An Anomaly Detection Algorithm for Detecting Attacks in Wireless Sensor Networks[1]

Tran Van Phuong, Le Xuan Hung, Seong Jin Cho, Young-Koo Lee and Sungyoung Lee

Computer Engineering Dept. Kyung Hee University
449-701 Suwon, Republic of Korea
{tvphuong, lxhung, sjcho}@oslab.khu.ac.kr, yklee@khu.ac.kr, sylee@oslab.khu.ac.kr

Abstract. Wide applications of Wireless Sensor Networks also make them more interesting to adversaries. WSNs' protocols are designed without security in mind so they are susceptible to many types of attacks. Some preventive mechanisms are deployed to protect WSNs but they are not enough. Thus, WSNs need an Intrusion Detection System (IDS) to detect intrusion of adversaries to response and diminish the damage. In this paper, we propose an algorithm for detecting a series of attacks in WSNs by applying Cumulative Sum (CuSum) algorithm to detect anomalies. We also show that our algorithm is very light-weight so it fits the demands and restrictions of WSNs.

1. Introduction

Wireless sensor networks (WSNs) consisting of thousands of sensor nodes have many potential applications nowadays from temperature, light monitoring in a smart house to detecting enemy's movement in a battle field. In most cases, sensor networks are deployed in open and unprotected environments so it is very attractive to adversaries. There are many ways adversaries can use to attack sensor networks [2, 3]. Although some preventive mechanisms were proposed and installed, they do not guarantee the security of sensor networks one hundred percent. Thus, it is necessary to have some mechanisms of intrusion detection as a second protecting wall to prevent intruders from causing damages to the networks.

A lot of work has been done on Intrusion Detection System (IDS) for traditional wired networks so far. However, it is not appropriate to apply directly IDSs in wired networks into sensor networks because of unique characteristics of sensor networks. From the intrusion detection viewpoint, the main challenges in sensor networks are

[1] This work was supported by IITA Professorship for Visiting Faculty Position in Korea from Ministry of Information and Communications. Dr. Young-Koo Lee is the corresponding author.

their flexible network topologies, lack of concentration points where traffic can be analyzed and the most important, sensor resource constraints. Sensor nodes are designed to be small and inexpensive so they have limited capabilities such as limited computational power, memory and energy. Thus, all security services for sensor networks must be designed with these constraints in mind. Some intrusion detection mechanism has been published however their performances is very limited, either in resource usage or in effectiveness.

In this paper, we proposed another approach to detect intrusion in sensor networks by using Cumulative Sum algorithm (CuSum) to detect anomalies based on statistical information of packets in the networks. Three features of the network are monitored including: the number of incoming packets, the number of outgoing packets and the number of collisions related to each node. The network is considered as under attacks if any abrupt change of one of these features is reported. The most important things of our contribution lie in the simplicity, low computation overhead and the high effectiveness of the proposal algorithm.

The remainder of the paper is organized as follows. Section 2 briefly discusses some related work. Section 3 mentions about background study in this context. Section 4 discribles the proposal algorithm. In section 5 we present a simple model using the algorithm to detect intrusion detection in sensor networks. Finally, in section 6, we discuss and summarize our results and future work.

2. Related works

Intrusion Detection is not a new research issue in the broad area of security. A lot of work has been done so far for Intrusion Detection in wired traditional networks [7, 8, 9, 10]. However, restrictions of WSNs make the direct application of these solutions inappropriate.

Recently, intrusion detection in WSNs is getting more and more attention of researchers. However, there is a limited number of papers about algorithms to detect attacks in WSNs so far. One of them is the “temporal packet leashes” algorithm used to detect wormhole attack [6]. In this approach, the time needed to transfer a packet between each pair of neighbors will be calculated. A larger than usual time will indicate a wormhole attack. This approach requires strictly the clock synchronization between nodes in the network which is not easy to obtain in WSNs.

In [5], the author proposed two statistical approaches to detect wormhole attack in WSNS. The first one called Neighbor Number Test bases on a simple assumption that a wormhole will increase the number of neighbors of the nodes in its radius. The base station will get neighborhood information from all sensor nodes, computes the hypothetical distribution of the number of neighbors and uses statistical test to decide if there is a wormhole or not. The second one called All Distance Test detects wormhole by computing the distribution of the length of the shortest paths between all pairs of nodes. In these two algorithms, most of the workload is done in the base station to save sensor nodes’ resources. However, one of the major drawbacks is that they do not pinpoint the location of wormhole which is necessary for a successful defense.

A rule-based algorithm was proposed in [16] to detect anomalies in WSNS. Monitor nodes will check traffic of their neighbors and compare to some predefined rules. If a rule is not satisfied, a failure is accumulated. An anomaly is reported if the number of failure is greater than an expected value which is calculated dynamically by the monitor node. However, some rules are not easy to implemente and resource consuming. More important, the detection effectiveness and accuracy depend on the buffer size which is strictly limited in WSNS. A similar algorithm proposed in [17] has the same drawbacks.

3. Background

For better understanding of our algorithm, we are going to present briefly a background for our study including detection techniques and common types of attacks in WSNs

3.1 Anomaly Detection

All of the intrusion detection techniques are classified into one of two methodologies: misused detection or anomaly detection. Misused detection techniques, sometimes refers to as signature-based detection techniques, look for behavior that matches a known attack scenario by analyzing the information in the network, compareing it to a large database of known attacks (signatures). Any new attack which is not in the database can not be detected so the database must be kept up to date, which is not easy to do in sensor networks. Anomaly detection tecniques look for behavior that deviates from normal system activities. These techniques do not require knowledge of know attacks and can detect new types of intrusion which is considered more suitable for sensor networks.

The key question in anommaly detection techniques is how to distinguish anomalies from normal. Which factors of behavior used to know whether one behavior is normal or not is the most important thing in an anomaly detection system. Some systems use the distribution of commands that users used in their session, some use statistics about system calls, … Generally, it depends on characteristics and common types of attacks against each systems.

3.2 Attacks on sensor networks

In order to construct a anomaly detection algorithm in sensor networks, it is necessary to analyze some of the most common attacks in sensor networks including: wormhole, blackhole, HELLO flood attack, Jamming, … Most of them focus on vulnerabilities of routing protocols.

- Wormholes: By some ways, an adversary creates communication links between some pairs of compromised sensor nodes. This may attract more sensor nodes to send their traffic via these links. After that, the adversary can eavesdrop, alter or simply drop these packets.

- Blackhole: a black hole is formed when a node tries to advertise a zero-cost route to all other nodes in the network. As a result, more sensors will send traffic through this zero-cost route and will be unsuccessful.
- HELLO flood attack: an adversary broadcasts HELLO packets with large enough transmission power to lure other sensor nodes that the adversary is their neighbor.
- Exhaustion: a sacrificed node keeps transmitting packets to another node to exhaust the target’s battery power.
- Collision: an adversary will try to corrupt a small part of traffic to induce much more collision in some weak protocol.
- Jamming: An adversary can disrupt the network by sending a more powerful signal over the frequency used by the nodes.

To realize the anomaly characteristics of these attacks, we divide them into three major categories: (1) attracting other nodes to send their traffic to a compromised node, (2) causing collision to disrupt sensor network and (3) exhausting a node’s resources by sending many packets to the target. It is straightforward to see that attacks in each category makes the network traffic deviated from that in normal condition in different ways. If the network is under attacks in the first category, traffic to some nodes (compromised nodes) will suddenly increase. Attacks in the second category will raise the number of packet collision and attacks in the third category are revealed by the increasing amount of outgoing traffic related to one node. Therefore, we can detect attacks in sensor networks by monitoring these anomalies. They are the changes in (1) the number of incoming packets to a node, (2) the number of collisions associated with packets sent by a node and (3) the number of outgoing packets from a node.

4. Proposed algorithm

A lot of techniques have been done for anomaly detection such as: neural network, audit data analysis and mining, statistical models, … Each of them has their own pros and cons. Here, we used a widely-used anomaly detection algorithm, Cumulative Sum (CUSUM). CUSUM is suitable to deploy in sensor network because it is a strong, light-weight and less memory consuming statistical model.

4.1 CUSUM algorithm

CUSUM is one of some change point detection algorithms used widely to detect the change of mean value of a random sequence (see [11, 12] for good survey). In brief, CUSUM detect changes based on the cumulative effect of the changes made in the random sequence instead of using a single threshold to check every variable. To detect abrupt changes in a random sequence {Xn}, CUSUM requires a parametric model for {Xn} which it not easy in some cases. Thus, a new approach called non-parametric CUSUM proposed by Wang [13] is used more popular especially in attack detection. Assume that {Xn} have a negative mean in normal condition and become large positive in anomaly operation, we set:

$$y_0 = 0$$
$$y_n = (y_{n-1} + X_n)^+ \qquad (n \geq 1)$$

where

$$(x)^+ = x : x > 0$$
$$= 0 : \text{otherwise}$$

y_n can be canculated in another way:

$$y_n = S_n - \min S_{k,}$$
$$S_k = \sum_{i=1}^{k} x_i$$

In normal operation, the mean of {Xn} is negative so $y_n \sim 0$. In anommaly condition, Xn will become positive. $\{y_n\}$ will accumulate with time. A large $\{y_n\}$ is a strong indication of abrupt changes. In attack detection, we set $d_n(y_n)$ be the decision function. $d_n()$ can be defined as:

$$d_n(y_n) \qquad = 0 \text{ if } y_n \leq N$$
$$= 1 \text{ if } y_n > N$$

(N is the threshold of the attack detection)

We can describe the CuSum algorithm in brief as following:

Algorithm 1

```
CuSum := 0
n := 0
Repeat
    n := n + 1
    CuSum := CuSum + Xn
    If CuSum > ThresHold then
          Signal attack indication
Until Finished
```

where *n* is the nth sampling period

The algorithm is straightforward. The most important thing is how to model {Xn}. In next parts, we will discuss the way to model {Xn} to detect abrupt changes in the number of incoming packets, the number of collision packets and the number of outgoing packets from a node.

4.2 Detecting changes in the number of incoming packets

Let $\{\Delta n$, n = 0, 1, …} be the number of incoming packets to the monitored node collected within one sampling period. However, $\{\Delta n\}$ depends on the size of sampling period and the density of the monitored node's vicinity. To normalize, we simply define $Z_n = \Delta n / \overline{F}$ where $\overline{F}$ is the average number of incoming packets to the monitor node in a sampling period. $\overline{F}$ can be calculated recursively as following:

$$\overline{F}(n) = \alpha \overline{F}(n-1) + (1 - \alpha) INC(n)$$

where INC(n) is the number of incoming packets to the monitor node in the nth sampling period. α is a constant lying between 0 and 1 indicating the memory in the estimation.

Thus, the mean of $\{Z_n\}$ is close to 1 in normal condition. To satisfy the assumption (2), we transform $\{Z_n\}$ to another random sequence without loss of any statistical feature.

$$X_n = Z_n - \beta$$

where β is a constant parameter depending on the network condition to produce $\{X_n\}$ with a negative mean. In general, β is selected to be larger than the mean of $\{Z_n\}$ during normal conditions.

So, we can apply nonparametric CUSUM with a random sequence $\{X_n\}$ to detect changes in the number of incoming packets to the monitored node.

4.3 Detecting changes in the number of outgoing packets

Similarly, let $\{\Delta n$, n = 0, 1, …$\}$ be the number of outgoing packets to the monitored node collected within one sampling period. We define $Z_n = \overline{F} / \Delta n$ where $\overline{F}$ is the average number of outgoing packets to the monitor node in a sampling period.

$$\overline{F}(n) = \alpha \overline{F}(n-1) + (1 - \alpha) OUT(n)$$

where OUT(n) is the number of incoming packets to the monitor node in the nth sampling period. α is a constant lying between 0 and 1.

Thus, the mean of $\{Z_n\}$ is close to 1 in normal condition & become larger under attack. To satisfy the assumption (2), we set

$$X_n = Z_n - \beta$$

β is selected to be larger than the mean of $\{Z_n\}$ during normal conditions.

4.4 Detecting changes in the number of collision

Similarly, we set

$$X_n = Z_n - \beta$$

$$Z_n = S_n - F_n$$

where S_n/F_n is the number of successful/unsuccessful packets in the nth sampling period.

β is selected to be larger than the mean of $\{Z_n\}$ during normal conditions.

Algorithm 2 summarizes our algorithm described above.

Algorithm 2

```
all CuSum = 0
n = 0
repeat
    n = n + 1
      for each neighbor i do
          CuSum(inc i) : = (CuSum(inc i) + Xn(inc i))$^+$
          CuSum(out i) : = (CuSum(out i) + Xn(out i))$^+$
          if any CuSum > Its Threshold then
                Signal attack indication
      end for
      CuSum(collision) : = (CuSum(collision) + Xn(collision))$^+$
    if CuSum(collision) > Its Threshold then
          Signal attack indication
until Finished
```

where *n* is the nth sampling period. inc i means the number of incoming packets of ith neighbor. *out i* means the number of outgoing packets of ith neighbor. *collision* means the number of collision packets of the monitor node.

5. Intrusion Detection Model

Because of the lack of central point to collect data, our Intrusion Detection System is distributed. That means some nodes, called monitor nodes, will be installed Intrusion Detection Agents to protect themselves and their neighbors (called monitored node). Monitor nodes are selected so that every node in the network is monitored by at least one monitor node. One node can be monitored by several monitor nodes. There is a trade off between security level and resources. The more monitor nodes, the higher security level.

Fig. shows the architecture of a monitor node. This node runs the common node functions, like sensoring and data message sending and retransmitting, in addition to the IDS functions. IDS functions are done in promiscuous listening mode in which the node captures all coming packets, analyzing and detecting anomaly behaviors. This architecture is similar to the architecture proposed by R. da Silva in [2]. The major different point here is the “Anomaly Detection” module.

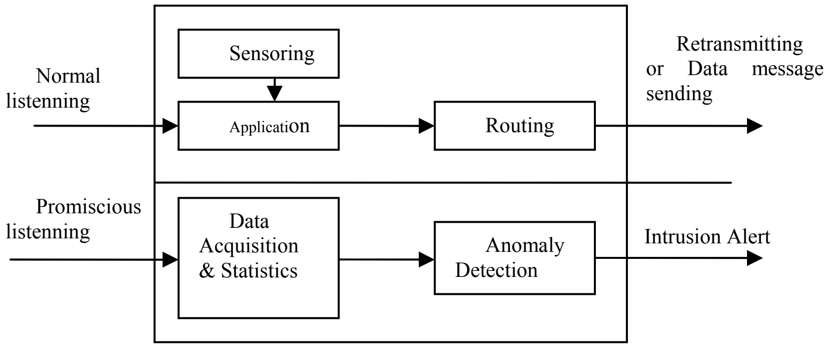

Fig. 1. IDS agent architecture

Data Acquisition & Statistics module

In this module, all packets coming to the monitor node are captured in promiscuous mode. Based on packets' header, this module will count the number of incoming, outgoing packets related to each neighbor. These data will be stored for further analysis in following form:

	Incoming packets	Outgoing packets
Neighbor 1	x	x
Neighbor 2	x	x
…	…	…

Table 1. Input data for anomaly detection module

Anomaly Detection Module

Based on the data from "Data Acquisition & Statistics" module, this module will detect abrupt changes by our proposal algorithm in **III**. If any change is detected, an "Intrusion Alert" will be raised.

6. Discussion

Our proposal approach in this paper bases on Cumulative Sum algorithm which is considered light-weight and powerful to detect abrupt changes in a random sequence. Suppose that the monitor node in the network has k neighbors, the algorithm 2 shows us that the complexity in each step (each sampling period) is O(k). In common sensor networks, k is often less than 10 so the monitor node just needs to do some basic operations in each sampling period. By comparison, the algorithms in [16, 17] have to analyze and check traffic data with a series of rules some of which are not straightforward and require a considerable amount of computational resource.

Besides, little amount of memory resource is required by our algorithm. The monitor node just allocates memory for some trivial variables and a small-size array for statistics data of packers of the neighbors. In [16, 17] they need a huge buffer to store all packets needing to be analyzed and the algorithm turns out poor result with a small-size buffer.

However, simulation is needed to prove strongly the result of this algorithm. In addition, monitor nodes in our system should work in cooperation to detect intrusion faster and with higher accuracy. These drawbacks are what we are focusing on to improve our intrusion detection system.

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci: Wireless sensor networks: A survey, Computer Networks, 38(4):393--422, March 2002.
2. Anthony D. Wood and John A. Stankovic: Denial of Service in Sensor Networks, IEEE Computer, October 2002, pp. 61-62.
3. C. Karlof and D. Wagner: Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures, Ad Hoc Networks, vol. 1, pp. 293-315, 2003
4. Yi-an Huang , Wei Fan , Wenke Lee , Philip S. Yu: Cross-Feature Analysis for Detecting Ad-Hoc Routing Anomalies, Proceedings of the 23rd International Conference on Distributed Computing Systems, p.478, May 19-22, 2003
5. Levente Buttyán, László Dóra, István Vajda: Statistical Wormhole Detection in Sensor Networks. ESAS 2005: 128-141
6. Y. Hu, A. Perrig, and D. Johnson: Packet leashes: a defense against wormhole attacks in wireless ad hoc networks. In Proceedings of the IEEE Conference on Computer Communications (Infocom), 2003.
7. K. Ilgun, R. A. Kemmerer, and P. Porras: State transition analysis: A rule-based intrusion detection approach, IEEE Trans on Software Engineering, 21 (1995), pp. 181–199.
8. Jake Ryan, Meng-Jang Lin, Risto Milikkulainen: Intrusion Detection with Neural Networks, Advances in Neural Information Processing Systems 10 (Proceedings of NIPS'97, Denver, CO), MIT Press, 1998.
9. P. A. Porras and P. G. Neumann: Emerald: Event monitoring enabling responses to anomalous live disturbances, in Proc of 20th NIST-NCSC Nat'l Info Systems Security Conf, 1997, pp. 353–365.
10. M.-Y. Huang, R. J. Jasper, and T. M. Wicks: A large scale distributed intrusion detection framework based on attack strategy analysis, Computer Networks, 31 (1999), pp. 2465–2475.
11. M. Basseville and I. V. Nikiforov: Detection of Abrupt Changes: Theory and Application, Prentice Hall, 1993.
12. B.E. Brodsky and B.S. Darkhovsky: Nonparametric Methods in Changepoint Problems, Kluwer Academic Publishers, 1993.
13. Haining Wang, Danlu Zhang, and Kang G. Shin: Detecting SYN Flooding Attacks, IEEE INFOCOM'2002, New York City, NY, 2002.
14. Riaz A. Shaikh, S.M.H. Zaidi, Saeed Rajput and Kashif Sharif: Review Over Anomaly Detection Algorithms For Detecting SYN Flooding Attacks, proceeding of

4th Annual IEEE Student Conference on Engineering Sciences and Technology (SCONEST 2005), Karachi, Pakistan,30th Aug, 2005

15. V. A. Siris, F. Ppapagalou: Application of Anomaly Detection Algorithms for Detecting SYN Flooding Attacks, Global Telecommunications, IEEE, 2004. 2050-2054.
16. P. R. Da-Silva, M. H.T Martins, B. Rocha,A. Loureiro, L. Ruiz, and H. C. Wong University of Bel-Horizon, Brasil: Decentralized Intrusion Detection in Wireless Sensor Networks.
17. Onat, Ilker, and Miri, Ali: An Intrusion Detection System for Wireless Sensor Networks, to appear in the Proceedings of in the Proceeding of the 2005 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Montreal, Canada, August 2005.