Trusting Anomaly and Intrusion Claims for Cooperative Distributed Intrusion Detection Schemes of Wireless Sensor Networks

Riaz Ahmed Shaikh, Hassan Jameel, Brian J. d'Auriol, Sungyoung Lee, Young-Jae Song
Dept. of Comp. Eng., Kyung Hee University, Global Campus, Suwon, Korea.
{riaz, hassan, dauriol, sylee}@oslab.khu.ac.kr, yjsong@khu.ac.kr

Heejo Lee
Dept. of Comp. Sci. & Eng., Korea University, Seoul, Korea.
heejo@korea.ac.kr

Abstract

Any unidentified malicious nodes in the network could send faulty anomaly and intrusion claims about the legitimate nodes to the other nodes. Verifying the validity of such claims is a critical and challenging issue that is not considered in existing cooperative-based distributed anomaly and intrusion detection schemes of wireless sensor networks. In this paper, we propose a validation algorithm that addresses this problem. This algorithm utilizes the concept of intrusion-aware reliability that helps to provide adequate reliability at the modest communication cost.

1 Introduction

Many anomaly and intrusion detection schemes have been proposed for wireless sensor networks (WSNs) e.g. [2, 6, 8, 4, 5], but those schemes mainly focus on the detection of malicious or faulty nodes. All those anomalies and intrusion detection schemes (IDS) which are cooperative in nature e.g. [2, 6, 4] need to share anomalies or intrusion claims with the other node(s). However those schemes are unable to assure that the report or claim received by the other node(s) is really sent by the trusted node(s). So the problem here is: any unidentified malicious node(s) in the network could send faulty anomaly and intrusion claims about the legitimate node(s) to the other node(s). Verifying the validity of such claims is a critical issue that is not considered in existing cooperative-based distributed anomaly and IDS schemes of WSNs [3].

In this paper, we propose first simple and easy to implement an intrusion-aware validation algorithm that provides a mechanism for trusting anomalies and intrusion claims sent by any unidentified malicious node(s). This algorithm consists of two phases: consensus phase and decision phase. Although the consensus approach is widely used in distributed computing domain to solve many problems like fault-tolerance [1], but here we used this approach with variation to solve problem of trusting anomalies and intrusion claims. In consensus phase, we uniquely introduce an intrusion-aware reliability concept that helps to provide an adequate reliability at a modest communication cost. In the decision phase, a node will make the decision regarding validation and invalidation of a claim based on the result of consensus phase.

The rest of the paper is organized as follows: Section 2 describes related work. Section 3 discussed the network model, assumptions and definitions. Section 4 describes the proposed validation algorithm. Section 5 provides the analysis and evaluation of proposed algorithm and Section 6 concludes the paper and highlighted some future work.

2 Related Work

Intrusion Detection Schemes (IDS) have often been categorized into two types: Signature-based IDS and Anomaly-based IDS. The signature-based IDS schemes detect intrusions based on the attack's signature such as specific byte sequence in the payload or specific information in the header fields like sender address, last hop address etc. On the other hand, the anomaly-based IDS (mostly implemented via statistical approach), first determines the normal network activity and then checks all traffic that deviates from the normal and marks it as anomalous. From an architectural point of view, IDS schemes are further categorized into three categories: centralized, distributed and hybrid. In the centralized approach, single designated node monitors the whole network. In the distributed approach, every node or a group of nodes monitor the network. This approach is further classified into cooperative and uncooperative distributed approaches. In the cooperative distributed

978-0-7695-3398-8/08 $25.00 © 2008 IEEE
DOI 10.1109/ICYCS.2008.489

Table 1. Summarization of proposed Anomalies and IDS schemes of WSNs

		V. Bhuse et al. [2]	W. Du et al. [6]	C. E. Loo et al. [8]	V. Chatzigiannakis et al. [4]	A.P.R. da Silva et al. [5]
Classification	Technique	Signature-based	Statistical-based	Statistical-based	Statistical-based	Statistical-based
	Architecture	Distributed & cooperative	Distributed & cooperative	Distributed & uncooperative	Hybrid	Distributed & uncooperative
Specifications	Installation of IDS	On every sensor node	On every sensor node	On every sensor node	On every primary node of a group	Special monitor nodes in network
	IDS Scope	Multilayer (Appl., Net., MAC & Phy.)	Application layer	Network Layer	Application layer	Multilayer (Appl., Net., MAC & Phy.)
	Attacks detects	Masquerade attack, and forged packets attacks	Localization anomalies	Routing attacks e.g. Periodic error route attack, active & passive sinkhole attack	Correlated anomalies/ attacks (invalid data insertion)	Worm holes, data alteration, selective forwarding, black hole, & jamming
Network	Sensor node	Static / Mobile	Static	Static / Mobile	Static / Mobile	Static
	Topology	Any	Any	Any	Cluster-based	Tree-based

approach, every node or a group of nodes exchanges information about the anomalies and intrusions in order to detect collaborative intrusion attacks. On the contrary, in the uncooperative distributed approach, nodes do not share information about anomalies and intrusion with each others. In the hybrid approach, every group has one selected primary node responsible for monitoring and detecting anomalies and intrusions. Once the information is gathered, it is forwarded to the central base station which calculates the impact of those anomalies and intrusions on the whole network. So this approach is, by default, cooperative in nature.

Intrusion detection schemes are not in itself the main focus of this paper. However in order to give brief overview of those, we have summarized the existing proposed anomalies and IDS schemes of WSNs in Table 1, in which [2], [6], and [4] are distributed and cooperative in nature. Brief description of some of the proposed schemes is given below.

V. Bhuse et al. [2] have proposed different lightweight techniques for detecting anomalies for various layers such as application, network, MAC and physical. The main advantage of proposed techniques is the low overhead that makes them energy efficient. This is due to the fact that they reuse the already available system information (e.g. RSSI values, round trip time etc.) which are brought forth at various layers of network stack.

V. Chatzigiannakis et al. [4] have proposed an application level anomaly detection approach that fuses data (comprises of multiple metrics) gathered from different sensor nodes. In the proposed scheme, the authors have applied Principal Component Analysis (PCA) to reduce the dimensionality of a data set. So this approach will help to detect correlated anomalies/attacks that involve multiple groups of sensors.

W. Du et al. [6] have proposed a localization anomalies detection (LAD) scheme for the wireless sensor networks. This scheme takes the advantage of the deployment knowledge and the group membership of its neighbors, available in many sensor network applications. This information is then utilized to find out whether the estimated location is consistent with its observations. In case of an inconsistency LAD would report an anomaly.

Recently Q. Zhang et al. [13] have proposed a nice application-independent framework for identifying compromised nodes. This framework is based on alerts generated by specific intrusion detection system. The authors have adopted a centralized approach and used a simple graph theory. However, this scheme has some limitations such as: it provides some late detection of compromised nodes. Because detection process will always start at the end of each time window. If the size of the time window is large (as authors have mentioned the example of one hour) then in that case it is quite possible that an adversary would achieve its objective during that time window. If the time window is small then result may not be accurate. Also, detection accuracy is mainly dependent on the size of the network density. If the network size decreases then the detection accuracy will also decreases.

3 Network Model, Assumptions and Definitions

3.1 Network Model and Assumptions

Sensor nodes are deployed in an environment either in a random fashion or in a grid fashion which are organized in any form of topology (e.g. cluster-based [12] etc.). Any data-centric (e.g. directed diffusion [7] etc.) or address-centric (e.g. AODV [9] etc.) routing scheme could be used.

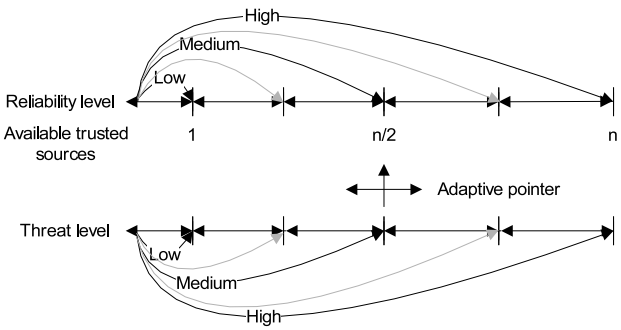


Figure 1. Intrusion-aware reliability mode concept

We assumed that any cooperative-based distributed anomaly or IDS is already deployed in the WSNs: which forward claims to the other node(s) whenever it detects some anomalies or intrusions. The malicious node must fall into the radio range of the monitoring node. And the node (who received the claim from the monitoring node) has the knowledge about the neighboring nodes of the monitoring and malicious nodes. We have also assumed that the multiple sensor nodes in a neighborhood can sense the same anomaly/intrusion. We also assumed that all information is exchanged in a secure encrypted manner. For this purpose, every monitoring node share a unique secret key [10] with the node who received the claims.

3.2 Definitions

A legitimate node which is compromised by an adversary is called a malicious node. So the malicious node could performed malicious activities like dropping and fabrication of packets etc. Also in order to hide the presence of the adversary, a malicious node could also perform all the activities like normal nodes do such as monitoring, ciphering of data, forwarding of packets etc.

Reliability means the confidence level on a certain decision. It can simply be categorized into three levels: 1) low, 2) medium, and 3) high. In the low reliability mode, validation is based on the confirmation from any one available reliable source. In the medium reliability mode, validation is based on the confirmation from half of the available reliable sources. In the high reliability mode, validation is based on the confirmation from all of the reliable sources. In more generic way, reliability level (R_L) is define as:

$$R_L = m \ ; \quad m \leq n \tag{1}$$

where n represents the total number of available nodes, and m represents the number of consulting nodes. However, in order to achieve more flexibility and adaptability, we have adopted intrusion-aware reliability mode concept, in which validation is based on the level of a threat of an anomaly or intrusion. This approach will also reduce the communication cost as described in Section 5.1. Threats could also be categorized into low, medium, high or other. Depending on the level of the threat, intrusion-aware reliability mode is set to low, medium, high or other as shown in Figure 1.

4 Proposed Algorithm

Our proposed intrusion-aware validation algorithm works in two phase: First phase is a consensus phase, which initiates when the node receives the claim from the monitoring node and second phase is a decision phase, which initiates just after the end of first phase.

Algorithm 4.1 Phase 1: Consensus Phase

```
1:  Received Claim Packet (ID_sender, ID_mal,detail);
2:  if ID_sender != malicious and ID_mal is new then
3:     N_s = GetNeighorList(ID_sender);
4:     N_m = GetNeighorList(ID_mal);
5:     N_sm = N_s ⋂ N_m;
6:     N_t = Eliminate-Known-Malicious-Nodes(N_sm);
7:     if N_t ≠ φ then
8:        if ThreatLevel(detail) is Low then
9:           Send conf-req-pkt(rand(N_t), ID_mal,det);
10:       else if ThreatLevel(detail) is Medium then
11:          for i = 1 to len(N_t)/2 do
12:             Send conf-req-pkt(rand(N_t), ID_mal,det);
13:          end for
14:       else
15:          for i = 1 to len(N_t) do
16:             Send conf-req-pkt(ID_i, ID_mal,det);
17:          end for
18:       end if
19:    end if
20: else
21:    Update Record;
22: end if
```

4.1 Phase 1 (Consensus Phase)

A claim packet contains three types information: 1) identity of the sender node (ID_{sender}), 2) identity of the malicious node (ID_{mal}) and 3) details about anomaly or intrusion. Whenever a designated node receives the claim packet it first checks two things: 1) is the sender malicious? and 2) identity of a new malicious node is already declared as a malicious node or not (Algorithm 4, Line 1:2). If not then the node will first get the common neighborhood list (N_{sm}) of the sender and malicious nodes respectively (Line

3:5). After that the node will perform filtering by eliminating any known malicious node(s) from that list (Line 6). Based on the threat level, confirmation request packet(s) is forwarded to the randomly selected node(s) from the N_t list (Line 7:19). For example, if the threat is of low level, then the conformation request is forwarded to the one randomly selected trusted node from the list N_t (Line 8:9). If the threat is of medium level, then the conformation request packet is forwarded to the half of the randomly selected trusted nodes from the list N_t (Line 10:13). If the threat is of high level, then the conformation request packet is forwarded to all the trusted nodes contains in the list N_t (Line 14:17). If the information about the malicious node is already present (line 20) then the node will just update its old record (Line 21).

4.2 Phase 2 (Decision Phase)

Once the confirmation request packet(s) is forwarded to the particular node(s) then the phase 2 of the validation algorithm is triggered. In this phase algorithm will first wait for the confirmation response packets until Δt time, where Δt is calculated as:

$$\Delta t = 2\left[2t_{prop} + t_{proc}\right] \tag{2}$$

Here, t_{prop} is the propagation time between the requester and farthest responder (in terms of hops or geographical location) among nodes where the request packets were forwarded. The t_{proc} is the estimated processing time of the request at the responder end.

A node will expect three types of responses (r) from the nodes where confirmation request packets were forwarded:

$$r_{i,j} = \begin{cases} 1 & if \quad \text{agree with claim} \\ 0 & if \quad \text{don't know} \\ -1 & if \quad \text{not agree with claim} \end{cases} \tag{3}$$

where $r_{i,j}$ represents that the node i received the response packet from the node j and $j \in N_t$. A node i will make the decision (D) about the validity and invalidity of the claim based on the following rule:

$$D_i = \begin{cases} \text{validate} & iff \quad \sum_{j=0}^{n_{res}} r_{i,j} > 0 \\ \text{no consensus} & iff \quad \sum_{j=0}^{n_{res}} r_{i,j} = 0 \\ \text{invalidate} & iff \quad \sum_{j=0}^{n_{res}} r_{i,j} < 0 \end{cases} \tag{4}$$

where n_{res} represents the total number of the response packets received by the node i in response to the number of the request packets (n_{req}). Here $0 \geq n_{res} \leq n_{req}$.

If the claim is found to be invalidate then the sender of the claim will declare as a malicious node. That helps to provide protection against any possible security threats such as flooding, or denial of service attacks etc.

If no consensus builds then the algorithm will make the decision based on its mode that is set by the administrator. There are two types of modes: aggressive and defensive. If the algorithm is set as an aggressive mode then the node will validate the claim and if it is set as a defensive mode then the node will invalidate the claim.

Table 2. Communication Overhead of reliability modes

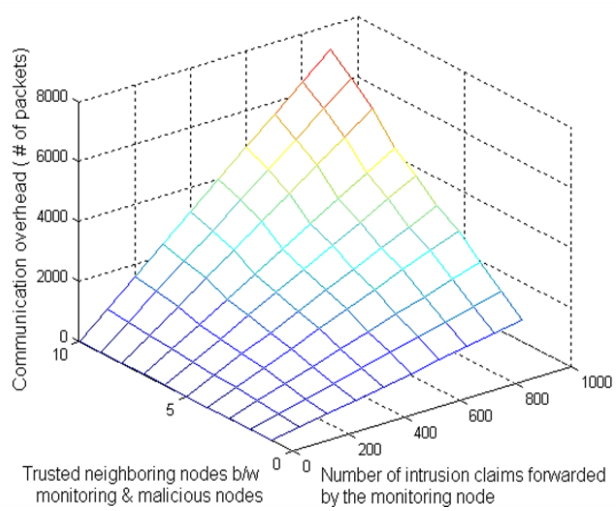
	Cost
Low	$2I_c$
Medium	$m_t I_c$
High	$2m_t I_c$
Intrusion-aware	$2I_l + (I_m + 2I_h)m_t$

5 Evaluation

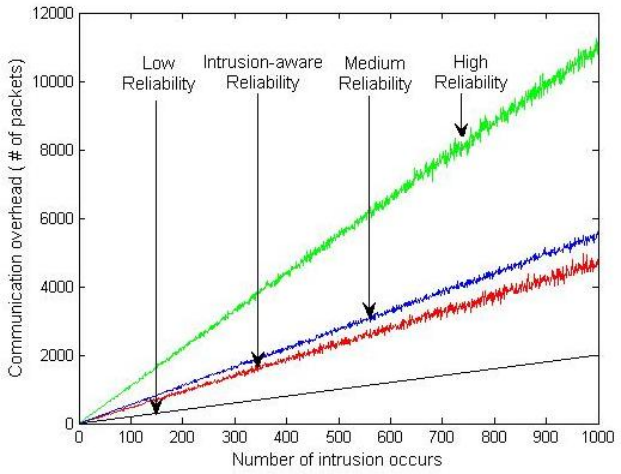
5.1 Communication Overhead

Communication overhead of the validation algorithm is depended on three factors: 1) total number of intrusion claims (I_c), 2) number of common trusted neighboring nodes, and 3) threat level of intrusion or anomaly. Table 2 shows the communication overhead, in which m_t represents the average number of trusted common neighboring nodes between the monitoring and malicious nodes and I_l, I_m, and I_h represents the total number of low, medium and high intrusion level threats respectively. Here $I_c = I_l + I_m + I_h$.

Figure 2 shows the average communication overhead (1000 simulation runs) of the proposed validation algorithm. During the simulation, different levels (low, medium or high) of threats of anomalies and intrusions occur randomly. Figure 2(a) shows the effect of average number of common trusted neighboring nodes (between the monitoring and the malicious nodes) m_t and the total number of intrusions I_c occurs in the network. It shows that as the number of m_t or I_c increases the communication overhead of the validation scheme also increases linearly. Figure 2(b) shows the comparison between the four different levels of the reliability modes. In the simulation, each monitoring node has random number of common trusted neighboring nodes. This figure shows that the intrusion-aware reliability mode introduces less communication overhead then the medium and high level reliability modes. And at a modest communication cost it provides adequate reliability required by the nature of the intrusion claim.



(a) Effect of m_t and I_c



(b) Comparison from reliability mode perspective

Figure 2. Average communication overhead of validation algorithm after 1000 simulation runs in which different levels of intrusions occurs randomly.

5.2 Reliability

If we assumes that the responding node have equal probability of sending any one out of three possible responses (agree, disagree and don't know) then the total probability (P_c) of a algorithm to reach at the consensus state (validate or invalidate) is:

$$P_c = \frac{N_c}{K^{n_{res}}} \tag{5}$$

where N_c represents the number of nodes reaching a consensus and K represents the number of possible outcomes (agree, disagree and don't know) produces by the node. If

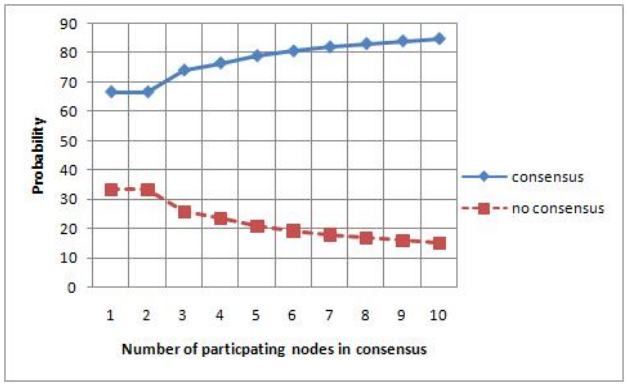


Figure 3. Probability of reaching at consensus and no consensus state

the probability distribution is not uniform between possible outcomes, then the total probability (P_c) of a algorithm to reach at the consensus state (validates or invalidate) is:

$$P_c = \sum_{m=1}^{M} (\Pi_{i=1}^{n_{res}} PMF_i(S_m(i))) \times \delta(m) \quad \text{where } M = K^{n_{res}} \tag{6}$$

where $\delta(m)$ is one if m node reaches the consensus, otherwise it will be zero. PMF_i is the probability mass function that captures the probability distribution of the symbol produced by the node i. $S_m(i)$ is the i^{th} symbol in the m^{th} node result. More details and derivation of these two probability equations are given in [11].

Figure 3, shows the simulation result about the probability of reaching at consensus (validate or invalidate) of our validation algorithm. It shows that as the number of participating nodes increases in the consensus process, the probability of a reaching at some consensus also increases linearly.

6 Conclusion and Future Work

Existing cooperative-based distributed anomaly and intrusion detection schemes of WSNs does not provide assurance that the reports/alerts/claims received by the other node(s) are really send by the trusted legitimate node(s). Therefore, in this paper we have proposed first validation algorithm for trusting anomalies and intrusion claims. This algorithm uses the concept of intrusion-aware reliability parameter that helps to provide adequate reliability at a modest communication cost.

The proposed work is still in preliminary stage and is based on a few strict assumptions, such as multiple nodes can sense same anomaly/intrusion. In practical, it is quite possible that only one node can detect some specific

anomaly/intrusion. In this case, our scheme will not be able to validate the claim. Therefore, more work is needed to make proposed scheme further flexible. Also, our proposed scheme should needed to be evaluate from the security resiliency perspective.

Acknowledgment

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)(IITA-2008-C1090-0801-0002) and by the MIC(Ministry of Information and Communication), Korea, Under the ITFSIP(IT Foreign Specialist Inviting Program) supervised by the IITA (C1012-0801-0003). Also, this work is financially supported by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energy (MOCIE) and the Ministry of Labor (MOLAB) through the fostering project of the Lab of Excellency.

References

[1] M. Barborak, A. Dahbura, and M. Malek. The consensus problem in fault-tolerant computing. *ACM Computing Surveys*, 25(2):171–220, 1993.

[2] V. Bhuse and A. Gupta. Anomaly intrusion detection in wireless sensor networks. *Journal of High Speed Networks*, 15:33–51, 2006.

[3] V. S. Bhuse. Lightweight intrusion detection: A second line of defense for unguarded wireless sensor networks. *PhD thesis, Dept. of Comp. Sci., Western Michigan University, USA*, 2007.

[4] V. Chatzigiannakis and S. Papavassiliou. Diagnosing anomalies and identifying faulty nodes in sensor networks. *IEEE Sensors Journal*, 7(5):637–645, 2007.

[5] A. P. R. da Silva, M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz, and H. C. Wong. Decentralized intrusion detection in wireless sensor networks. In *Proc. of the 1st ACM Int. workshop on Quality of service & security in wireless and mobile networks (Q2SWinet'05)*, pages 16–23, Canada, oct 2005.

[6] W. Du, L. Fang, and N. Peng. LAD: Localization anomaly detection for wireless sensor network. *Journal of Parallel and Distributed Computing*, 66:874–886, 2006.

[7] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. on Networking*, 11(1):2–16, Oct 2003.

[8] C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami. Intrusion detection for routing attacks in sensor networks. *Int. Journal of Distributed Sensor Networks*, 2:313–332, 2006.

[9] C. Perkins, E. Belding-Royer, and S. Da. Ad hoc on-demand distance vector (AODV) routing. In *RFC 3561*, Canada, Jul 2003.

[10] R. A. Shaikh, S. Lee, M. A. U. Khan, and Y. J. Song. LSec: Lightweight security protocol for distributed wireless sensor network. In *11th IFIP Int. Conf. on Personal Wireless Comm., LNCS 4217*, pages 367–377, Albacete, Spain, Sept. 2006.

[11] S. Yacoub, X. Lin, and J. Burns. Analysis of the reliability and behavior of majority and plurality voting systems. In *Technical Report (HPL-2002-118), Information Infrastructure Laboratory, HP Laboratories Palo Alto*, Apr, 2002.

[12] O. Younis and S. Fahmy. HEED: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks. *IEEE Trans. on Mobile Computing*, 3(4):366–379, Oct 2004.

[13] Q. Zhang, T. Yu, and P. Ning. A framework for identifying compromised nodes in wireless sensor networks. *ACM Trans. Inf. Syst. Secur.*, 11(3):1–37, 2008.