Secure neighbor discovery is a challenging issue in mobile wireless networks. There are proposed solutions, some of which are [23], [24], [25], and [26], in the literature to overcome this difficulty considering the mobility of nodes as well as energy-efficiency. We assume that each node can discover its neighbors securely.

4.3. Simulation Results

The organization of this section is as follows: Section 4.3.1 explains the details of the performance metrics which are: (i) detection rate, (ii) false positive rate, (iii) detection round, and (iv) memory requirements. Section 4.3.1 analyzes the detection rates; Section 4.3.2 analyzes the false positive rates; Section 4.3.3 discusses the average detection duration in terms of round; Section 4.3.4 shows the average memory requirement in the simulations in a comparative way; and finally, Section 4.3.5 analyzes the effect of node density and size of deployment area on detection and false positive rates.

4.3.1. Detection Rates

Figure 4.1 shows the detection rate with varying node threshold (T_{revoc}) and alarm threshold (T_{alarm}) values. Details of these values are given in Section 4.2 which explains the simulation setup. Round threshold (T_{round}) is set to 10. When a node, *a*, witnesses a suspicious behavior of another node, *b*, *a* adds *b* in its list for locally suspicious nodes. If *a* does not detect any anomaly about *b* for 10 rounds, then *a* deletes *b* from its list. Increasing T_{revoc} means that more nodes are needed to claim a node as malicious and revoke that node. Hence, detection rate increases when T_{revoc} decreases as expected. If T_{alarm} is increased, a node needs to witness more suspicious behaviors of a node to broadcast it as globally *suspected*. As a result, detection rate decreases with the increase in T_{alarm}.

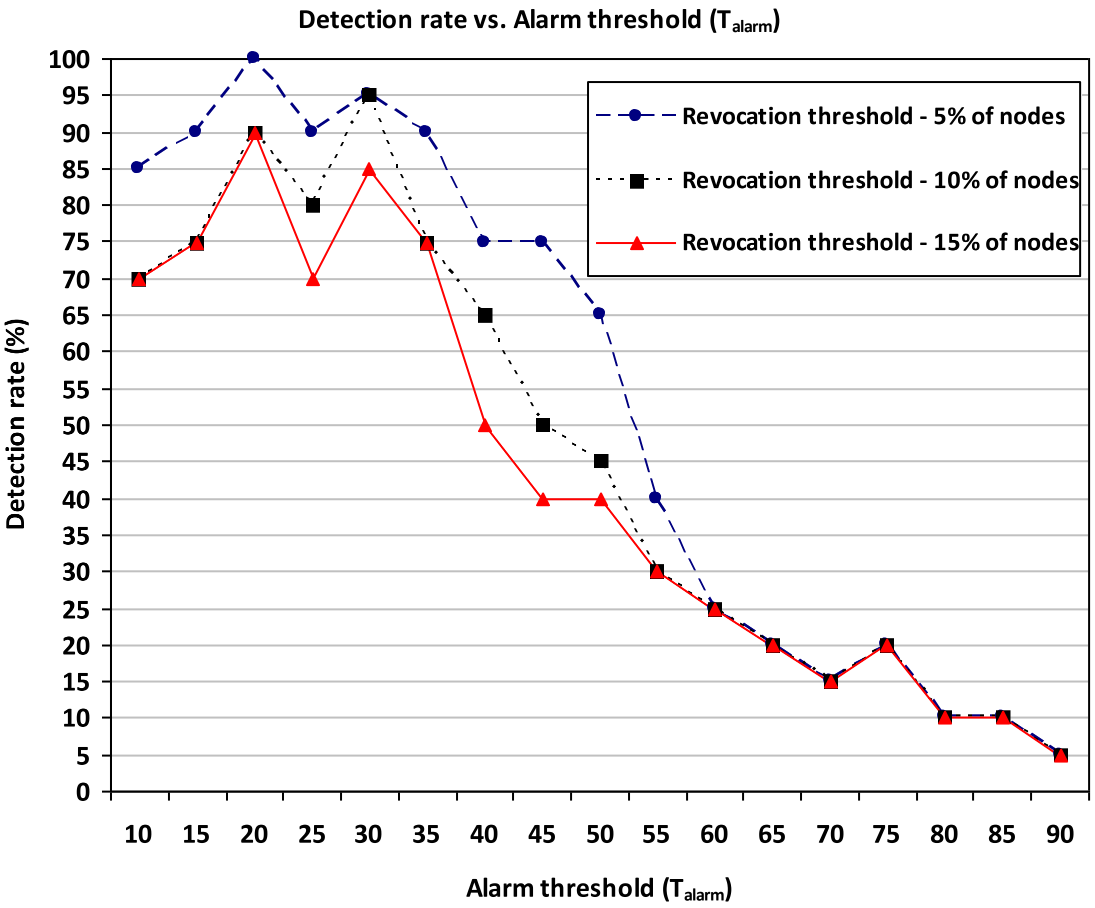

Figure 4.1: Detection rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 10$. Wormhole ends are chosen randomly.

Figure 4.2 shows the impact of round threshold (T_{round}) on the detection rate under varying node threshold (T_{revoc}) and alarm threshold (T_{alarm}) values. Round threshold (T_{round}) is set to 20 which is the only difference from the results shown in Figure 4.1. When a node, *a*, witnesses a suspicious behavior of another node, *b*, *a* adds *b* in its list for locally suspicious nodes. If *a* does not detect any anomaly about *b* for T_{round} rounds, then *a* deletes *b* from its list. Exceeding T_{alarm} becomes more difficult as T_{round} increases unless a node continuously shows suspicious behaviors which imply it is a potential malicious node. Comparing to the results presented in Figure 4.1, the detection rate is more or less higher in

Figure 4.2. Also, detection rate decreases more gradually when T_{revoc} is set 20 as compared to Figure 4.1.

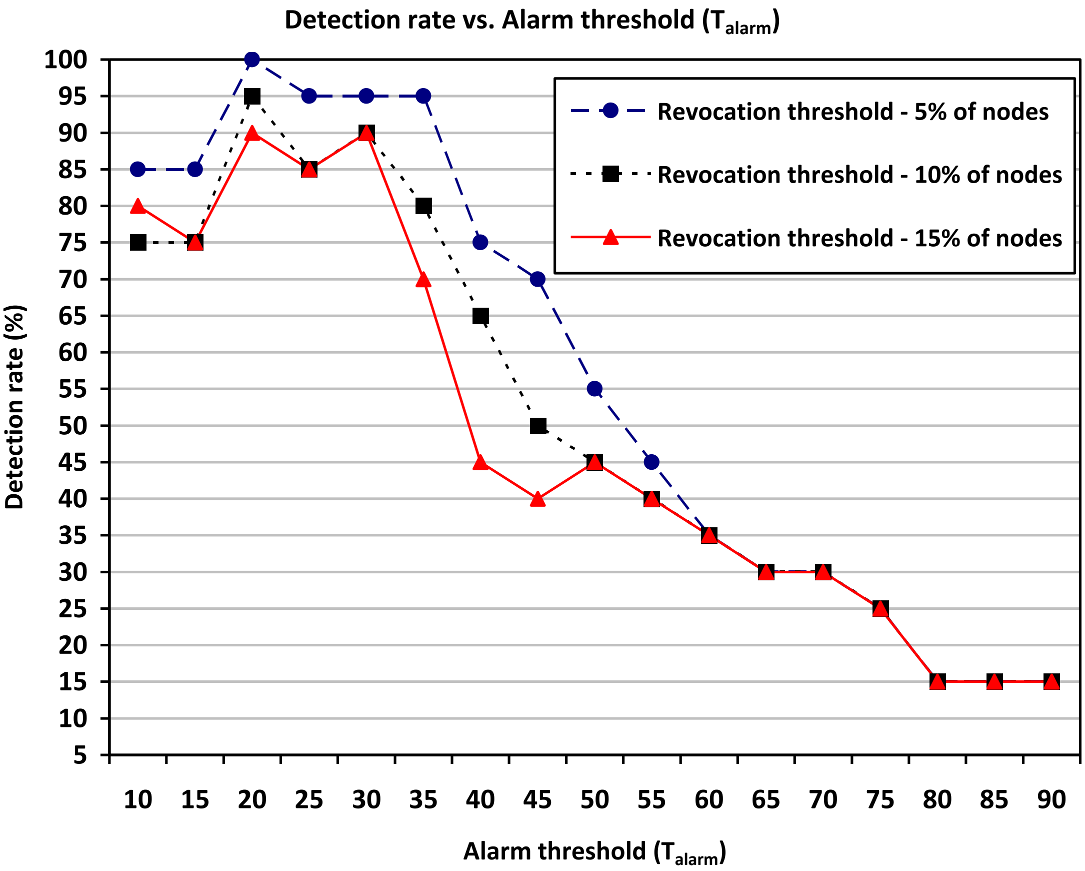

Figure 4.2: Detection rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 20$. Wormhole ends are chosen randomly.

In Figure 4.3, the effects of wormhole location on detection rate are presented under varying node threshold (T_{revoc}) and alarm threshold (T_{alarm}) values. Round threshold (T_{round}) is set to 10. Location of the wormhole is the only difference from the results presented in Figure 4.1. Locating wormhole at $(25,25)$ and $(75,75)$, we make sure that the wormhole is not on the borders of the deployment area, and thus, it affects more nodes in the network. The probability to detect wormhole increases due to the fake neighboring connections which are introduced by the wormhole link. This increase in fake neighbors creates more anomalies in terms of the deviation from the pre-computed network density. Detection rate is higher as compared to the results presented in Figure 4.1. A detection rate of 100% is achieved up to $T_{alarm} = 40$ when T_{revoc} is 10 which is 5% of the nodes in the network. However, the decrease in detection rate after $T_{alarm} = 40$ sharper compared to Figure 4.1.

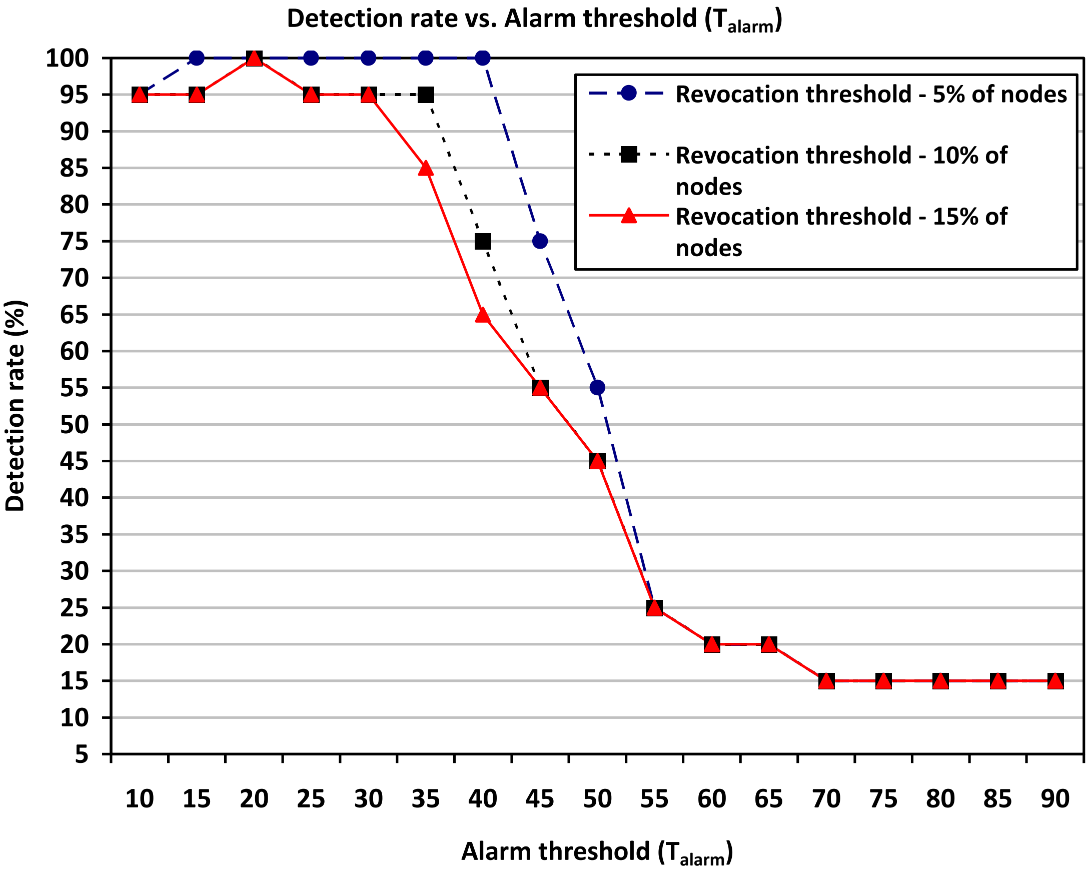

Figure 4.3: Detection rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 10$. Wormhole ends are at $(25,25)$ and $(75,75)$.

The impact of round threshold (T_{round}) is presented in Figure 4.4 under varying node threshold (T_{revoc}) and alarm threshold (T_{alarm}) values. T_{round} is set to 20 which is the only difference from the results shown in Figure 4.3. Increase in T_{round} smoothes the sharp decrease shown in Figure 4.3. In other words, detection rates decrease more gradually T_{alarm} increases. Moreover, the detection rates at high T_{alarm} increases as T_{round} increases from 10 to 20. Its impact is more obvious when T_{revoc} is 10. Also, the detection rate is over 50% up to $T_{alarm} = 70$.

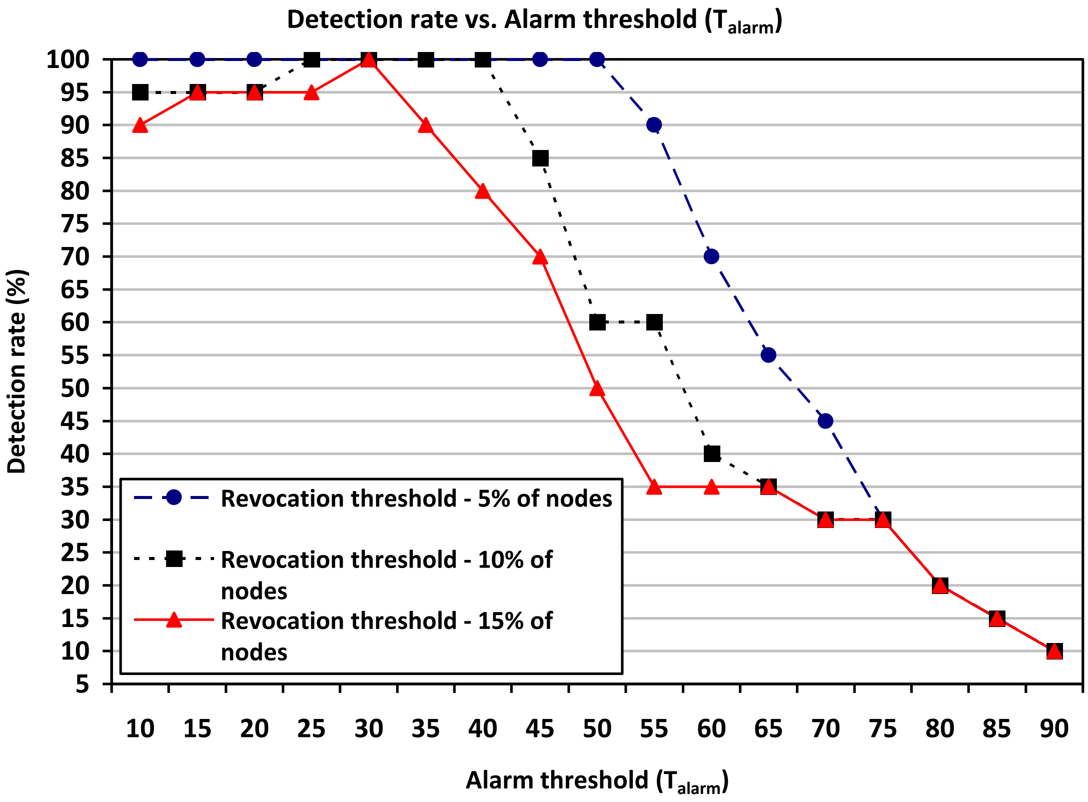

Figure 4.4: Detection rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 20$. Wormhole ends are at $(25,25)$ and $(75,75)$.

4.3.2. False Positive Rates

Figure 4.5 shows the false positive rate with different node threshold (T_{revoc}) and alarm threshold (T_{alarm}) values which are explained in detail in Section 4.2. Round threshold (T_{round}) is set to 10. False positive rate varies between 0.004 and 0.014 with the given values. Increasing T_{alarm} implies that a node needs to witness more anomalies of a node to broadcast it as globally *suspected*. Hence, we can say that the number of falsely detected nodes decreases as T_{alarm} increases. The simulation results verify that observation. Increasing T_{alarm} decreases the false positive rate up to a point; and false positive rate does not change much after a high enough T_{alarm} value. T_{revoc} is also inversely proportional to the false positive rate since high T_{revoc} means more nodes are required to agree on revoking a node. Hence, if we increase T_{revoc}, the false positive rate decreases.

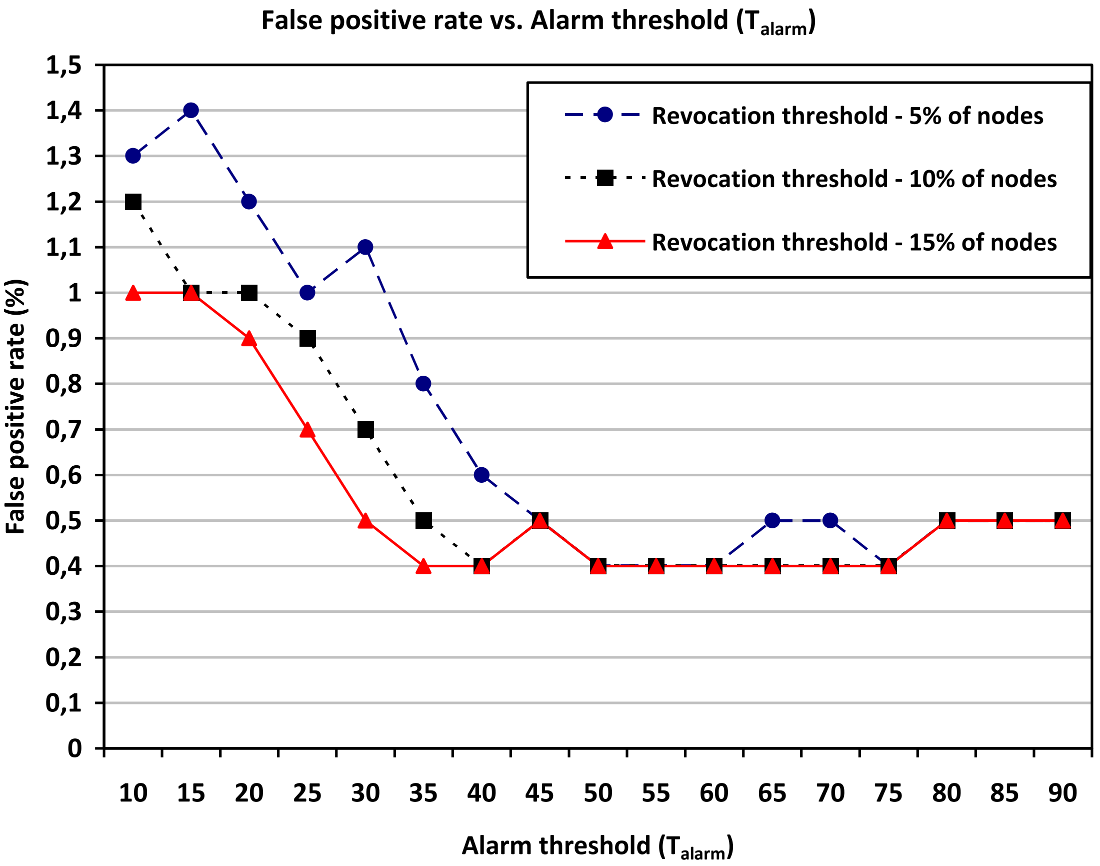

Figure 4.5: False positive rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 10$. Wormhole ends are chosen randomly.

The impact of round threshold (T_{round}) on the false positive rate under different node threshold (T_{revoc}) and alarm threshold (T_{alarm}) values is presented in Figure 4.6. The only difference from simulations shown in Figure 4.5 is the choice of round threshold (T_{round}) which is 20 in this case. Increasing T_{round} makes it more difficult to exceed T_{alarm} unless a node continuously shows suspicious behaviors. Depending on this observation, one can say that increase in T_{round} decreases the false positive rates. However, the simulation results do not verify this implication. This may be because of the low increase in T_{round}, or the effect of detecting wormhole. In order to verify it for sure, higher values for T_{round} should be analyzed.

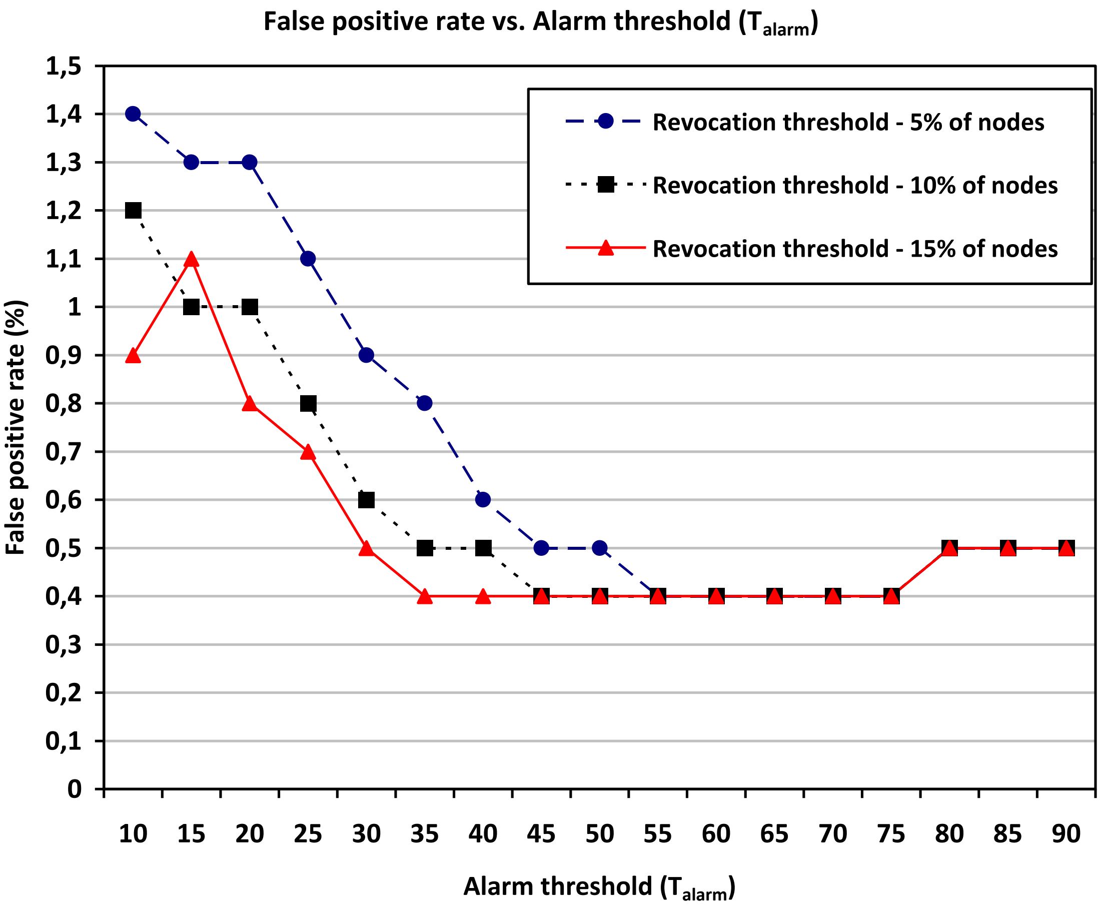

Figure 4.6: False positive rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 20$. Wormhole ends are chosen randomly.

In Figure 4.7, the impact of location of wormhole on the false positive rates under various node threshold (T_{revoc}) and alarm threshold (T_{alarm}) values. The value for round threshold (T_{round}) is 10. Only difference from the simulations presented in Figure 4.5 is the location of wormhole. We locate the wormhole ends at $(25,25)$ and $(75,75)$ which means that the wormhole ends are not on the borders of the deployment area. This implies that more nodes are affected by the wormhole link. Due to the fake neighboring connections introduced by the wormhole link, the probability of detecting wormhole becomes higher. In other words, when a node is under the effect of wormhole, it witnesses more suspicious behaviors which lead to detection of wormhole sooner. By intuition, one can say that affecting more nodes may result in the increase of the false positive rate. However, the impact of detecting wormhole earlier decreases the false positive rate which can be seen more obviously when T_{revoc} is lower. The results shown in Figure 4.5, at $T_{alarm} = 35$ and when T_{revoc} is 10 and $T_{alarm} = 35$, the value of false positive rate is 0.08% in Figure 4.5 while it is 0.05% Figure 4.7.

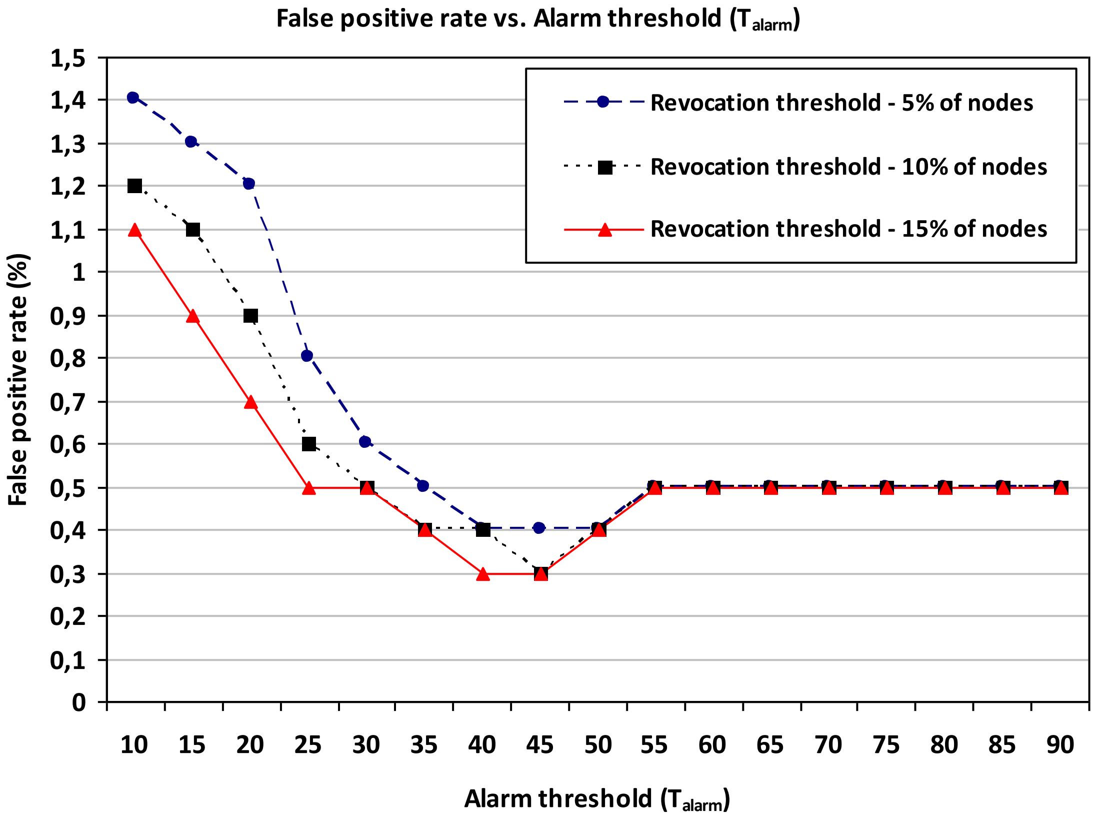

Figure 4.7: False positive rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 10$. Wormhole ends are at $(25,25)$ and $(75,75)$.

Figure 4.8 shows the effects of round threshold (T_{round}) with different node threshold (T_{revoc}) and alarm threshold (T_{alarm}) values. T_{round} is chosen as 20 which is different from the results shown in Figure 4.7. There is a slight increase in false positive rates depending on the change in T_{round}. However, as T_{alarm} increases, especially after 50, false positive rate becomes lower as compared to the simulation results shown in Figure 4.7 which may be a result of the increase in detection rates (over 50% up to $T_{alarm} = 70$) presented in Figure 4.4.

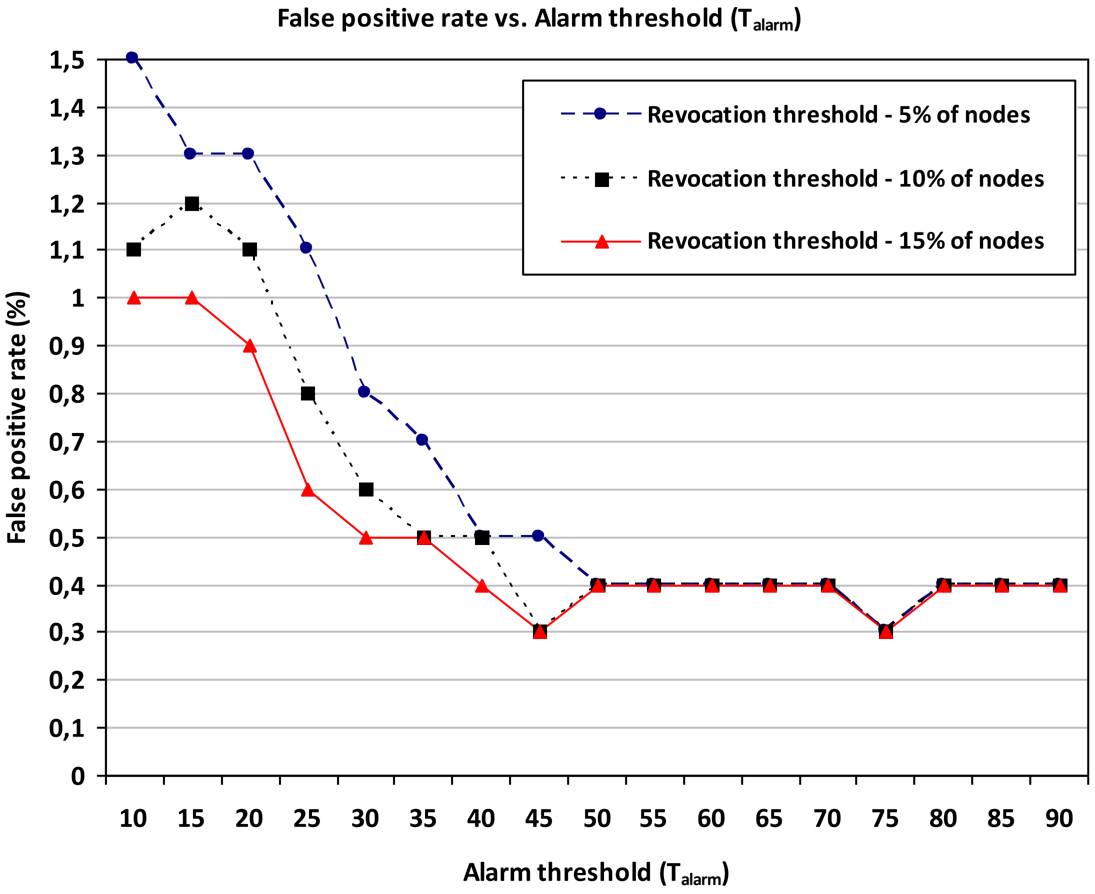

Figure 4.8: False positive rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 20$. Wormhole ends are at $(25,25)$ and $(75,75)$.

4.3.3. Detection Round

Figure 4.9 presents the average detection rounds with varying node threshold (T_{revoc}) and alarm threshold (T_{alarm}). Details about these values are explained in detail in Section 4.2 which gives simulation setup. Round threshold (T_{round}) is chosen as 10. As mentioned in Section 4.2, stabilization phase runs 1000 rounds and detection phase starts right after stabilization phase ends. Hence, if detection round is shown as 1200, it means that the wormhole is detected at 200th round. High T_{alarm} values indicate that to broadcast a node *n* as globally *suspected*, node *m* needs to witness more suspicious behaviors of node *n*. Hence, if we increase T_{alarm}, detection round also increases which is an expected result. Increase in T_{revoc} results in increase in detection round due to the requirement of more nodes to agree on revoking a node.

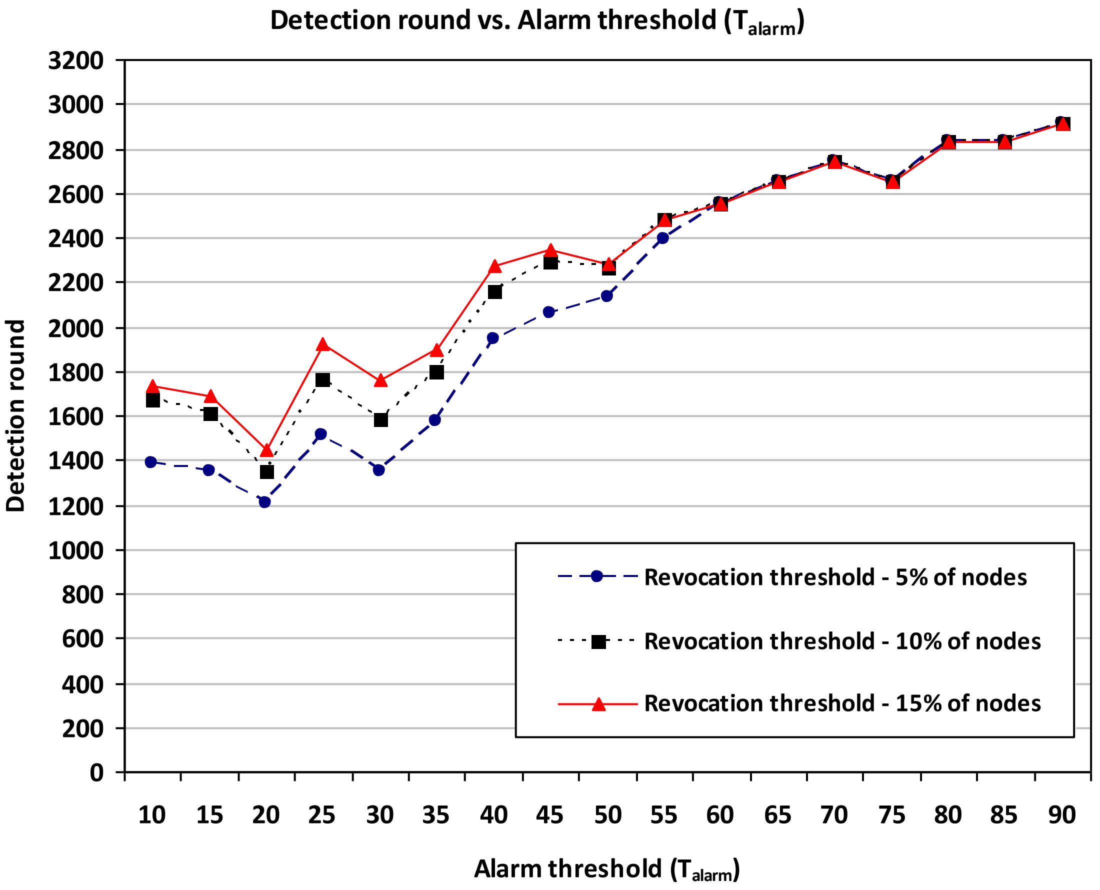

Figure 4.9: Detection round vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 10$. Wormhole ends are chosen randomly.

In Figure 4.10, the effect of round threshold (T_{round}) on detection round is presented under different node threshold (T_{revoc}) and alarm threshold (T_{alarm}) values. T_{round} is set to 20 which differs from the case shown in Figure 4.9. Exceeding T_{alarm} becomes more difficult when T_{round} increases unless a node continuously shows anomalies. By intuition, one can say that this decreases the number of false positives. Hence, the network does not loose nodes which can be helpful in detection of wormhole. This may affect the detection round. However, depending on the simulation results, we cannot say detection round changes much. The results are more or less same as compared to the case presented in Figure 4.9.

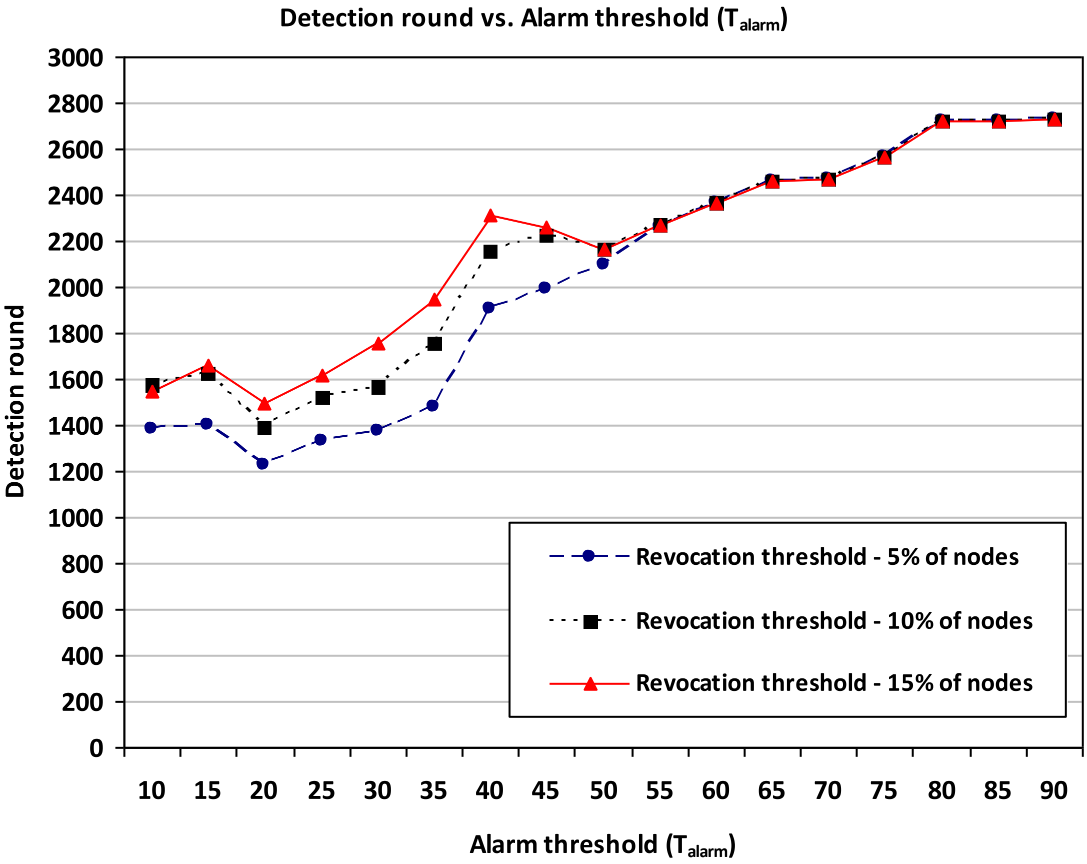

Figure 4.10: Detection round vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 20$. Wormhole ends are chosen randomly.

The impact of location of wormhole on the detection round is analyzed in the simulation presented in Figure 4.11. Location of the wormhole is the only difference from the case shown in Figure 4.9. By locating wormhole ends at $(25,25)$ and $(75,75)$, we guarantee that wormhole ends are not on the borders of the deployment area, and thus, wormhole affects more nodes in the network. Wormhole becomes more detectable due to the increase in neighbors caused by the wormhole link. Simulation results show that detection rounds are lower in the results shown in Figure 4.11 as compared to the case presented in Figure 4.9.

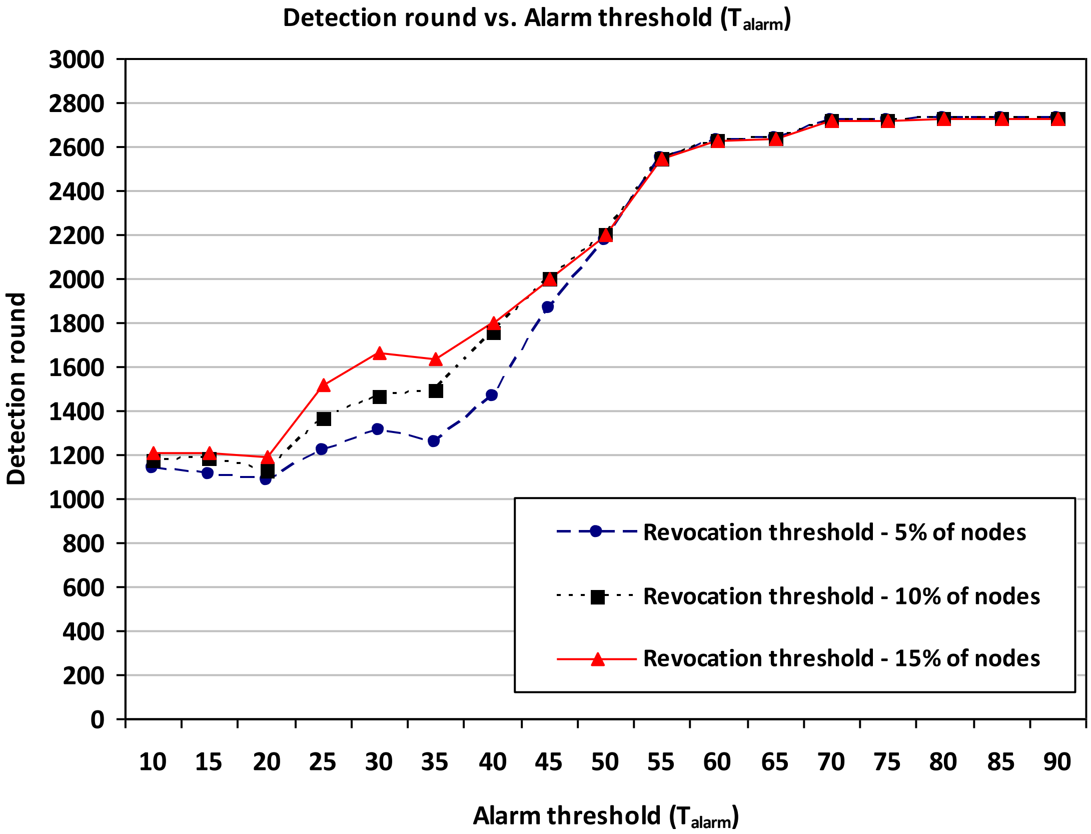

Figure 4.11: Detection round vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 10$. Wormhole ends are at $(25,25)$ and $(75,75)$.

In Figure 4.12, the results of increasing round threshold (T_{round}) are analyzed under different node threshold (T_{revoc}) and alarm threshold (T_{alarm}) values. T_{round} is set to 20 which is the only difference from the case presented in 4.11. There is a slight decrease in detection rounds due to the increase in T_{round}. The effect of this change can be seen for the case where T_{revoc} is 10. The sharp increase in detection round in Figure 4.11 is smoothed in Figure 4.12. One can observe that increasing T_{round} when T_{revoc} is low enables detection of wormhole sooner even at high T_{alarm} values. For instance, when T_{alarm} is 50 and T_{revoc} is 10, wormhole is detected at 2200th round in Figure 4.11 while it is detected less than 1800th round in Figure 4.12.

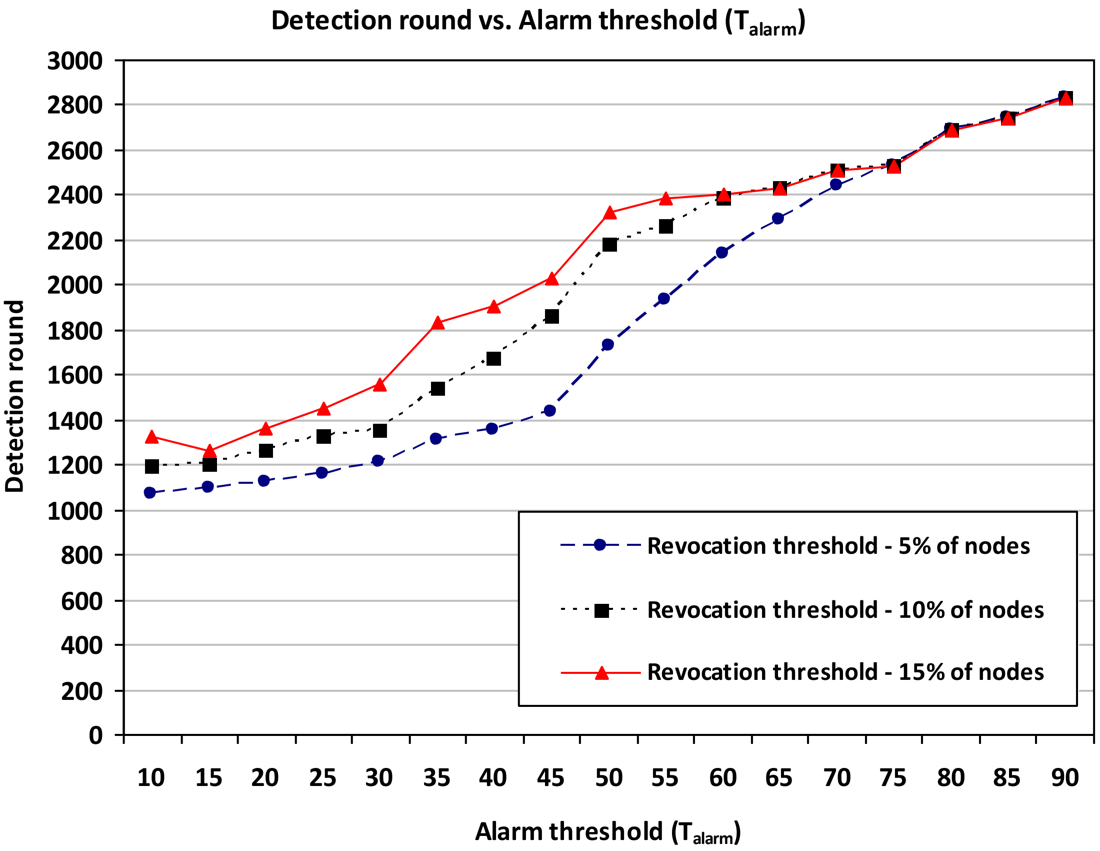

Figure 4.12: Detection round vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 20$. Wormhole ends are at $(25,25)$ and $(75,75)$.

4.3.4. Memory Requirements

LocalSuspectsList is the list which keeps locally *suspected* nodes before broadcasting them to the network as globally *suspected*. Each entry in *LocalSuspectsList* contains the identity of the *suspected* node (2 bytes), an alarm counter (1 byte) for it, and last round (2 bytes) in which an anomaly detected about it. So, 5 bytes are required for each entry in the *LocalSuspectsList*. Hence, the memory requirement for *LocalSuspectsList* is calculated via multiplication of list size by 5 bytes.

GlobalSuspectsList is for keeping globally *suspected* nodes, and it is more or less same at all nodes. Each entry in *GlobalSuspectsList* contains the identity of the global suspect (2 bytes), and the identities of nodes that broadcasted it as globally *suspected*. In order to cover the worst case, we assume that all *suspected* nodes in the *GlobalSuspectsList* are broadcasted by T_{revoc} many nodes. Hence, the memory required for each entry in the *GlobalSuspectsList* is calculated via the following formula:

$$(T_{revoc} \times 2) + 2 \tag{7}$$

Hence, the required memory for *GlobalSuspectsList* is obtained via multiplication of list size by (7).

Figure 4.13 presents the average size of the list kept for locally *suspected* nodes with different node threshold (T_{revoc}) and alarm threshold (T_{alarm}) which are detailed in Section 4.2. Round threshold (T_{round}) is chosen as 10. Increasing T_{revoc} increases the average size of lists kept for locally *suspected* nodes. The average list size linearly increases with the increase in T_{alarm} which is an expected result. When T_{alarm} is high, a node needs to detect more anomalies to deem a node as globally suspected, and thus, delete it from its locally *suspected* node list as explained in Figure 4.1.

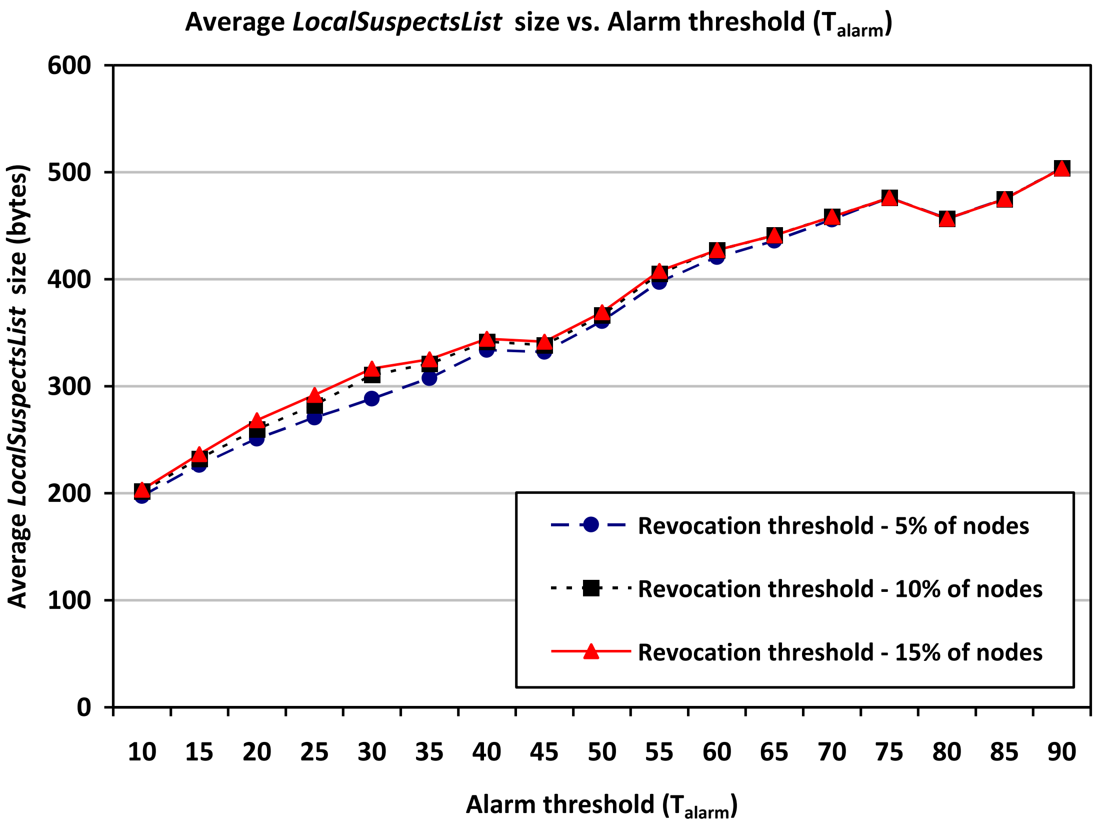

Figure 4.13: Average *LocalSuspectsList* size vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 10$. Wormhole ends are chosen randomly.

In Figure 4.14, the impact of round threshold (T_{round}) on detection round is analyzed with varying node threshold (T_{revoc}) and alarm threshold (T_{alarm}) values. T_{round} is set to 20 unlike the case in Figure 4.13. Simulation results do not show a major difference except a slight increase in when T_{alarm} is low in Figure 4.14 as compared to the results presented in Figure 4.13. The results are more or less same as compared to the case presented in Figure 4.13.

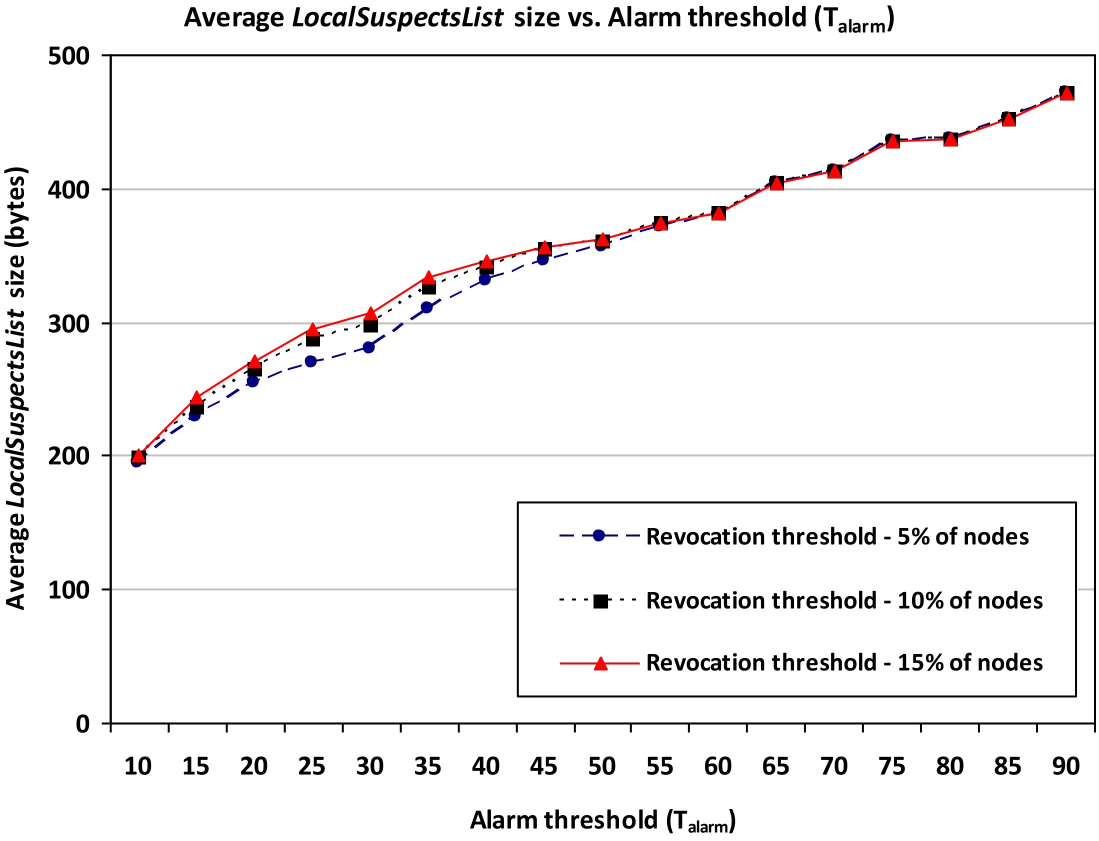

Figure 4.14: Average *LocalSuspectsList* size vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 20$. Wormhole ends are chosen randomly.

Figure 4.15 analyzes the effect of wormhole location on the list size which is for locally *suspected* nodes. Only location of the wormhole is different from the case shown in Figure 4.13. Using the same reasoning, when we locate the wormhole ends at $(25,25)$ and $(75,75)$, we make sure that the wormhole ends are not on the borders of the deployment area, and as a result, more nodes are affected by the wormhole. Due to the increase in neighboring connections which creates more anomalies in the network, wormhole becomes more detectable. This case is mentioned in Section 4.3.2 while discussing the impact of wormhole location on detection round. Wormhole is detected sooner if wormhole is located in such a way. Hence, the list size for keeping locally *suspected* nodes decreases as compared to the results presented in Figure 4.13.

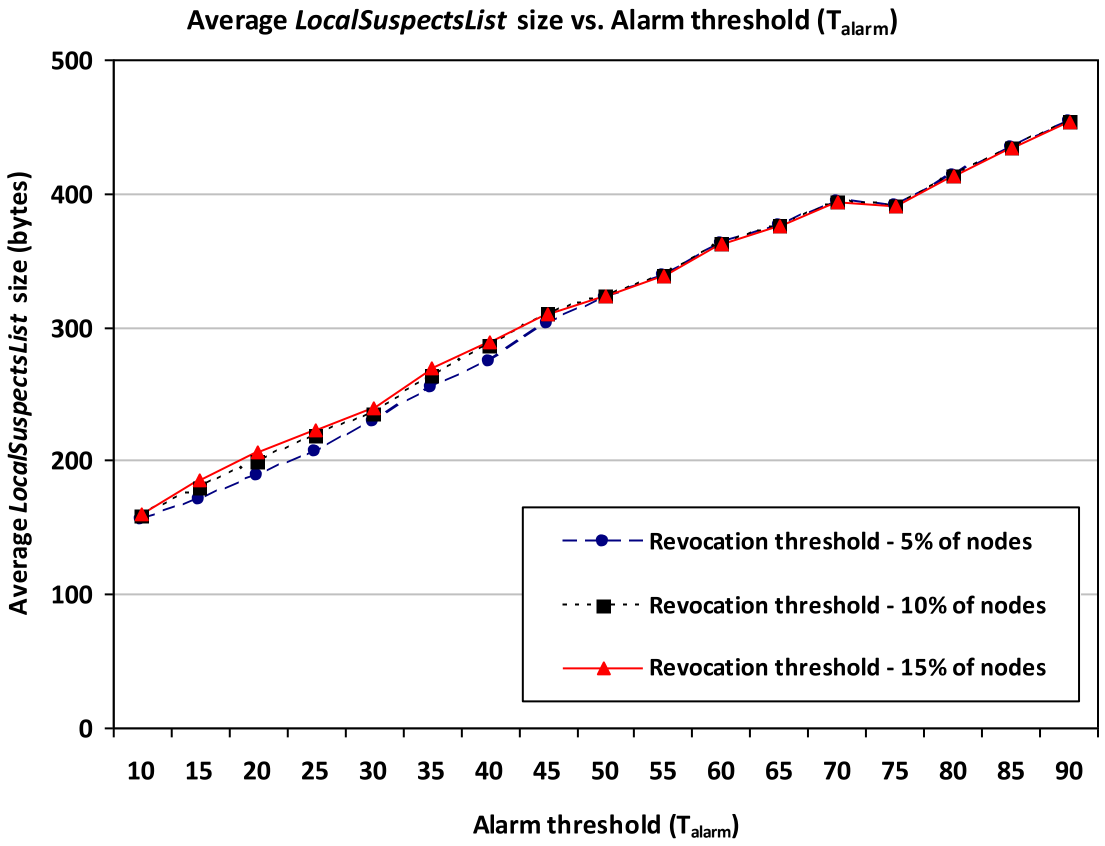

Figure 4.15: Average *LocalSuspectsList* size vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 10$. Wormhole ends are at $(25,25)$ and $(75,75)$.

The impact of increasing round threshold (T_{round}) is shown in Figure 4.16 under different node threshold (T_{revoc}) and alarm threshold (T_{alarm}) values. T_{round} is set to 20. The list size increases with the increase of T_{round} as compared to the results presented in Figure 4.15.

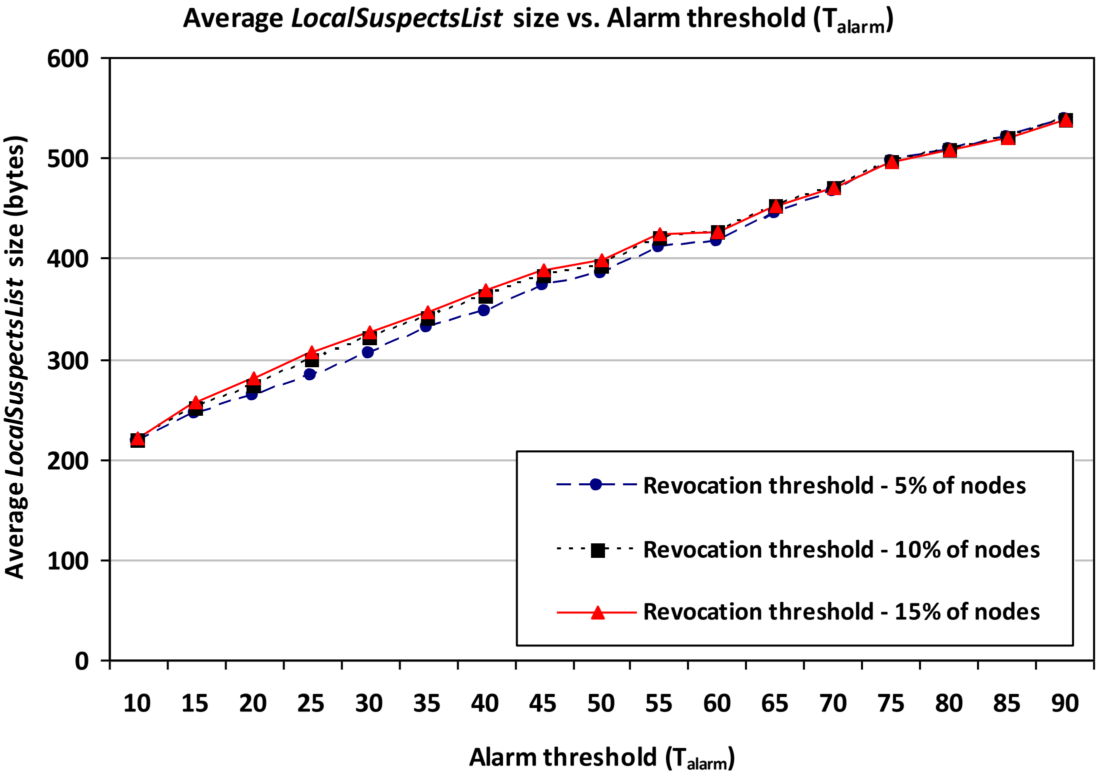

Figure 4.16: Average *LocalSuspectsList* size vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 20$. Wormhole ends are at $(25,25)$ and $(75,75)$.

Figure 4.17 shows the average size of the list kept for globally suspected nodes with varying node threshold (T_{revoc}) and alarm threshold (T_{alarm}) which are detailed in Section 4.2. Round threshold (T_{round}) is set to 10. Increase in T_{revoc} means that more nodes are required to agree on revoking a node, and thus, delete it from its globally *suspected* node list as explained in Figure 4.3. If T_{alarm} increases, the frequency of broadcasting globally suspected nodes decreases since more anomalies need to be detected to deem a node as globally suspected. Hence, the list size decreases as expected.

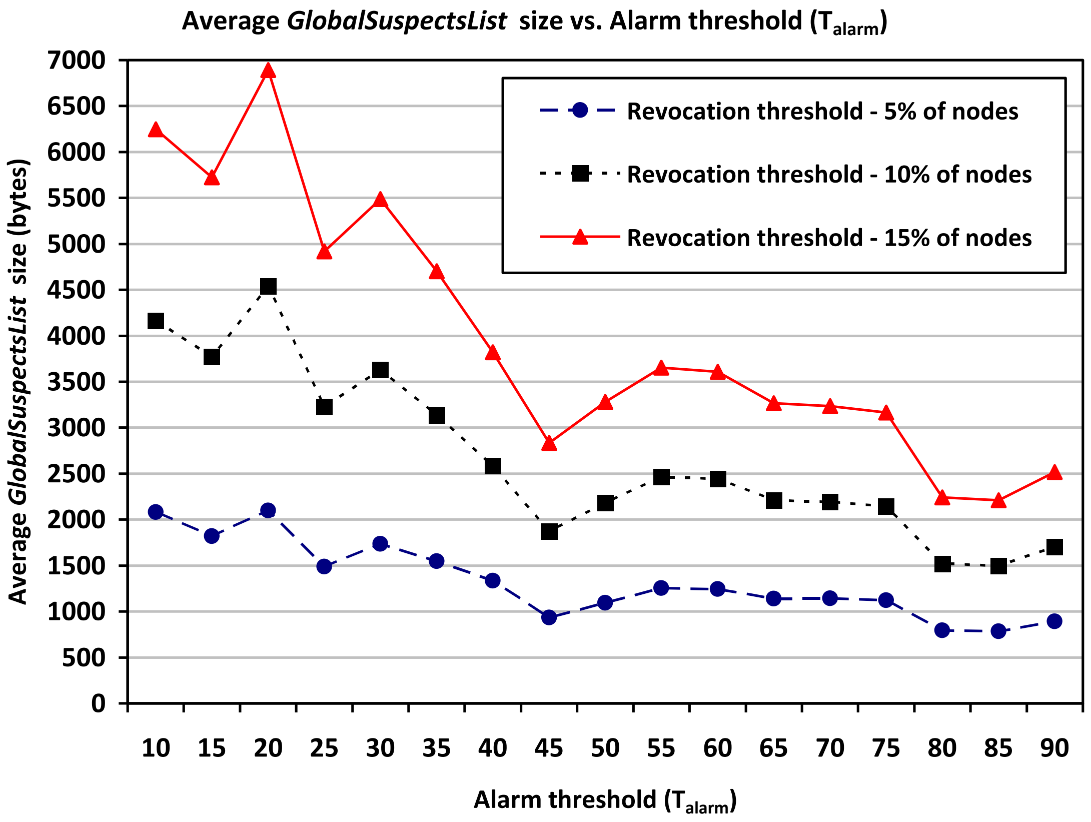

Figure 4.17: Average *GlobalSuspectsList* size vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 10$. Wormhole ends are chosen randomly.

Figure 4.18 analyzes the effects of round threshold (T_{round}) on the average list size which is kept for globally *suspected* nodes. T_{round} is set to 20 unlike the case in Figure 4.17. Although, there are not major differences from the results shown in Figure 4.17, the list size slightly decreases. Also, the decrease in the list size with the increase in T_{alarm} is sharper and more observable when T_{revoc} is 10.

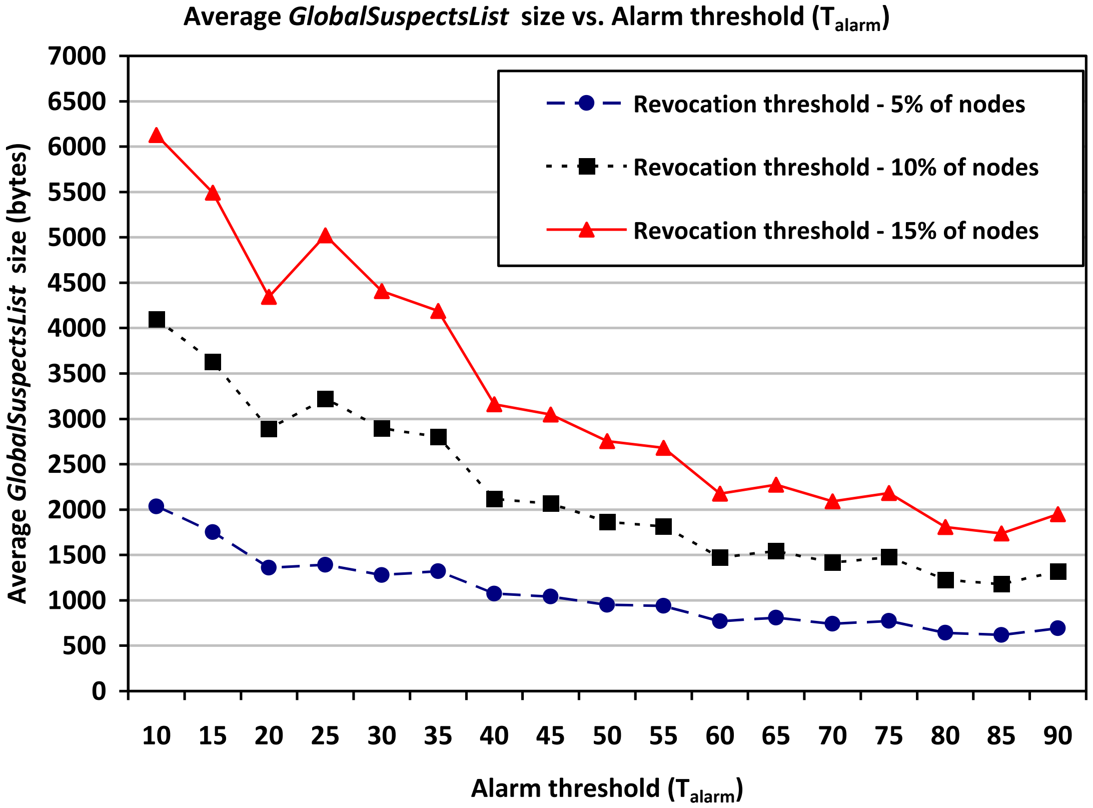

Figure 4.18: Average *GlobalSuspectsList* size vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 20$. Wormhole ends are chosen randomly.

Figure 4.19 shows the effect of wormhole location on the list size which is for keeping globally *suspected* nodes. Due to locating wormhole ends at $(25,25)$ and $(75,75)$, wormhole affects more nodes in the network. However, by doing so, it increases the fake neighboring connections and creates more anomalies which lead to broadcasting more globally *suspected* nodes to the network. The list size increases as compared to the results presented in 4.17.

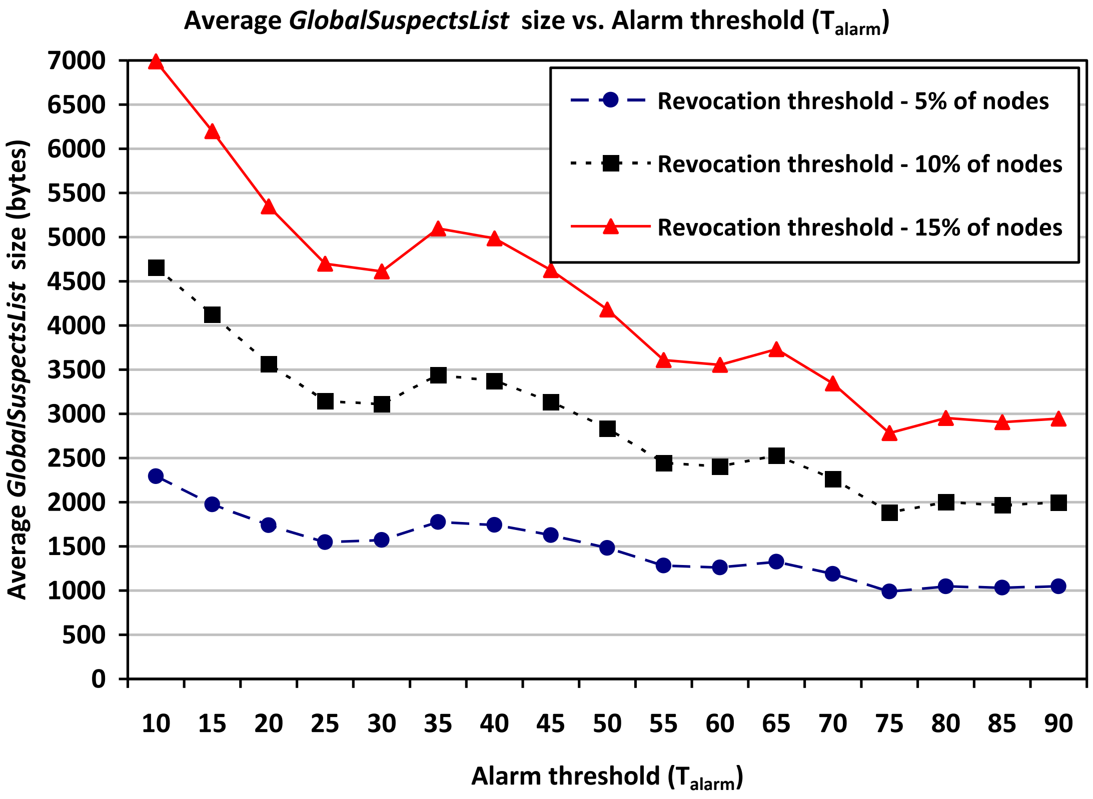

Figure 4.19: Average *GlobalSuspectsList* size vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 10$. Wormhole ends are at $(25,25)$ and $(75,75)$.

The effect of increasing round threshold (T_{round}) on the average list size is analyzed in Figure 4.20. T_{round} is chosen as 20. As T_{round} increases, the average list size decreases slightly compared to the results in Figure 4.19.

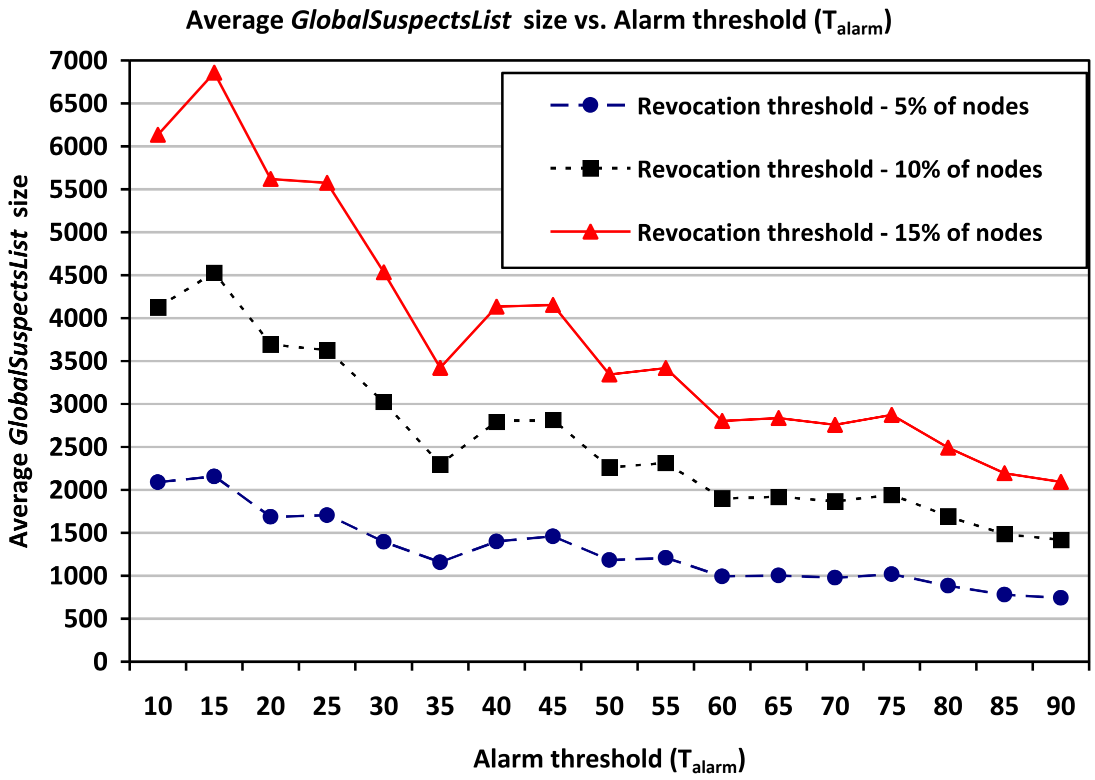

Figure 4.20: Average *GlobalSuspectsList* size vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 10$, $T_{revoc} = 20$, and $T_{revoc} = 30$. $T_{round} = 20$. Wormhole ends are at $(25,25)$ and $(75,75)$.

4.3.5. Sensitivity against Node Density and Size of Deployment Area

In this section, detection rate and false positive rate are analyzed based on the changes in node density and size of the deployment area. In Figure 4.21, Figure 4.22, Figure 4.23 and Figure 4.24, the deployment area is increased to $200 \times 200\,\text{m}^2$ without changing the node density in the area. To do so, the number of nodes is set to 800. In Figure 4.25, Figure 4.26, Figure 4.27 and Figure 4.28, the node density is increased by increasing the number of nodes to 400 while the deployment area is the same, $100 \times 100\,\mathrm{m}^2$. Two values of round threshold, $T_{round} = 10$ and $T_{round} = 20$, are simulated where the locations of the wormhole ends are set to $(25,25)$ and $(75,75)$.

Figure 4.21 shows the detection rate when $T_{round} = 10$. Since the number of nodes is increased, the required number of nodes to agree on revocation of a *suspected* node (T_{revoc}) also increases. Although the node density is the same, increase in deployment area and T_{revoc} causes a sharp decrease in the detection rate. This decrease can be observed in a clearer way when the results are compared with the case presented in Figure 4.3, where deployment area is $100 \times 100\,\mathrm{m}^2$ and the number of nodes is 200. Detection rate is close to 100% until $T_{alarm} = 40$ in the results shown in Figure 4.3, while it decreases below 50% in Figure 4.21.

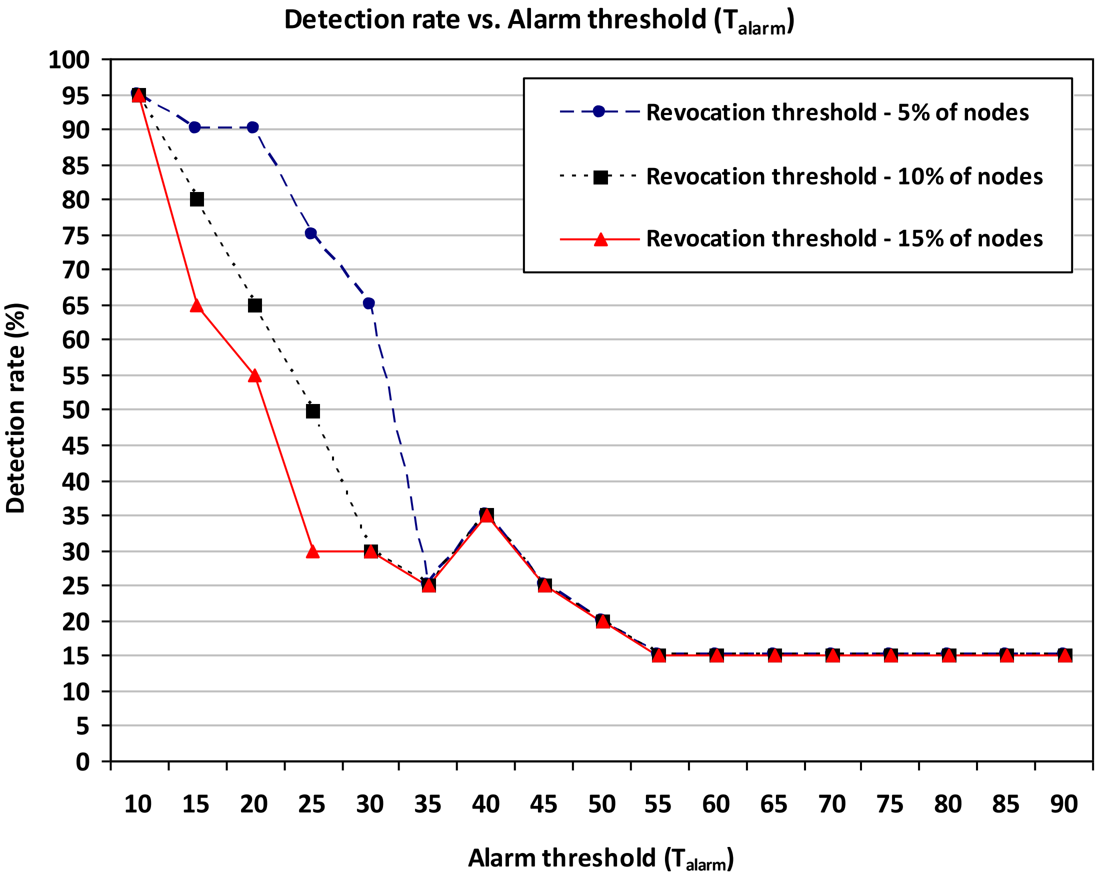

Figure 4.21: Detection rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 40$, $T_{revoc} = 80$, and $T_{revoc} = 120$. $T_{round} = 10$. Wormhole ends are at $(25,25)$ and $(75,75)$.

In Figure 4.22, the only difference is the value of T_{round}. T_{round} is set to 20 in this case. Detection rate is not high as compared to Figure 4.21 when T_{alarm} is low. However, the decrease in detection rate presented in Figure 4.21 is sharper as compared to the results shown in Figure 4.22 when T_{alarm} does not exceed 50.

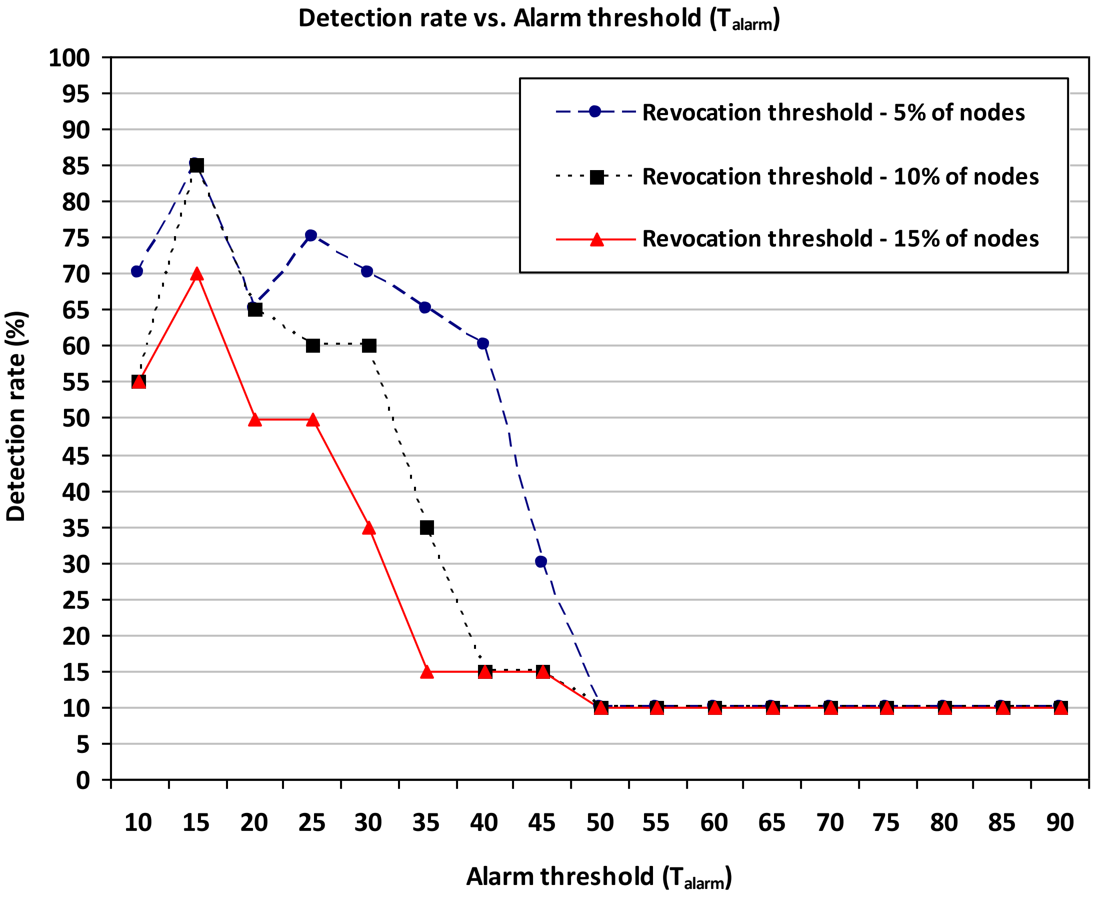

Figure 4.22: Detection rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 40$, $T_{revoc} = 80$, and $T_{revoc} = 120$. $T_{round} = 20$. Wormhole ends are at $(25,25)$ and $(75,75)$.

Figure 4.23 presents the effect of increase in deployment area and T_{revoc} on false positive rate. T_{round} is set to 10. Due to increase in T_{revoc}, more nodes need to agree on revoking a suspected node; thus, the number of falsely revoked nodes decreases as compared to the results presented in Figure 4.7 where deployment area is $100 \times 100\,\mathrm{m}^2$ and the number of nodes is 200. Also, since the deployment area is much larger, the probability of a node witnessing an anomaly also decreases which can be seen as a reason of the decrease in the false positive rate.

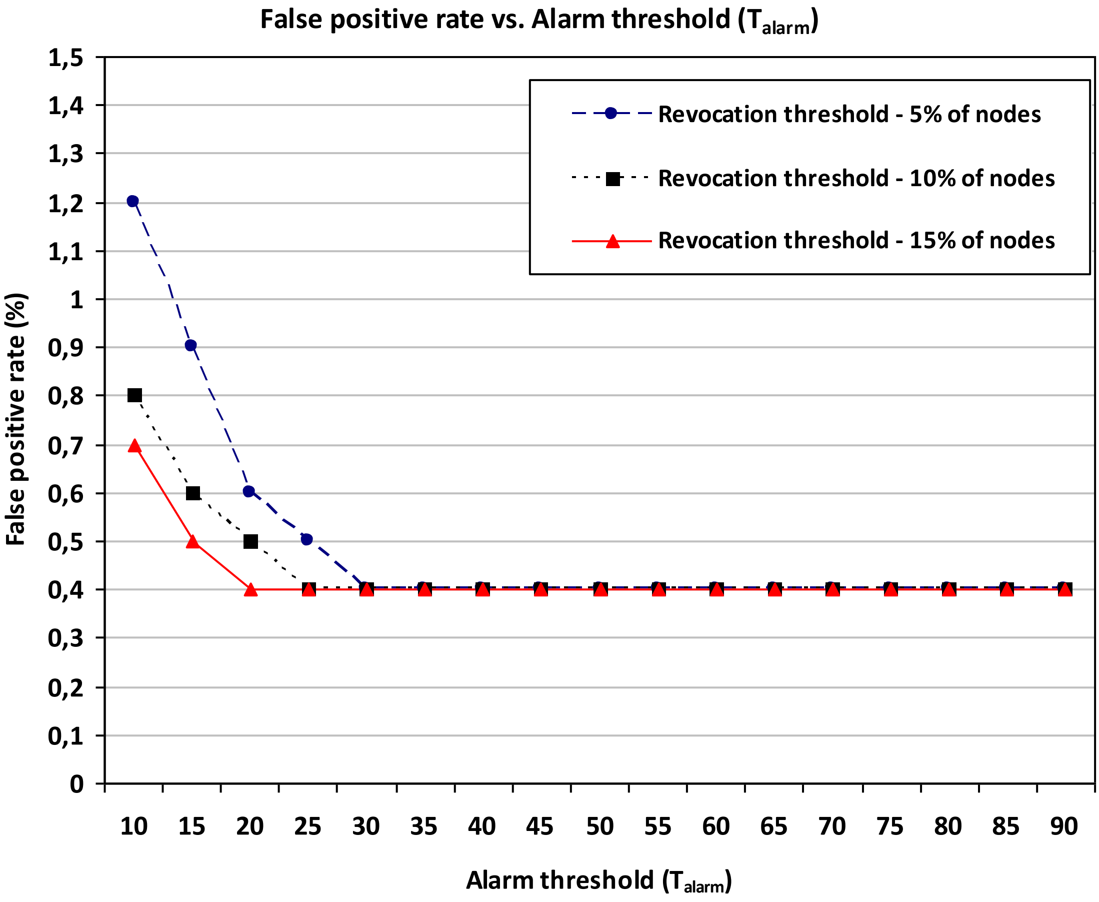

Figure 4.23: False positive rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 40$, $T_{revoc} = 80$, and $T_{revoc} = 120$. $T_{round} = 10$. Wormhole ends are at $(25,25)$ and $(75,75)$.

In Figure 4.24, the impact of the increase in T_{round} is shown under the new simulation setup. T_{round} is set to 20, which is the only difference from the case presented in Figure 4.23. False positive rate increases with the increase of T_{round}. This is an expected result since increasing T_{round} means that a node has more time, compared to the case where T_{round} is 10, in order to witness a repetitive anomaly. In Figure 4.24, false positive rate stabilizes at 0.5%, while it stabilizes at 0.4% in the results presented in Figure 4.23.

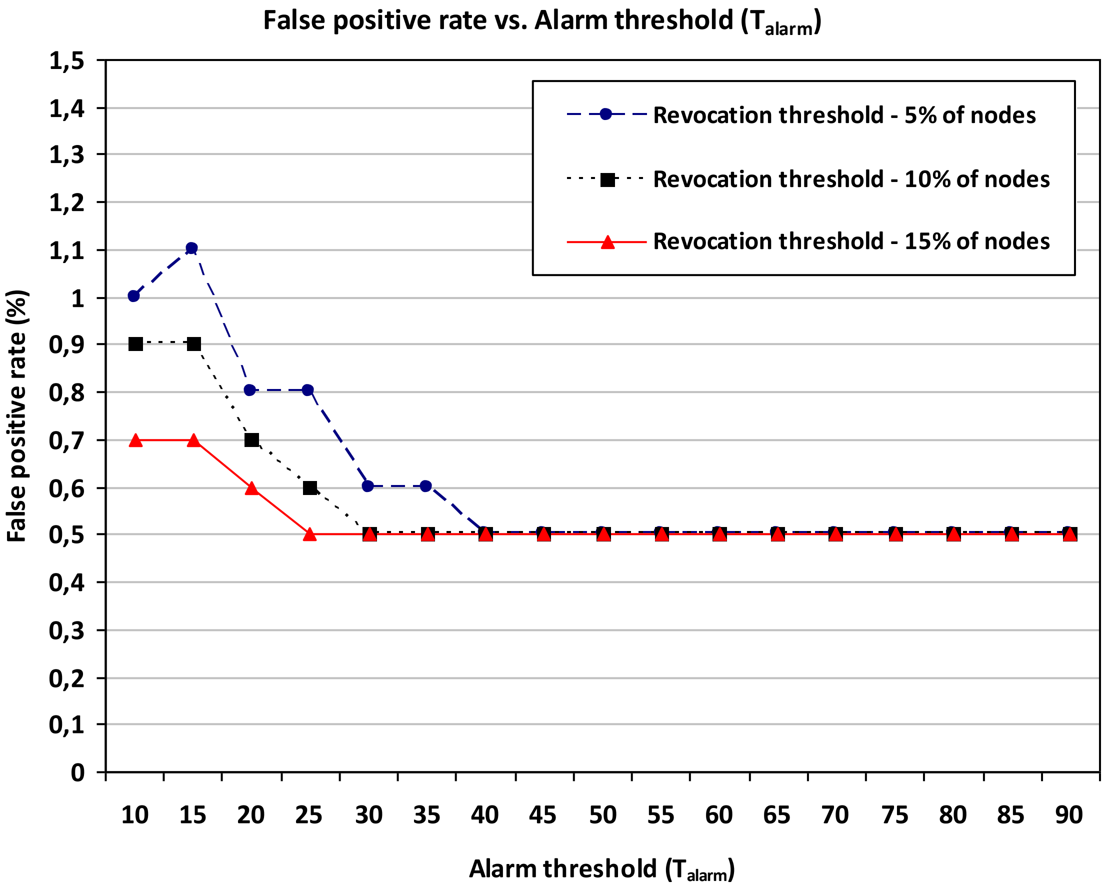

Figure 4.24: False positive rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 40$, $T_{revoc} = 80$, and $T_{revoc} = 120$. $T_{round} = 20$. Wormhole ends are at $(25,25)$ and $(75,75)$.

In Figure 4.25, Figure 4.26, Figure 4.27 and Figure 4.28, the deployment area is the same, $100 \times 100\,\mathrm{m}^2$, but the number of nodes is increased to 400. The goal of these simulations is to analyze the effect of node density to our detection scheme.

Figure 4.25 presents the detection rate under the new case with increased density. The number of nodes to revoke a *suspected* node, which is revocation threshold, increases with the increase in the number of nodes. In overall, the detection rate is not high for low T_{alarm} values as compared to the results shown in Figure 4.3. However, for T_{alarm} values above 55, the detection rate does not decrease as much as of the case in Figure 4.3. The results presented in Figure 4.3, detection rate is 15% while in this case it is 20% for high T_{alarm} values.

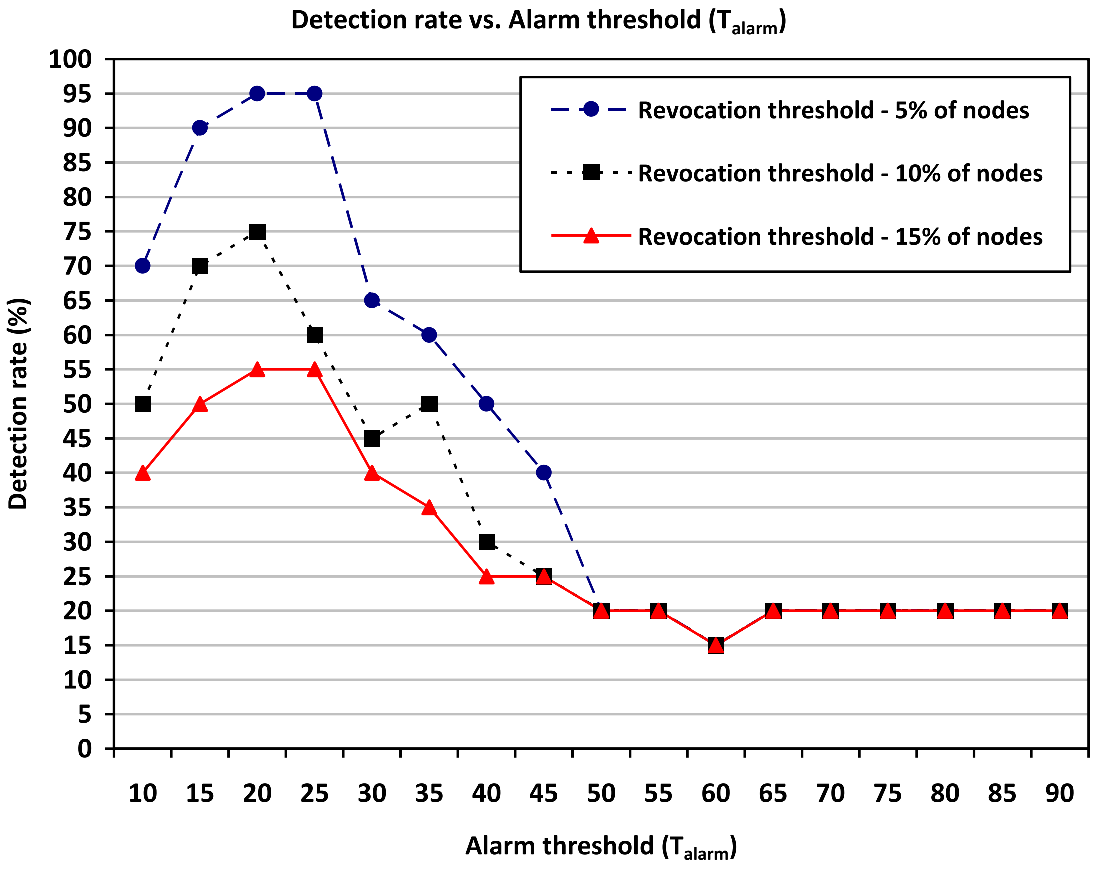

Figure 4.25: Detection rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 20$, $T_{revoc} = 40$, and $T_{revoc} = 60$. $T_{round} = 10$. Wormhole ends are at $(25,25)$ and $(75,75)$.

Figure 4.26 shows the impact of the change in T_{round} on detection rate. The results get better when T_{round} is set to 20. Especially for $T_{revoc} = 20$, it gets close to the results presented in Figure 4.4. Moreover, a detection rate of 20% is achieved for T_{alarm} values above 65, which is decreasing to 10% in the Figure 4.4.

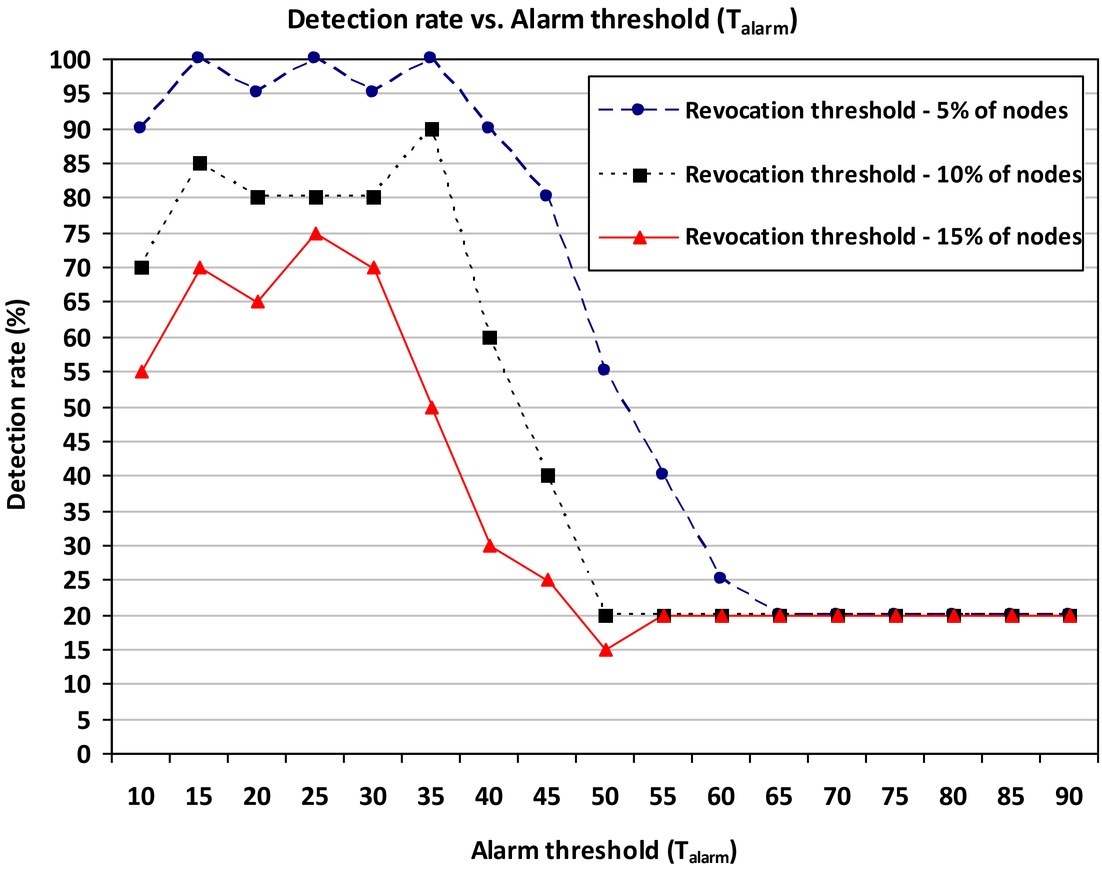

Figure 4.26: Detection rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 20$, $T_{revoc} = 40$, and $T_{revoc} = 60$. $T_{round} = 20$. Wormhole ends are at $(25,25)$ and $(75,75)$.

In Figure 4.27, the effect of the increase in node density on false positive rate is presented. T_{round} is set to 20. The number of falsely revoked nodes decreases with the increase of node density as compared to the results shown in Figure 4.3. Increasing the number of nodes also means increasing the revocation threshold, T_{revoc}. When T_{revoc} increases, it becomes hard to revoke a *suspected* node since more nodes are required to broadcast alarm for that *suspected* node.

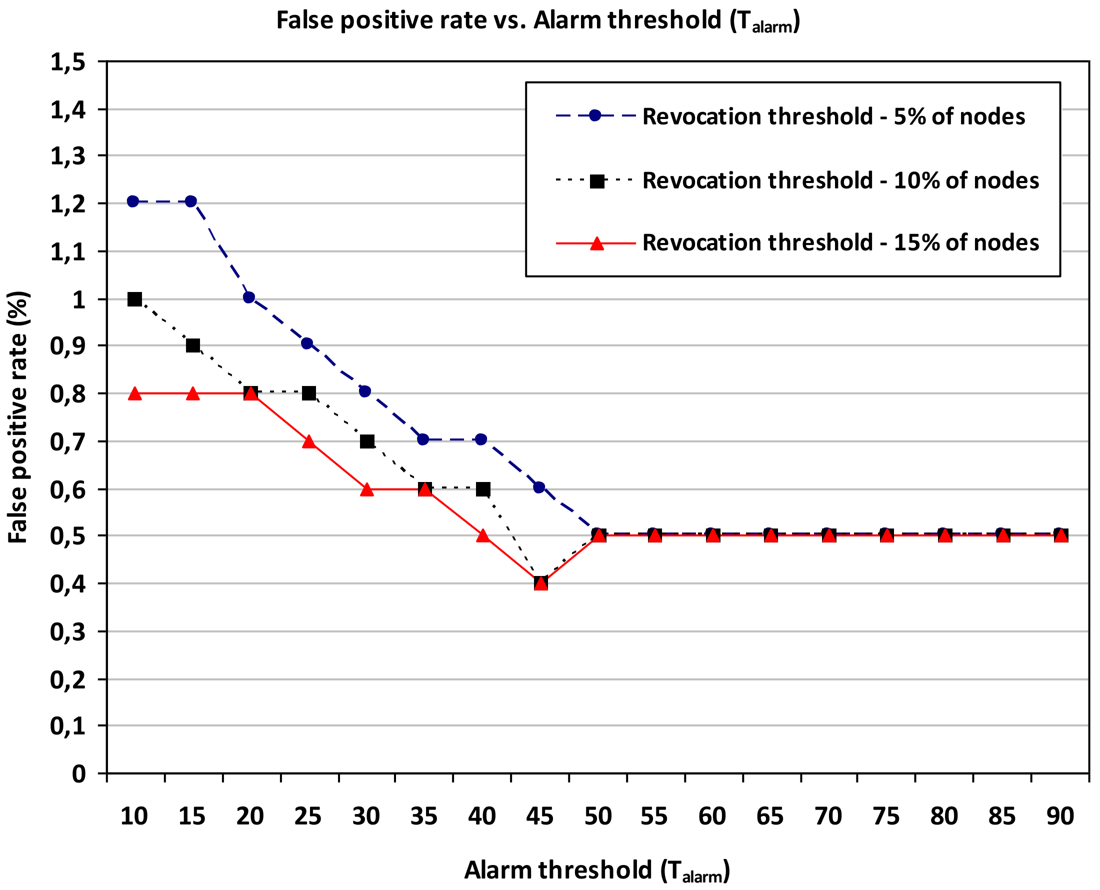

Figure 4.27: False positive rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 20$, $T_{revoc} = 40$, and $T_{revoc} = 60$. $T_{round} = 10$. Wormhole ends are at $(25,25)$ and $(75,75)$.

Figure 4.28 presents the impact of the change in T_{revoc} on false positive rate. T_{revoc} is set to 20. For T_{alarm} values below 50, there is a slight increase in false positive rate as compared to the results shown in Figure 4.27. On the other hand, when T_{alarm} exceeds 55, false positive rate decreases to 0.4% while this value is 0.5% for the case $T_{revoc} = 10$.

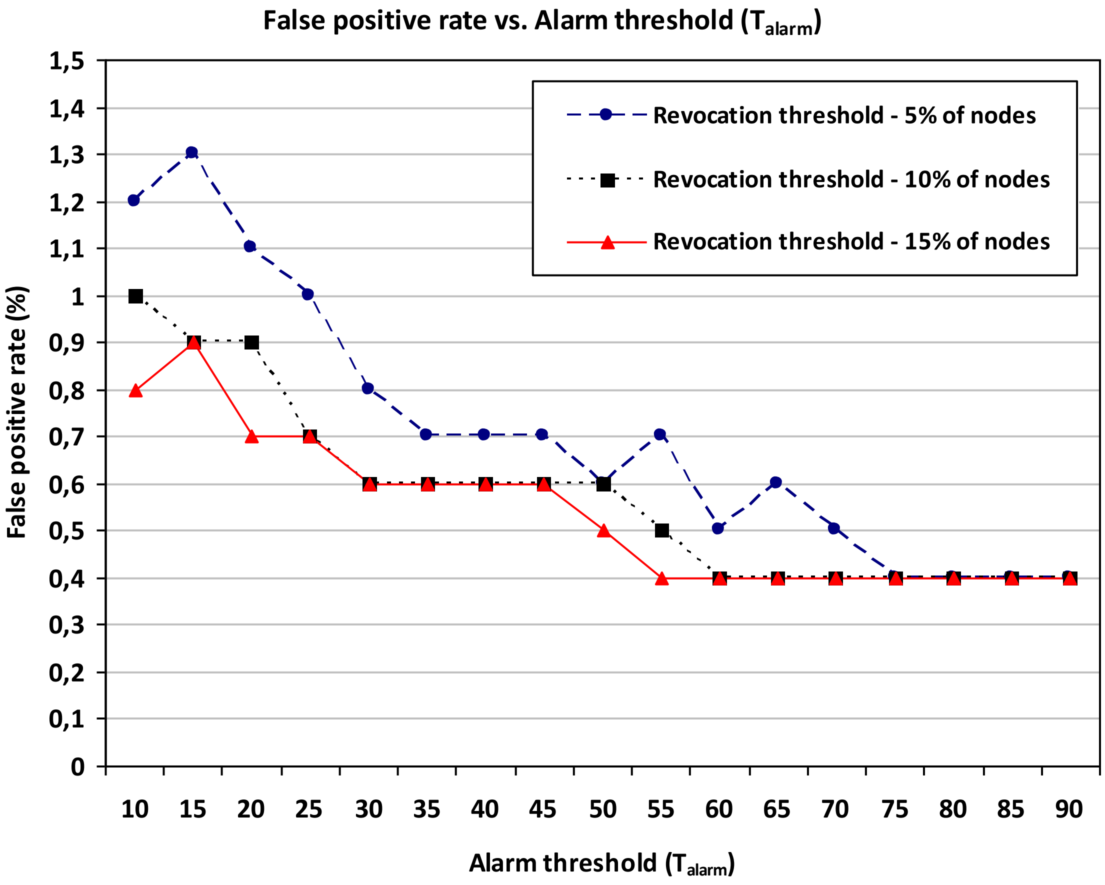

Figure 4.28: False positive rate vs. Alarm threshold (T_{alarm}) for $T_{revoc} = 20$, $T_{revoc} = 40$, and $T_{revoc} = 60$. $T_{round} = 20$. Wormhole ends are at $(25,25)$ and $(75,75)$.

5. CONCLUSION

Wormhole attack is a physical attack which is a serious threat especially on routing protocols. Attracting the network traffic on a low-latency wormhole link empowers an attacker to perform various malicious activities such as traffic analysis, denial of service attacks, or just selectively drop data packets via controlling this wormhole link.

In this thesis, we propose a distributed wormhole detection scheme for mobile wireless sensor networks which utilizes mobility of sensor nodes to detect wormhole attack. Our detection scheme is composed of two phases which are: (i) stabilization phase, and (ii) detection phase. In stabilization phase, two network features (i.e. network node density, standard deviation in network node density) are estimated through using neighboring information in a local manner. In detection phase, wormhole attack is detected via observing anomalies in the neighbor nodes' behaviors based on these estimated network features and the neighboring information. We analyzed the performance of proposed scheme via simulations using different system parameters. The results show that our scheme achieves a detection rate up to 100% with very small false positive rate (at most 1.5%) if the system parameters are chosen accordingly. Moreover, our solution requires neither additional hardware nor tight clock synchronization which are both costly for sensor networks.

6. REFERENCES

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002) Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422.

[2] Hu, Y.C., Perrig, A., and Johnson, D.B. (2003) Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks. *IEEE INFOCOM*, 3:1976-1986.

[3] Hu, L., and Evans, D. (2004) Using Directional Antennas to Prevent Wormhole Attacks. *Proceedings of the 11th IEEE Network and Distributed System Security Symposium (NDSS)*, p.22-32.

[4] Zhou, Y., Fang, Y., and Zhang, Y. (2008) Securing Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials*, vol. 10, no. 3, pp. 6-28.

[5] Capkun, S., Buttyan, L., and Hubaux, J. (2003) SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks. *Proceedings of 1st ACM workshop on Security of Ad Hoc and Sensor Networks (SASN)*, pp. 21-32.

[6] Buttyan, L., Dora, L., and Vajda, I. (2005) Statistical Wormhole Detection in Sensor Networks. *Lecture Notes in Computer Science (LNCS)*, 3813:128-141.

[7] Wang, W., and Bhargava, B. (2004) Visualization of Wormholes in Sensor Networks. *Proceedings of the ACM workshop on Wireless security (Wise'04)*, pp. 51-60.

[8] Song, N., Qian, L., and Li, X. (2005) Wormhole Attacks Detection in Wireless Ad Hoc Netwroks: A Statistical Analysis Approach. *Proceedings of the 19th International Parallel and Distributed Processing Symposium (IPDPS'05)*.

[9] Qian, L., Song, N., and Li, X. (2007) Detection of wormhole attacks in multi-path routed wireless ad hoc networks: A statistical analysis approach. *Journal of Network and Computer Applications*, 30(1):308-330.

[10] Lazos, L., Poovendran, R., Meadows, C., Syverson, P., and Chang, L.W. (2005) SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks. *Proceedings of ACM workshop on Wireless Security,* pp. 21-30.

[11] Khalil, I., Bagchi, S., and Shroff, N.B. (2005) LITEWORP: A Lightweight Countermeasure for the Wormhole Attack in Multihop Wireless Networks. *International Conference on Dependable Systems and Networks (DSN)*, pp. 612-621.

[12] Ko, Y., Shankarkumar, V., and Vaidya, N. (2000) Medium access control protocols using directional antennas in ad hoc networks. *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications (INFOCOM),* pp. 13-21.

[13] Choudhury, R., Yang, X., Ramanathan, R., and Vaidya, N. (2002) Using directional antennas for medium access control in ad hoc networks. *8th ACM International Conference on Mobile Computing and Neyworking (MobiCOM)*.

[14] Khalil, I., Bagchi, S., and Shroff, N.B. (2008) MOBIWORP: Mitigation of the Wormhole Attack in Mobile Multihop Wireless Networks. *Ad Hoc Networks*, 6(3):344-362.

[15] Hu, L., and Evans, D. (2004). Localization for Mobile Sensor Networks. *ACM MobiCOM'04*, pp. 45-57.

[16] Liu, D., Ning, P., and Du, W. (2005) Detecting Malicious Beacon Nodes for Secure Location Discovery in Wireless Sensor Networks. *The 25th International Conference on Distributed Computer Systems (ICDCS'05)*, pp. 609-619.

[17] Du, W., Fang, L., and Ning, P. (2005) LAD: Localization Anomaly Detection for Wireless Sensor Networks. *Proceedings of the 19th IEEE International Parallel & Distributed Processing Symposium (IPDPS'05)*.

[18] Sastry, N., Shankar, U., and Wagner, D. (2003) Secure verification of location claims. *ACM workshop on Wireless Security (WiSe'03)*, pp. 1-10.

[19] Xu, Y., Ouyang, Y., Le, Z., Ford, J., and Makedon, F. (2007) Analysis of Range-Free Anchor-Free Localization in a WSN under Wormhole Attack. *Proceedings of the 10th ACM Symposium on Modelling, Analysis, and Simulation of Wireless and Mobile Systems*, pp. 344-351.

[20] Kong, F., Li, C., Ding, Q., Cui, G., and Cui, B. (2009) WAPN: a distributed wormhole attack detection approach for wireless sensor networks. *Journal of Zhejiang University – Science*, 10(2):279-289.

[21] Juang, P., Oki, H., Wang, Y., Martonosi, M., S.Peh, L., and Rubenstein, D. (2002) Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. *SIGOPS Oper. Syst. Rev*., 36(5):96-107.

[22] Yick, J., Mukherjee, B., and Ghosal, D. (2008) Wireless sensor network survey. *Computer Networks*, 52(12):2292-2330.

[23] Ahmad-Kassem, A., and Mitton, N. (2010) Adapting dynamically neighborhood table entry lifetime in wireless sensor networks. *Wireless Communications and Signal Processing (WCSP'10)*.

[24] Pham, H., and Jha, S. (2004) Addressing Mobility in Wireless Sensor Media Access Protocol. *Intelligent Sensors, Sensor Networks and Information Processing Conference*.

[25] Kohvakka, M., Suhonen, J., Kuorilehto, M., Kaseva, V., Hannikainen, M., and Hamalainen, T.D. (2009) Energy-efficient neighbor discovery protocol for mobile wireless sensor networks. *Ad Hoc Networks*, 7 (1), pp. 24-41.

[26] Bagchi, S., Hariharan, S., and Shroff, N. (2007) Secure Neighbor Discovery in Wireless Sensor Networks. *ECE Technical Reports. Paper 360*.

[27] Curiac, D.-I., Plastoi, M., Banias, O., Volosencu, C., Tudoroiu, R., and Doboli, A. (2009) Combined Malicious Node Discovery and Self-Destruction Technique for Wireless Sensor Networks. *Sensor Technologies and Applications (SENSORCOMM '09)*, pp.436-441.

[28] Plastoi, M., and Curiac, D.-I. (2009) Energy-driven methodology for node self-destruction in wireless sensor networks. *Applied Computational Intelligence and Informatics (SACI '09)*, pp.319-322.