IF : 3.62 | IC Value 70.36

Volume-5, Issue-11, November - 2016 • ISSN No 2277 - 8160

Original Research Paper

Computer Science

IDS Foundation for Wireless Sensor Netowrk

D. P. Mishra Research Scholar CSVTU Bhilai

Ramesh Kumar Professor, Department of Computer Sc. & Engineering, CSVTU Bhilai

ABSTRACT *Intrusion detection system for wireless sensor network is one of the growing research field in recent years. Wireless sensor networks (WSN) consist of tiny devices. These tiny devices have limited energy, computational power, transmission range and memory. However, wireless sensor networks are deployed mostly in open and unguarded environment. Therefore, intrusion detection is one of the most important security aspects for wireless sensor networks. There are two types of intrusion detection mechanism: anomaly based and signature based. In this paper, we have tried to set the theoretical foundation of this new research area first, before doing design and implement an Intrusion Detection System (IDS) specifically for sensor networks.*

KEYWORDS : WSN, IDS, Sensor Networks, Anomaly, Security, Threat, Security, attack, Denial of Service (DoS), IDS Architectures, Cluster-based IDS, Anomaly-based IDS, Signature based IDS & Hybrid IDS

INTRODUCTION

Intrusion is set of actions that try to compromise the data integrity, user's confidentiality or service availability can be termed as intrusion, while a system that attempts to detect such malicious actions of network or compromised nodes is called IDS. An Intrusion Detection System (IDS) that can *detect* a third party's attempts of exploiting the insecurities of the network, even such attacks have not been experienced before. Intrusion detection systems provide a wrapper of in-depth protection for wired networks. However, little research has been performed about intrusion detection in the areas of wireless sensor networks. The reason may be the concept of "intrusion detection" is not known or clearly specified context of such networks. However, the security level of wireless networks can be enhanced up to certain limit by implementing IDS.

The primary functions of IDS are to monitor users' activities, network behavior and different layers. Still a perfect single defense is neither feasible nor possible in wireless networks, as there always exist some architectural weaknesses, software issues or design issues which may be compromised by the intruders. The best practice to secure the wireless networks is to implement multi lines of security mechanisms, that is why, IDS is more critical in wireless networks which is viewed as a passive defense, as it is not intended to prevent attacks, instead it alert network administrator about possible attacks well in time to stop or reduce the impact of the attack. The accuracy of intrusion detection is measured in terms of false positives and false negative alarms for indicating occurrence of intrusion, where an ideal IDSs attempt to minimize both these [1]. It is essential to set the theoretical foundation of this new research area first, before trying to design and implement an Intrusion Detection System (IDS) specifically for sensor networks.

Designing an IDS for Sensor Networks

In intrusion detection, we wish to provide an automated mechanism that identifies the source of an attack and generates an alarm to notify the network or the administrator, so that appropriate preventive actions can take place. As an attack we consider any set of actions that target the computing or networking resources of our system. Attackers may be using an external system without authorization or have legitimate access to our system but are abusing their privileges (i.e., an insider attack). It is important to realize that the IDS will be activates after an intrusion attempt has occurred. It does not prevent these attempts in the first place.

Limitations and Requirements of IDS for WSN

Every sensor node is having limited communication and computational capability and a very short radio range. Furthermore, every node is acting as weak unit that can be easily compromised by an adversary, who may load malicious software to launch an insider attack, figure-1 shows weakness of WSN. In this context a distributed architecture, based on cooperative node would be appropriate solution. An IDS for sensor networks must satisfy the following properties:

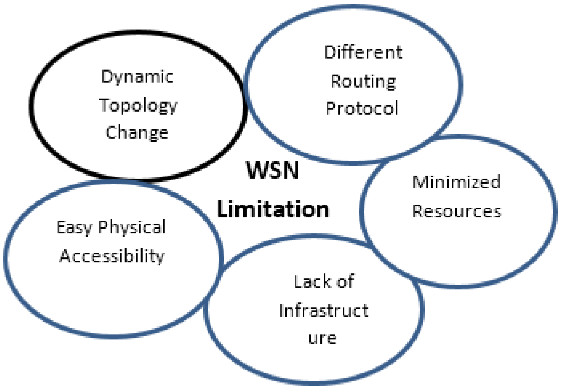

Figure 1: *Limitations of Wireless Sensor Network*

Localize auditing. An IDS for sensor networks should work with localized and partially audited data. In such networks there are no centralized points (apart from the base station) that can collect information or data for the entire network, so this approach fits the sensor networks paradigm. Dealing with partial data means that the IDS should also address the problem of high false alarm rate.

Minimize resources. An IDS for sensor networks must use minimal amount of resources. The wireless network is not having stable connections, physical resources of network and devices, such as bandwidth and power is limited, disconnection may take place at any time. In addition, communication between nodes for intrusion detection should not utilize too much of the available bandwidth.

Not to trust single node. In a collaborative IDS, it is assumed that nodes cannot be trusted. Unlike wired networks, since wireless sensor nodes are prone to get easily compromised. These nodes may behave normal with respect to the routing of the information in order to overcome detection by the IDS. However, it is expected that they may expose a malicious behavior to obstruct the successful detection of another intruder node. Therefore, in cooperative IDS system, it ex expected that IDS should not trust on even as single node.

Be truly distributed. In order to distribute the load of the intrusion detection, process of data collection and analysis must be performed at different locations. The distributed approach is applicable for the operation and execution of IDS engine and alert correlation module.

Support addition of new nodes. Sensor network must support addition of new nodes as in practice it is highly needed to populate sensor network with more and more nodes after its deployment. An IDS should be able to support this operation and distinguish it from an attack (e.g. wormhole attack) that has the same effect.

Be secure. An IDS should resist hostile attack against monitoring node. Compromising a monitoring node and controlling the behavior of the embedded IDS agent should not allow an adversary to revoke a genuine node from the network, or keep another intruder node undetected.

Intrusion Detection Architectures

Traditionally, intrusion detection systems for fixed networks were categorized in two categories - host-based and network-based. The host-based architecture was the first architecture to be explored in intrusion detection. A host-based intrusion detection system (HIDS) is designed to monitor, detect, and respond to a given host (node). Decision made is based on information and audit review for suspicious activity of concerned node. This approach contradicts the distributed nature of sensor networks and makes infeasible to detect network attacks. A network-based architecture is clearly more appropriate here.

Network-based intrusion detection systems (NIDS) use raw packets as the data source. A network-based IDS typically listens on the network, and captures and examines each and every individual packets in real time. It can analyze the entire packet, not just the header. In wired networks, active scanning of packets from a network-based intrusion detection system is usually done at specific traffic concentration points, such as switches, routers or gateways. On the other hand, wireless sensor networks do not have any issue or bottlenecks specific to concentration. Since any of the participating node may act as a router and traffic is usually distributed for load balancing purposes. So, it is impossible to monitor the traffic at certain points.

while designing an IDS for sensor networks, we have to wisely take decision of locating decision agents, due to the distributed nature of the network and traffic routed within. One possible solution is to have an identical agent inside every node. That would be a realistic solution, in case agents were designed to be lightweight and cooperative through a distributed algorithm. Another solution would be to have a hierarchical model, where some more computationally intensive agents were placed on certain nodes, while other agents with restrictive tasks were placed on the rest of the nodes. We review systems using both solutions

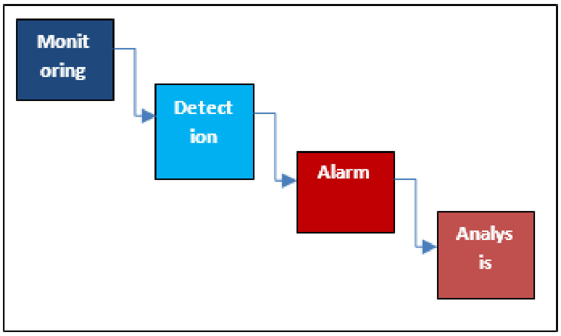

Figure-2 IDS Components

IDS is not capable of taking preventive action, since it is passive in nature, it can only detect intrusion and generate an alarm. Figure-2 shows major four components of IDS.

Broadly speaking, IDS has two major components such as:

Monitoring component, it is used for local events monitoring as well as neighbors monitoring.

Intrusion database, contains the records of latest misbehavior and reputation value for the neighbors.

Response component, is responsible for giving responding in case of intrusion, is detected. The response may be used to raise an alarm to alert the administrator or to broadcast the information to neighboring nodes about the misbehaving node.

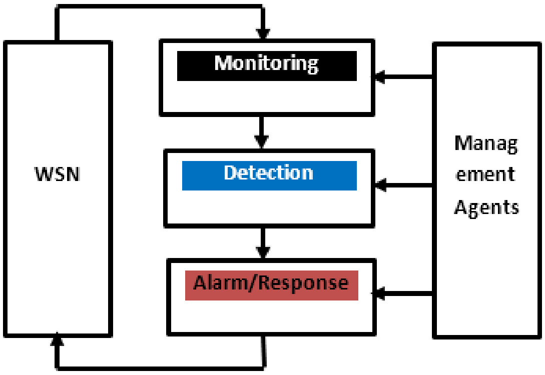

Figure-3: Basic IDS Architecture

The components and the response nature of IDS depends on the purpose and services of the IDS. For example, IDS designed for routing misbehavior would have different components and responses as compared to an IDS designed for physical and MAC layers' anomalies. Management agents are responsible for supporting the major operations of IDS System.

Intrusion Detection Techniques

In order to detect an intruder, we have to use a model of intrusion detection. We need to know what an IDS should look out for. In particular, an IDS must be able to distinguish between normal and abnormal activities in order to discover malicious attempts in time. However, this can be difficult, since many behavior patterns can be unpredictable and unclear. There are three major methods that an intrusion detection system can use to classify actions

Misuse detection. In misuse detection or signature-based detection systems [1], the observed behavior is compared with known attack patterns (signatures). So, action patterns that may pose a security threat must be defined and given to the system. The misuse detection system tries to recognize any "bad" behavior according to these patterns and depicted in figure-4. Any action that is not clearly prohibited is allowed. The main disadvantage of such systems is that they cannot detect novel attacks. Someone must continuously update the attack signature database. Another difficulty is that signatures must be written in a way to encompass all possible variations of the pertinent attack, and yet avoid flagging non-intrusive activity as an intrusive one.

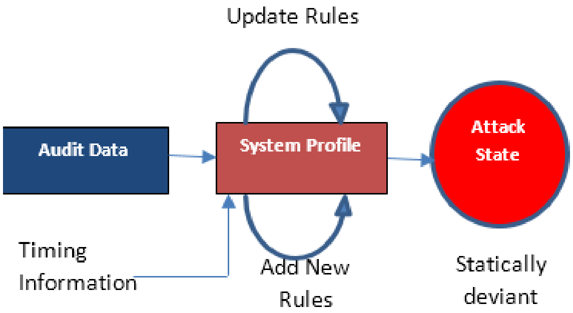

Figure-4: Misuse Detection

Anomaly detection. Anomaly detection [2] overcomes the limitations of misuse detection by focusing on normal behaviors, rather than attack behaviors. This technique first describes what constitutes a "normal" behavior (usually established by automated training) and then flags as intrusion attempts any activities varying from this behavior by a statistically significant amount. In this way there is a considerable possibility to detect novel attacks as intrusions shown in figure-5. There are two problems associated with this approach: First, a system can exhibit legitimate but previously unseen behavior. This would lead to a substantial false alarm rate, where anomalous activities that are not intrusive are flagged as intrusive. Second, and even worse, an intrusion that does not exhibit anomalous behavior may not be detected, resulting in false negatives.

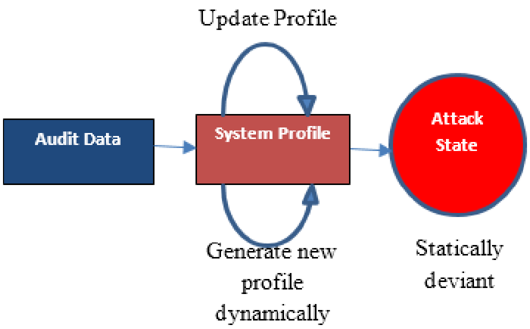

Figure-5: Anomaly Detection

Specification-based detection. [3] tries to combine the strengths of misuse and anomaly detection. It is based on deviations from normal behavior. However, in this case, the normal behavior is not defined by machine learning techniques and training.

It is based on manually defined specifications that describe what a correct operation is and monitors any behavior with respect to these constraints. In this way, legitimate but previously unseen behaviors will not cause a high false alarm rate, as in the anomaly detection approach. Also, since it is based on deviations from legitimate behaviors, it can still detect previously unknown attacks. On the other side, the development of detailed specifications by humans can be time-consuming and bare the inherent risk that certain attacks may pass undetected.

Caution must be taken when applying the anomaly detection technique in sensor networks. It is not easy to define what is a "normal behavior" in such networks, as they usually adapt to variations in their environment or according to other parameters, such as the remaining battery level. So, these legitimate changes of behavior may easily be mistaken from the IDS as intrusion attempts. Moreover, sensor networks cannot bear the overhead of automatic training, due to their low energy resources. Specification-based detection seems the most appropriate approach in this case, if someone may design appropriate rules to cover the broad range of attacks as possible.

Decision Making Techniques

IDS is also classified according to the decision making techniques that they use in order to detect and initiate a response to an intrusion attempt. This decision can be made either collaboratively or independently by the nodes.

Since the nature of sensor networks is distributed and most of the services provided require cooperation of other nodes, it's always better to carry intrusion in a cooperative manner. In this case, every node participates in intrusion detection and response by having an IDS client installed on them. Every WSN node is responsible for detecting intrusion attempts locally. In case an anomaly is detected by specific node with weak evidence, or if the evidence is inconclusive, then a cooperative mechanism is initiated with the neighboring nodes in order to take a network level intrusion detection action.

When designing a cooperative decision making mechanism for intrusion detection in sensor networks, one should consider the fact that a node can be compromised and hence, send falsified data to its neighbors trying to affect decision. So, one must be skeptical as to which nodes to trust. The fact that it is difficult for an adversary to compromise the majority of the nodes in a specific neighborhood can play an important role here. Moreover, a cooperative mechanism has to consider the bandwidth and energy resources of the nodes. The nodes cannot exchange security data and intrusion alerts without considering the energy that has to be spent for sending, receiving and processing these messages.

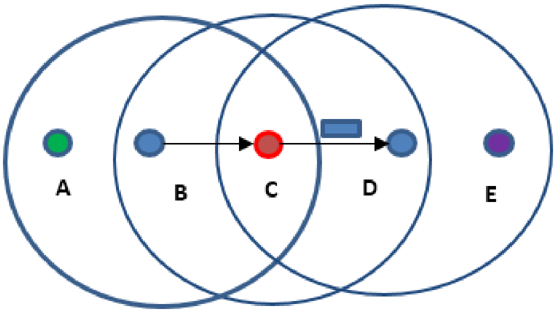

Figure 6: *Node B is selectively forwarding packets to node C. Node A promiscuously listens to node B's transmissions.*

In an independent decision-making system, there are certain nodes responsible to perform the decision-making functionality. Node will collect intrusion and anomalous activity evidences from other nodes and based on that it would take decision specific to network-level intrusions. The remaining of the nodes do not participate in this decision. In such architectures, the decision-making nodes may attract the interest of an attacker, since their elimination may leave the network undefended. Furthermore, the information that they process is limited, since it originates from specific nodes. Another disadvantage of such approaches is that, it restricts computation-intensive analysis of overall network security state to a few key nodes. Their special mission of processing the information from other nodes and deciding on intrusion attempts results in an extra processing overhead, which may quickly lead to energy exhaustion, unless and until different nodes are dynamically elected periodically

The Watchdog Approach

For implementing a network-based intrusion detection system in sensor networks, packet monitoring must take place in multiple nodes of the network, technique that can be used for packet monitoring, called the watchdog approach [4]. The watchdog approach relies on the broadcast nature of the wireless communications and presumes fact that sensors are densely deployed. Each packet transmitted in the network is not only received by the sender and the receiver, but also from a set of neighboring nodes within the sender's radio range. Normally these nodes would discard the packet, since they are not the intended receivers, but for intrusion detection this can be used as a valuable source of information.

Hence, a node can activate its IDS agent and monitor the packets sent by its neighbors, by overhearing them. However, this is not always adequate to draw safe conclusions on the behavior of the monitored node. There are certain concerns that arise in this case which will be highlighted by way of an example. In the setting shown in Figure 6, suppose that a packet should follow the path *A -> B -> C -> D*. Now, suppose that *C* is compromised and exhibits a malicious behavior, selectively dropping packets. There are three cases, arising from the wireless nature of communications, where having a node.

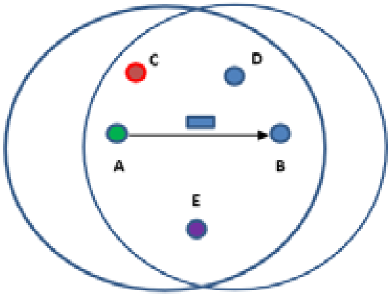

Figure-7: Nodes A, C, D and E can be watchdogs of the link A→! B.

B **monitoring node** ***C*** **cannot result in a successful detection of node** ***C*****:**

Node *C* forwards its packet and node *A* sends a packet to *B* at the same time. Then a collision occurs at *B*. Node *B* cannot be certain which packets caused this collision, so it cannot conclude on *C*'s behavior.

Node *C* forwards its packet to node *D* at the same time that node *E* makes a transmission. Then a collision occurs at *D*, which cannot be detected by *B*. Node *B* thinks that *C* has successfully forwarded its packet and therefore, *C* can skip retransmitting the packet, without being detected.

Node *C* forwards its packet to node *D* at the same time that *D* makes a transmission. Then a collision occurs at *D*. Again, node *B* thinks that *C* has successfully forwarded its packet, even though it never reached node *D*.

From the above cases we can conclude that only one watchdog is not always enough to detect an attack, so this approach should involve information from more nodes. Then these nodes could cooperate and exchange their partial views in order to draw their final conclusions.

Detecting certain attack is not enough to watch certain nodes and links For example, to detect selective forwarding, a watchdog should be able to hear packets arriving at a node and transmitted by that node. So, if we want to see whether a node *B* forwards packets sent by node *A*, we must activate a watchdog that resides within the intersection of *A*'s and *B*'s radio range. For example, in Figure, the nodes *A*, *C*, *D* and *E* can be watchdogs for the communication between *A* and *B*.

One could argue that the watchdog approach increases the energy consumption of the nodes, since they have to overhear packets not destined for them. However, let us note that in most radio stacks of today's sensor platforms each node receives packets sent by neighboring nodes anyway. They cannot know if a packet is addressed to them unless they receive it and check the destination field. So, the only overhead imposed to the nodes is any further processing of the packet

Existing Approaches

Several proposed architectures of intrusion detection systems already exist for Ad Hoc networks. The First scheme to be proposed was introduced by Zhang et al. [5], which is a distributed and cooperative IDS model, where every node in the network participates in the detection process. Another architecture, called LIDS [6] utilizes mobile agents on each of the nodes. These agents are used to collect and process data on remote hosts and transfer the results back to their home nodes, or migrate to another node for further investigation. Also based on mobile agents is the IDS proposed by Kochanski and Guha [7]. The agents are categorized as monitoring, decision-making and action agents. All nodes accommodate host-based monitoring agents but only a few nodes chosen by a distributed algorithm host agent with network monitoring and decision capabilities.

IDS architectures for Ad Hoc networks cannot be applied directly to sensor networks. The differences in the nature of the two kinds of networks impose different requirements, which forces us to design new solutions. First attempt to apply anomaly detection in sensor networks is presented by da Silva et al. [8]. According to the author's proposed algorithm, there are some monitor nodes in the network, which are responsible for monitoring their neighbors looking for intruders. These nodes listen to messages in their radio range and store certain message fields that might be useful to the rule application phase. The rules concern simple observations, such as:

Message sending rate must be within some limits,

Payload of a forwarded message should not be altered,

Retransmission of a message must occur before a defined timeout, and

Same message can only be retransmitted a limited number of times.

Then they try to detect some attacks, like message delay, repetition, data alteration, blackhole and selective forwarding. It is concluded from the paper that the buffer size to store the monitored messages is an important factor that greatly affects the false positives number. Given the restricted memory available in motes, it turns out that the detection effectiveness is kept to lower levels. that the buffer size to store the monitored messages is an important factor that greatly affects the false positives number. Given the restricted memory available in motes, it turns out that the detection effectiveness is kept to lower levels.

Similar approach is followed by Onat and Miri [9], where each node has a fixed-size buffer to store the packets received from neighbors and their corresponding arrival time and received power. If its power is not within certain limits, the packet is characterized anomalous. An intrusion alert is raised if the rate at which anomalous packets are detected over the overall rate at which packets are received is above a given threshold. In this way the authors claim that it is possible for a node to effectively identify an intruder impersonating a legitimate neighbor.

Roman et al. [10] propose an IDS architecture where all nodes are loaded with an IDS agent. This agent is divided into two parts: local agents and global agents. Local agents are active in every node and are responsible for monitoring and analyzing only local sources of information. Global agents are active at

only a subset of nodes. They are in charge of analyzing packets °owing in their immediate neighborhood. In order for the whole communication in the network to be covered by global agents, the global agents must be activated at the right nodes. For example, if clusters are used, the global agents will be activated at the cluster-heads. In case of a °at architecture, the authors propose another solution (called spontaneous watchdogs) that tries to activate only one global agent for a packet circulating in the network.

A completely different approach is presented by Anjum et al. [11], where the authors assume signature-based intrusion detection. This is the only work that takes a position against promiscuous monitoring and argues that detection should be based only on the analysis of packets that pass through a node. The problem then is to determine at which nodes the IDS modules should be placed, such that all the packets are inspected at least once. The proposed solution is based on the concepts of dominating set and minimum cut set and on the requirement that the nodes running the IDS module should be tamper resistant.

Theoretical model for WSN Sensor Mote Communication

We would present a systematic model and necessary conditions for intrusion detection, that address the impossible problems to solve. If the problem is not solved by our model, it is impossible to solve in weaker models which are closer to the reality[12].

As per our model sensor networks consists of a set $S = \{S1, Ss, …….\ Sn\}$ of *n* sensor nodes. Nodes communicate by sending messages over a unguided broadcast medium, i.e. message broadcasted may be received by many other nodes. Our assumption is due to broadcast nature and collision possibility, there is delay in the receipt of message

For sensor node *s*, group of node which may directly communicate is denoted by *N*(*s*). here we have done the following assumptions

Assumption for symmetric neighborhood relation, i.e., if *s* *N*(*s'*) then *s* *N*(*s*).

All the interconnected sensor node knows its 2-hop-neighborhood.

We assume a synchronous system model, i.e., sensor nodes are able to mea-sure time reliably using e.g., hardware clocks that run within a linear envelope of real time.

No assumptions for topology defined by all neighborhood

All the nodes behave according to the protocol used for connecting them with a path having the honest nodes only.

Model for Attacker

Our assumption is, an attacker may capture at most *t* nodes to launch an attack on the sensor network. We model this by allowing at most *t* nodes to behave in an arbitrary manner (Byzantine failure [13]).

Predicate *faulty*(*s*) on *S* is true if and only if (if) s is captured by the attacker.

We may define *honest*(*s*)   *Faulty*(*s*).

Initially attacker may follow the rules and regulation specified in protocol for a specific period of time so that it cannot be detected. But after some interval attacker may deviate or bypass the rules and regulations or protocol of faulty nodes to launch attack and under such condition we may say that attacker attacked. Here we focus on the case where *t* = 1, since its complex. Under these circumstances we may call the faulty node as the *source* of the attack, or the attacker, and use the predicate *source*(*s*) which is true if *s* is the attacker.

Alert module

Intrusion detection systems objective is to identify the attacker, and attacks detected by local IDS, detected details may be summarized and reported to alert module for alerting preconfigured node or admin about attack [14].

Whenever the alert module at node *s* notices something wrong in its neighborhood, the alert module simply outputs some set *D*(*s*) of *suspected sensors*, called the *suspected set*. The size of *D*(*s*) depends on the quality of the alert module. If |*D*(*s*)| = 1, then the sensor has identified the attacker.

Problem of Intrusion Detection

IDS is not only responsible to identify or detect the nodes being attacked but, it also includes identifying the source of an attack. Since we proposed, the cooperative intrusion detection, the process is triggered by an attack and the subsequent alerts by the local alert modules of the neighboring sensors. The overall process ends by the *expose of sensor node jointly*. More formally, the predicate *exposes* (*s0*) is true if node *s* exposes node *s0*. Intrusion detection problem may be defined as follows:

Definition (Intrusion Detection Problem (IDP)). *Find an algorithm that satisfies the following properties:*

If an honest node s exposes a node s', then s is in the alerted set and s' is the source, i.e., s   S : honest(*s*) *^ exposes*(*s'*) *=> A*(*s*) *^ source*(*s'*)

If the attacker attacks, then at most after some time all honest nodes in the alerted set expose some node.

Both the properties specified in IDP definition are equally important the first property refers to the aspect of *partial correctness*, i.e. it basically restricts the behavior of honest nodes to output something meaningful (if they output anything), if we won't give weightage to the first property it would allow implementations that output information which is not useful [16].

The second property refers to the aspect of *termination*: Algo is supposed to do something against intrusion, in case second property is not considered during implementation it won't result anything.

Conditions for Solving Intrusion Detection

Cooperative intrusion detection system uses to exchange the outputs of local alert modules, thereby narrowing the set of possible nodes that could be the attacker node. So we won't have any other way to learn something about the attacker apart from using their alert modules [15].

As an initial example, consider the case depicted in Figure-8 (a). Node *p* suspects the source *q*, i.e., *D* (*p*) = *{q}*. Being Byzantine, *q* can claim to output *D* (*q*) = *{p}*. Since *p* implicitly knows that it is honest, it will ignore the information provided by *q* and expose *q*, solving IDP.

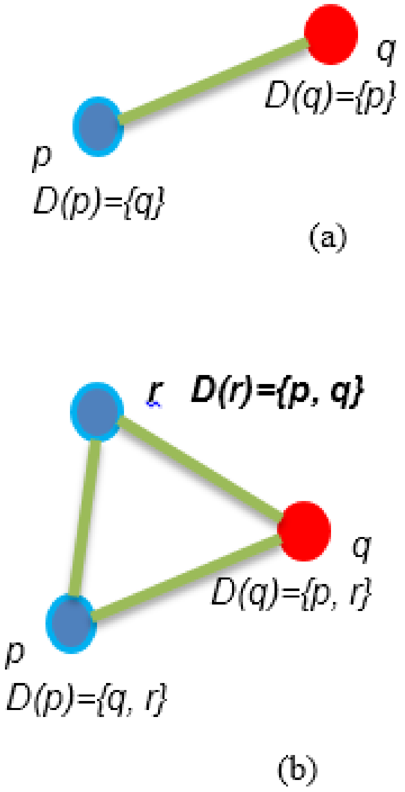

Figure-8: Different types of alerted neighborhoods. Sources of attacks are marked black. In case (a) the IDP is solvable, while in case (b) the IDP is not solvable.

Now consider a slightly updated example (see Figure 8 (b)). There three nodes *p*, *q*, and *r* all suspect each other (node *q* is the source). Every node occurs in exactly two suspect sets, *p* cannot distinguish node *r* from node *q*, if it only may use the suspect sets. Conclusion is, it is impossible to solve IDP in this case.

Generalizing these two examples, the question arises about general conditions for the solvability of the intrusion detection problem (IDP). In general, two types of conditions are interesting: Necessary conditions and sufficient conditions. A condition is *sufficient* for IDP if the truth of the condition implies that there exists an algorithm solving the IDP. A condition is *necessary* if the existence of an algorithm to solve IDP implies that the condition is true. In the following, we give necessary and sufficient conditions for solvability of IDP using a deterministic algorithm for *t* = 1 in our System model.

Solving IDP Conditions

We now give a sufficient condition for IDP solvability for *t* = 1 and deterministic algorithms. The intuition behind the condition is a generalization of the observation made in Figure 8(b): If the suspect sets about some node *s* are structurally equivalent to those of the source, then in general the problem is not solvable.

Formally, for a node *s* we define the set *AN*(*s*) to be the set of *alerted neighbors* of *s*, i.e.:

AN(*s*) = {*t* | *A*(*t*) ^ *t*   *N(s)}*

Furthermore, we define the set of alerted neighbors of *p* with respect to *q A˜ N* (*p; q*) to be the set of alerted neighbors of *p* without *q*, i.e.:

A~N (*p; q*) = *AN*(*p*) \ *{ g }*

As an example, consider Figure 8(b). Here, all three nodes are in alert mode and *AN*(*s*) = *D*(*s*). The value of *A˜N*(*b; a*) is the information content of *AN*(*b*) that is valuable to *a*. Since *a* itself knows that it is honest, it will exclude itself from the set *AN*(*b*), yielding *A~N*(*b; a*) = *{c}*

The Intrusion Detection Condition (IDC)

Definition (Intrusion Detection Condition (IDC)). *The intrusion detection condition (IDC) is defined as:*

$$\forall p, q\ \ S : source(q) => A{\sim}N(p,\ \ q)\ \ A{\sim}N(q,\ \ p)$$

IDC means that no other node has the same alerted neighborhood as the attacker. If *p* and *q* are neighbors, both are in each other's neighborhood and so they are always different. To exclude this case, we defined *A~N*. Note that if *p* and *q* are not neighbors, then IDC simplifies to:

$$\forall p, q\ \ S : source(q) => AN(p)\ \ AN(q)$$

Theorem1 (Sufficiency of IDC). *If t* = 1, *IDC is sufficient for ID, i.e., if IDC holds then IDP can be solved.*

Proof. Let all alerted nodes exchange their suspected sets. This is possible in our system model because each pair of honest nodes is connected by a path consisting of honest nodes, and communication is reliable.

The attacker can also go into alert mode. Moreover, it can send different suspected sets to different nodes. However, as we assume that all nodes know their 2-hop-neighborhood, the suspected set of the attacker may only contain its neighbors. Otherwise, the attacker's messages would be discarded [17].

Assume the suspected sets received by honest node *p*. since attacker is included in the suspected set of every honest node, and no honest node has the same alerted neighborhood as the attacker. Thus, if some node is suspected by more nodes than all other nodes, this node can be immediately identified as the attacker.

A more complicated case arises when there are two or more nodes which are suspected by the same number of nodes. This situation can arise, e.g., if the attacker also goes into the alert mode and accuses some of its neighbors[18].

We denote the attacker as *q*. Assume that there is a node *p q* which is suspected by the same number of nodes as *q*. How can a node *r* distinguish between *q* and *r*?

(1) If *p* = *r*, then *r* knows that it is honest, and exposes *q*.

(2) Consider *p r*. If all honest nodes suspect *p*, then the IDC does not hold. Thus, for some honest node *s* holds: *p* =2 *D*(*s*) and *q* 2 *D*(*s*). It follows that *q* is alerted and *p* 2 *D*(*q*), as the number of nodes which suspect *p* is the same as the number of nodes which suspect *q*.

Node *r* must now decide which of nodes *s* and *q* lies about their suspicion. We now show that there is an alerted node *v* which is not neighbor of *s*. Indeed, if all alerted nodes were neighbors of *s*, than *s* and *q* would have the same alerted neighborhood with respect to each other, which contradicts the IDC. Thus, node *r* has to find out which of the nodes *s* and *q* is not a neighbor of some alerted node. This is possible as all nodes know their 2-hop neighborhood. This node has to be honest, and the remaining node is identified as the attacker.

As an example, consider Figure-9 Nodes *s* and *r are* honest nodes and alerted. Node *p* is also honest, but not alerted. The attacker is node *q*, which is alerted. In this example, nodes *p* and *q* are both suspected by two nodes. How can node *r* distinguish the attacker?

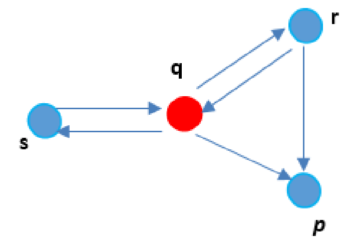

Figure 9: *Node q is the attacker, nodes s, r and q are alerted, while p is not alerted and it is marked white. x ! y means that node x suspects node y.*

D(*r*) = {*q,p*},

D(*q*) = {*p, r, s* }, *and D*(*s*) = {*q* }.

IDC holds here:

A~N(*p, q*) = Φ, *A~N*(*q, p*) = {*s*}

A~N(*r, q*) = {*p*}, *A~N*(*q, r*) = {*p, s*}

A~N(*s, q*) = Φ, *A~N*(*q, s*) = {*p*}

Nevertheless, node *p* collects two suspicions for each of *q* and *r*. Thus, either *q* or *s* is lying about their suspicions. However, nodes *r* and *s* are not neighbors, and therefore, *s* cannot be the attacker. (In this example, the node *v* from the proof is equal to *r*.)

The Neighborhood Conditions (NC)

Now the major question that arise are, what happens if IDC is not satisfied? Can IDP still be solved, or is IDC also a necessary condition for solving IDP? So we would show that IDC is not a necessary condition. We give an-other sufficient conditions for IDP solvability which can be valid in the network independently of the validity of the IDC.

Definition (Neighborhood Conditions). *The Neighborhood Conditions (NC) is having two conditions:*

NC1. *All neighbors of the attacker are alerted.*

NC2. *If two or more nodes are suspected by the majority of nodes, then all honest nodes suspected by the majority have non-alerted neighbors.*

Theorem 2 (Sufficiency of NC). *If the NC holds, i.e., NC*1 *and NC*2 *hold, then the IDP can be solved.*

*Proof. I*nformal reasoning is given here.

Let all alerted nodes exchange their suspected sets. If only one node is suspected by the majority of nodes, then this node is the attacker, as all neighbors of the attacker are alerted (*NC*1). If there are two or more nodes which are suspected by the majority, the nodes in the alerted set have to find out which of these nodes have non-alerted neighbors. According to *NC*2, only the attacker does not have non-alerted neighbors.

Necessary and Sufficient Conditions for Solving IDP

We now show that for the solvability of IDP either the IDC or the NC (i.e., NC1 and NC2) should be satisfied in the sensor network.

Theorem 3. *IDP can be solved using a deterministic algorithm if and only if the IDC or NC holds.*

Proof. As shown in Theorems 1 and 2, if IDC holds or if NC holds, then the intrusion detection problem can be solved. We now show that the IDC or the NC is necessary for the solvability of the IDP.

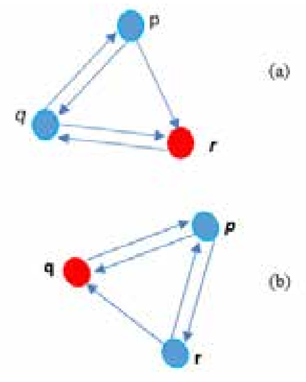

Figure 10: *Case (a): Node p suspects q and r, node q suspects p and r, node r is the attacker and suspect's q. IDC and* **NC2** *are not*

satisfied. Case (b): The suspicions remain as in case (a), but node q is the attacker. No algorithm for solving the IDP can distinguish between (a) and (b). Therefore, it is impossible to expose the attacker.

If the IDC does not hold and the NC does not hold, then the IDP cannot be solved.

Assume that the above claim is not true. That is, there exists a deterministic algorithm A that always exposes the attacker in case both the IDC and the NC do not hold. Consider Figure 10(a). The IDC does not hold there because

$$A\sim N(p, r) = A\sim N(r, p) = \{q\}.$$

Also NC does not hold, because NC2 does not hold: The attacker r and the honest node q are suspected by two nodes, but q does not have any non-alerted neighbors. In this case, the algorithm A should expose r. However, the situation in Figure 10(b) is exactly the same as in (a) from A's point of view. The suspicions remain the same, the topology also does not change. Thus, there is no additional information to help A to distinguish between situations (a) and (b). However, A should be able to distinguish between (a) and (b) and to expose r or q accordingly. It follows that A does not exist.

In IDP, the honest nodes have to jointly expose the attacker. That is, they have to reach agreement on the attacker's identity. At First glance, its similar to Byzantine Agreement, where the nodes have to reach agreement on their inputs. Nevertheless, we show that these two problems cannot be reduced to each other. In some cases, Byzantine Agreement can be solved, whereas Intrusion Detection is not solvable. On the other hand, sometimes Intrusion Detection is solvable, whereas Byzantine Agreement is not.

Consider Figure 11(a). Here, the three nodes p, q and r are connected, q is the attacker. It participates in the protocol and suspects both p and r. The honest nodes, on the other hand, both suspect q. In this case, Intrusion Detection is trivially solvable. However, Byzantine Agreement for three participants with $t = 1$ cannot be solved. In Figure 11(b), all nodes suspect each other. IDC does not hold, as nodes s and q have the same alerted neighborhood with respect to each other. NC also does not hold, as no node has non-alerted neighbors. Thus, Intrusion Detection is not solvable. However, Byzantine Agreement for $t = 1$ can be solved here.

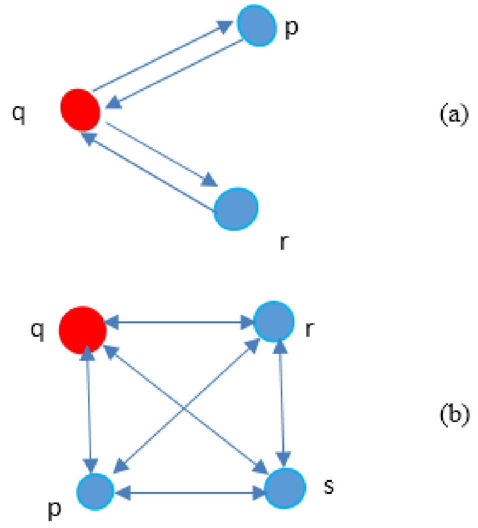

Figure 11: *Byzantine Agreement and Intrusion Detection cannot be reduced to each other. Case (a): Honest nodes p and r both suspect only the attacker q, thus Intrusion Detection can be solved, but Byzantine Agreement is not solvable. Case (b): Intrusion detection cannot be solved; Byzantine Agreement is solvable.*

Conclusions

Intrusion Detection System for sensor networks must have to identify and locate its detection agents inside all the nodes. It would provide partial views of attack, that can be combined through a cooperative mechanism and provide the nodes with strong evidence of the attack. We made an attempt to formalize the problem of intrusion detection in sensor networks, and showed the benefits and theoretical limitations of the cooperative approach to intrusion detection system. We have presented necessary and sufficient conditions for successfully exposing the attacker under a general threat model. For the proofs, we used a strict theoretical model, which can be weakened to reflect the conditions in realistic sensor networks.

REFERENCES:

1. K. Ilgun, R. A. Kemmerer, and P. A. Porras. State transition analysis: A rule-based intrusion detection approach. Software Engineering, vol. 21(3):pp. 181–199, 1995.
2. H.S.Javitzand A.Valdes. The NIDE Sstatisticalcomponent: Description and justification. Annual report, Computer Science Laboratory, SRI International, Menlo Park, CA, March 1994.C. Ko, P. Brutch, J. Rowe, G. Tsafnat, and K. N. Levitt. System health and intrusion monitoring using a hierarchy of constraints. In RAID '00: Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection, pp. 190–204. 2001.
3. S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom '00), pp. 255–265. August 2000
4. Y. Zhang, W. Lee, and Y.-A. Huang. Intrusion detection techniques for mobile wireless networks. Wireless Networks, vol. 9(5):pp. 545–556, 2003
5. P. Albers, O. Camp, J.-M. Percher, B. Jouga, L. M´e, and R. Puttini. Security in ad hoc networks: A general intrusion detection architecture enhancing trust based approaches. In Proceedings of the First International Workshop on Wireless Information Systems (WIS-2002), pp. 1–12. April 2002.
6. O. Kachirski and R. Guha. Intrusion detection using mobile agents in wireless ad hoc networks. In KMN '02: Proceedings of the IEEE Workshop on Knowledge Media Networking, p. 153. 2002
7. A. P. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. C. Wong. Decentralized intrusion detection in wireless sensor networks. In Proceedings of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks (Q2SWinet '05), pp. 16–23. ACM Press, October 2005
8. I. Onat and A. Miri. An intrusion detection system for wireless sensor networks. In Proceeding of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, vol. 3, pp. 253–259. Montreal, Canada, August 2005.
9. R. Roman, J. Zhou, and J. Lopez. Applying intrusion detection systems to wireless sensor networks. In Proceedings of IEEE Consumer Communications and Networking Conference (CCNC '06), pp. 640– 644. Las Vegas, USA, January 2006.
10. F. Anjum, D. Subhadrabandhu, S. Sarkar, and R. Shetty. On optimal placement of intrusion detection modules in sensor networks. In BROADNETS '04: Proceedings of the First International Conference on Broadband Networks (BROADNETS'04), pp. 690–699. 2004.
11. S.Ganeriwal, S. Capkun, C.-C.Han, and M.B.Srivastava. Securetime synchronization service for sensor networks. In WiSe '05: Proceedings of the 4th ACM workshop on Wireless Security, pp. 97–106. ACM, New York, NY, USA, 2005.
12. F. C. G¨artner. Byzantine failures and security: Arbitrary is not (always) random. Technical Report IC/2003/20, Swiss Federal Institute of Technology (EPFL), 2003
13. I.KrontirisandT.Dimitriou. Secure network programming in wireless sensor networks. In Y. Xiao and Y. Pan, editors, Security in Distributed and Networking Systems, chap. 12, pp. 289–310. World Scientific Publishing Co., 2007.
14. I. Krontiris, T. Dimitriou, and F. C. Freiling. Towards intrusion detection in wireless sensor networks. In Proceedings of the 13th European Wireless Conference. Paris, France, April 2007.
15. I. Krontiris, T. Dimitriou, T. Giannetsos, and M. Mpasoukos. Intrusion detection of sinkhole attacks in wireless sensor networks. In M. Kutylowski, J. Cichon, and P. Kubiak, editors, Algorithmic Aspects of Wireless Sensor Networks – ALGOSENSORS, vol. 4837 of Lecture Notes in Computer Science, pp. 150–161. Springer, 2007.
16. M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. Journal of the ACM, vol. 27(2):pp. 228–234, 1980
17. Mishra, D. P., & Kumar, R. (2015). Vision of Hybrid Security Framework for Wireless Sensor Network. Indian Journal of Applied Research, 5, 167.
18. Mishra, D. P., & Kumar, R. (2016). Analysis of Wireless Sensor Networks Security Solutions and Countermeasures. Journal of Scientific and Technical Research, 8, 10.