Detecting and Confining Sybil Attack in Wireless Sensor Networks based on Reputation Systems coupled with Self-organizing Maps

Zorana Banković, David Fraga, José M. Moya, Juan Carlos Vallejo, Álvaro Araujo, Pedro Malagón, Juan-Mariano de Goyeneche, Daniel Villanueva, Elena Romero, Javier Blesa

ETSI Telecomunicación, Universidad Politécnica de Madrid,
Av. Complutense 30, 28040 Madrid, Spain
{zorana, dfraga, josem, jcvallejo, araujo, malagon, goyeneche, danielvg, Elena, jblesa}@die.upm.es

Abstract. The Sybil attack is one of the most aggressive and evasive attacks in sensor networks that can affect on many aspects of network functioning. Thus, its efficient detection is of highest importance. In order to resolve this issue, in this work we propose to couple reputation systems with agents based on self-organizing map algorithm trained for detecting outliers in data. The response of the system consists in assigning low reputation values to the compromised node rendering them isolated from the rest of the network. The main improvement of this work consists in the way of calculating reputation, which is more flexible and discriminative in distinguishing attacks from normal behavior. Self-organizing map algorithm deploys feature space based on sequences of sensor outputs. Our solution offers many benefits: scalable solution, fast response to adversarial activities, ability to detect unknown attacks, high adaptability and low consumption. The testing results demonstrate its high ability in detecting and confining Sybil attack.

Keywords: wireless sensor networks, reputation system, self-organizing maps, outlier detection

1 Introduction

WSNs consist of a large number of sensor nodes (also called motes). These nodes have to be very cheap, so they exhibit very limited power and computational resources, small memory size and low bandwidth usage and usually no tamper-resistant hardware is incorporated with any of them.

The most aggressive and the most evasive of all the attacks on sensor networks is the Sybil attack [1]. In essence, it refers to the scenario when one (or more) node(s) claim to have multiple identities, either fabricated or stolen ones. In this way it is able to affect on various aspects on network functioning, some of them being routing protocols, voting (in trust schemes), fair resource allocation, etc. Thus, it is of highest importance to efficiently detect and confine this attack.

We believe that spatial and temporal characterization of the data coming from the sensors can be of great importance in discovering manipulated data and/or compromised nodes. Any major data inconsistence can be connected to malicious data manipulation.

In this work we propose to detect presence the of the Sybil attack using a self-organizing map (SOM) algorithm for detecting data outliers. The first step in deploying any machine learning technique is to define the model of data. The model consists of certain number of characteristics, i.e. features, that describe all possible aspects of the phenomenon. Furthermore, in our case it is essential to be able to distinguish normal from anomalous behavior.

For that reason, we deploy temporal and spatial models of the sensors using *n*-grams. The temporal model is defined for each sensor, while spatial model considers groups of close sensors. Each *n*-gram in the temporal model consists of a predefined number of successive sensor values, while an *n*-gram in the spatial model consists of outputs of all the sensors that make the group. Therefore, the features are the *n*-grams and the feature values are the number of occurrences or the frequency of the corresponding *n*-gram during a certain period of time. Considering that number of *n*-grams is not constant within consecutive periods of time, SOM deploys methods for measuring distance between sequences presented in [2].

We further propose to couple the system of detection agents based on SOM with a reputation system. In our proposal, the output of an agent affects on the reputation system in the way that it assigns lower reputation to the nodes where it detects adversarial activities and vice versa. We envision a reputation system where every node is being examined by at least one agent that resides on a node in its vicinity and listens to its communication in a promiscuous manner, and executes one of the algorithms for detecting attacks or temporal and spatial inconsistencies. We further advocate avoiding any contact with the nodes that have low reputation (below certain threshold). In this way, the compromised nodes remain isolated from the network and have no role in its further functioning. Comparing to our previous work on the subject [4, 5], in this work we propose improved way of calculating reputation based on the output of the SOM algorithm, which is more flexible and discriminative when it comes to distinguishing attacks from normal behavior. Furthermore, we present more thorough results on the behavior of the SOM algorithm in different scenarios.

The rest of the work is organized as follows. Section 2 present common solutions for treating the problem of the Sybil. Section 3 details the proposed solution, while Section 4 presents obtained results. Finally, conclusions are drawn in Section 5.

2 Previous Work on Coping with the Sybil Attack

The proposed solutions to the Sybil attack include [1]:

1. Radio resource testing which relies on the assumption that each physical device has only one radio;
2. Random key pre-distribution which associates the identity of the node to the keys assigned to it and validate the keys to see if the node is really who it claims to be;
3. Registration of the node identities at a central base station;

4. Position verification which makes the assumption that the sensor network topology is static.

Each of the above solutions has its own drawbacks. For example, we do not know in advance that every physical device is going to have only one radio interface. Moreover, some of the MAC protocols rely on the fact that each node has more than one radio interface. The key pre-distribution is challenging, since attackers can deploy side-channel attacks in order to discover secret keys [3]. Finally, the last solution is applicable only in static networks, which is very uncommon scenario since there is often a number of mobile nodes that change their position.

3 Proposed Solution

3.1 Feature Extraction and Formation of Model

As previously mentioned our idea is to find temporal and/or spatial inconsistence in sensed data in order to detect manipulated data and/or compromised nodes. For this reason, we follow the idea presented in our previous work [4] based on extracted *n*-grams and their frequencies within different time windows. For the purpose of illustration, we will give a short example for a sensor that detects presence. Let the sensor give the following output during the time window of size 20: 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0. If we fix the *n*-gram size on 3, we extract all the sequences of size 3 each time moving one position forward. In this way we can observe the following sequences and the number of their occurrences within the time window: 111 – occurs 6 times, 110 – 2, 100 – 2, 000 – 6, 001 – 1, 011 – 1. Thus, we can assign them the following sequences: 111 – 0.33, 110 – 0.11, 100 – 0.11, 000 – 0.33, 001 – 0.05, 011 – 0.05. In our model, the sequences are the features and their frequencies are the corresponding feature values. Thus, the sum of the feature values is always equal to 1. In our algorithm this characterization is performed in predefined moments of time and takes the established amount of previous data, e.g. we can perform the characterization after every 40 time periods based on previous 40 values.

In a similar fashion, we form features for spatial characterization. The first step is to establish vicinities of nodes that historically have been giving consistent information. Furthermore, since an agent is supposed to reside on a node, vicinities are established using the nodes which information can reach the agent. In this way, an *n*-gram for spatial characterization in a moment of time is made of the sensor outputs from that very moment. For example, if sensors S1, S2, S3 each give the following output: 1 1 1 0 during four time epochs, we characterize them with the following set of *n*-grams (each *n*-gram contains at the first position the value of S1, the value of S2 at the second and the value of S3 at the third at a certain time epoch): 111 – occurs 3 times, 000 – occurs once, thus the feature value of each *n*-gram is: 111 – 0.75, 000 – 0.25, i.e. the frequencies within the observed period of time.

3.2 Detection of Sybil

The achievement of our design is based on the following two important assumptions:

1. The adversary can capture only a limited number of nodes, which means that most of the output data produced by the sensor nodes is normal. If this is not the case, it means that the adversary has become very powerful, so he is able to subvert any protocol in the network, which would require for the network re-initialization.
2. Output data produced under the influence of an adversary are statistically different from the output produced during the normal operation of the network. For this reason, we establish the detection of anomalies in data as outlier detection (an outlier is an observation that lies an "abnormal" distance from other values in a random sample from a population, i.e. extreme points in the data cloud).

If any of these assumptions is not fulfilled, our model is not able to work properly.

We treat attacks as data outliers and deploy SOM explained in more detail in our previous works [5]. There are two possible approaches for detecting outliers [6] using clustering techniques depending on the following two possibilities: detecting outlying clusters or detecting outlying data that belong to non-outlying clusters. For the first case, we calculate the average distance of each node to the rest of the nodes (or its closest neighborhood) (*MD*). In the latter case, we calculate quantization error (*QE)* of each input as the distance from its group center.

In our case, due to the definition of features and the deployed distance function, the distance can take values from the range [0, 2]. The process of updating cluster centers results in the centers that have all the *n*-grams that appear in the elements that belong to them, and the sum of their values is 1. Thus, in the normal case, *QE* will have values between 0 and 1. However, if an adversary manipulates data, it will result in different *n*-grams, so the corresponding distance will be between 1 and 2. For the same reason, if we have anomalous data in the training, they will form their own clusters. In this case, *MD* will be between 1 and 2, which is taken as anomalous.

3.3 Recovery from Sybil

Every sensor node is being examined by agents that execute SOM algorithm and reside on nodes in its vicinity and listen to its communication. The agents are trained separately. The system of agents is coupled with a reputation system where each node has its reputation value that basically reflects the level of confidence that others have in it based on its previous behavior. In our proposal, the output of an agent affects on the reputation system in the way that it assigns lower reputation to the nodes where it detects abnormal activities and vice versa. We further advocate avoiding any kind of interaction with the low-reputation nodes: to discard any data or request coming from these nodes or to avoid taking them as a routing hop. In this way, compromised nodes remain isolated from the network and have no role in its further performance. After this, additional actions can be performed by the base station, e.g. it can revoke the keys from the compromised nodes, reprogram them, etc.

In this work the reputation is calculated in the following way. We define two reputation values, *repQE* and *repMD* based on the previously defined *QE* and *MD*

values and afterwards joint reputation *rep* used for updating overall reputation based on these two values:

```
if (QE<1) repQE = 1;            if (MD<1) repMD = 1;
else repQE=1-QE/2;              else repMD=1-MD/2;
```

For the reasons explained in the previous chapter, the value (*rep*) for updating overall reputation is calculated in the following way:

```
if (QE>1)rep=repQE;  else rep=repMD;
```

There are two functions for updating the overall reputation of the node, depending on whether the current reputation is below or above the established threshold that distinguishes normal and anomalous behavior. If the current reputation is above the threshold and the node starts behaving suspiciously, its reputation will fall quickly. On the other hand, if the reputation is lower than the established threshold, and the node starts behaving properly, it will need to behave properly for some time until it reaches the threshold. In order to achieve this, we use the function *x+log(1.2*x)* because it provides what we want to accomplish: if *x* is higher than 0.5, the output rises quickly, so the reputation rises; if *x* is around 0.5, the output is around 0, so the reputation will not change its value significantly; if *x* is smaller than 0.4, the output falls below 0. Finally, the reputation is updated in the following way:

```
if (last_reputation[node]>threshold)
new_reputation[node]=last_reputation[node]+rep+log(1.2*rep);
else new_reputation[node]=last_reputation[node]+0.05*(rep+log(1.2*rep));
```

If the final value falls out from the [0, 1] range, it is rounded to 0 if it is lower than 0 or to 1 in the opposite case. In this way, we achieve that once a node start behaving suspiciously, its reputation will fall quickly. Yet, if a malicious node starts behaving properly, it will have to maintain its correct behavior during some time in order to "redeem" itself.

However, if during the testing of temporal coherence, we get normal data different from those that the clustering algorithms saw during the training, it is possible to get high *QE* value as well. On the other hand, the spatial coherence should not detect any anomalies. Thus, the final reputation will fall only if both spatial and temporal algorithms detect anomalies. In the opposite case, its reputation will not change significantly. This is implemented in the following way:

```
if (value_rep < threshold) {
   if ( space_rep < threshold ) result = value_rep;
    else result = 1 - value_rep; }
 else result = value_rep;
```

where `value_rep` is the reputation assigned by the SOM for temporal characterization and `space_rep` is the reputation assigned by the SOM for spatial characterization.

4 Results

The proposed algorithm has been tested on a simulator of sensor networks developed by our research group. The simulated sensor network contains 200 sensor nodes that can take one of the possible 2000 positions. The network simulates a sensor network for detecting presence in the area of application, i.e. sensors give output 1 if they detect presence of a person or an object, or 0 if they do not detect any presence. The groups for spatial SOM algorithm are formed in a way that close sensors that should

give the same output are placed in the same group. The simulation was carried out on a general purpose computer.

In our experiments the Sybil attack impersonates 10 existing sensor IDs. The duration of the experiment is 1000 time ticks. In the following we will present results in different scenarios regarding the presence of Sybil in training data and regarding two different definitions of *MD* value. In the first case, *MD* is defined as the medium distance to the three closest groups, while in the second case *MD* is the maximum of the three closest groups. This can be expressed mathematically in the following way:

Case 1:

$$MD(\mathbf{x}) = \frac{1}{3}\sum_{i=i_1,i_2,i_3}\left\|\mathbf{v}(\mathbf{x})-\mathbf{m_i}\right\|^2 \quad (\mathbf{1})$$

$$i_1 = \arg\min_i \left\|\mathbf{v}(\mathbf{x})-\mathbf{m_i}\right\|^2, i_2 = \arg\min_i \left\|\mathbf{v}(\mathbf{x})-\mathbf{m_i}\right\|^2, i_2 \neq i_1, i_3 = \arg\min_i \left\|\mathbf{v}(\mathbf{x})-\mathbf{m_i}\right\|^2, i_3 \neq i_1, i_3 \neq i_2$$

Case 2:

$$MD(\mathbf{x}) = \max_i \left\|\mathbf{v}(\mathbf{x})-\mathbf{m_i}\right\|^2, i \in \{i_1, i_2, i_3\} \quad (\mathbf{2})$$

$$i_1 = \arg\min_i \left\|\mathbf{v}(\mathbf{x})-\mathbf{m_i}\right\|^2, i_2 = \arg\min_i \left\|\mathbf{v}(\mathbf{x})-\mathbf{m_i}\right\|^2, i_2 \neq i_1, i_3 = \arg\min_i \left\|\mathbf{v}(\mathbf{x})-\mathbf{m_i}\right\|^2, i_3 \neq i_1, i_3 \neq i_2$$

where **v(x)** is the centre to which the current input belongs.

Fig. 1.a and 1.b show the evolution of the reputation of every node after and before introducing the Sybil attack where the training stops at 600th time tick and Sybil starts at 650th. The *MD* value in the first case is calculated according to the formula (1), while in the latter case it is calculated according the formula (2).

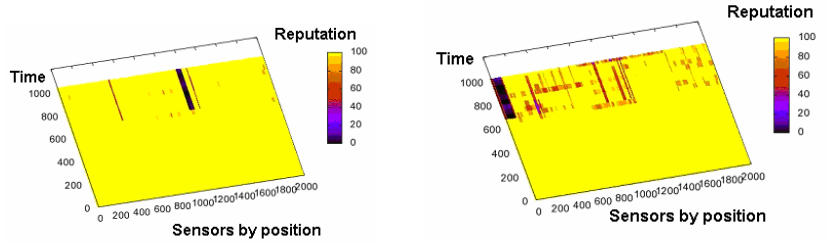

Fig. 1. Reputation Evolution (a) Case 1 (b) Case 2

Fig. 2.a and 2.b show the detection evolution in both of the cases. In these figures real positives are well-behaved nodes, real negatives are the ill-behaved nodes. Fake positives are non-detected ill-behaved nodes, while fake negatives represent the portion of well-behaved nodes falsely detected as ill-behaved.

In both Fig. 2.a and 2.b we can observe a thick dark line, which stands for the group of nodes attacked by Sybil. (Sybil attacks random nodes in each simulation, which is the reason why the dark lines are at different position.) The dark color reflects their low reputation. Fig. 2.a and 2.b confirm that in both cases all the attacked nodes have been detected (Fake Positive line). However, in Case 2 higher number of nodes should be sacrificed in order to confine the attack (Fake negative

line). This can also be concluded from Fig. 1.a and 1.b. However, the advantage of Case 2 is its robustness, which will be demonstrated in the following.

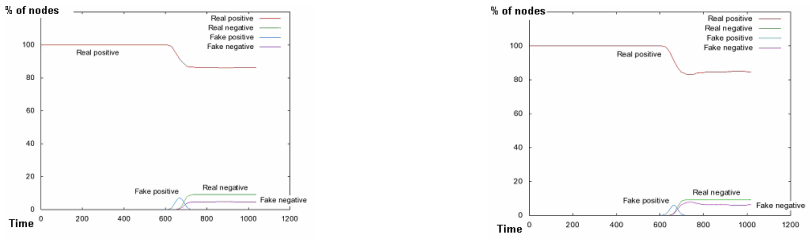

Fig. 2. Detection Evolution (a) Case 1 (b) Case 2

In the following experiment Sybil starts at time tick 300. In the Case 2, the detector identifies and confines all the malicious nodes without having to change any of the parameters from the previous case (Fig. 4), while in Case 1 the detector detects the presence of the attack, but it is not able to confine it completely (Fig. 3). Experimenting with various parameters, we concluded that the maximum point to stop the training is 350 in order to completely confine the attack (Fig. 5). It is obvious that Case 1 is more sensitive to the presence of outlying data as minority, while Case 2 is more robust. These experiments also demonstrate that our system functions properly without the limitation of having (or not having) traces of attack in training data. Furthermore, we have demonstrated that detection possibilities of detectors can be enhanced through parameter changing.

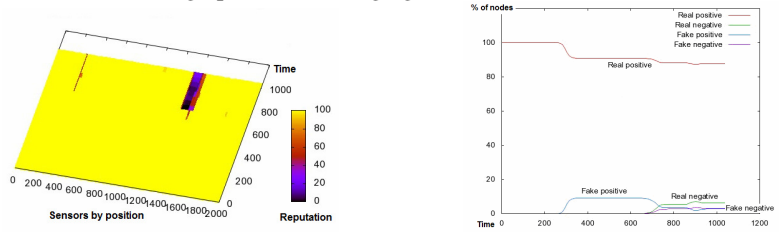

Fig. 3. Case 1 (a) Reputation Evolution (b) Detection Evolution

Concerning the time of detection and confinement of the Sybil, our system is capable of detecting and completely confining the attack if up to 30% of the existing IDs have been taken by the Sybil. The presence of the attack is detected at the end of the first testing cycle in all the cases, while the confinement time spans from one to four testing cycles and becomes higher as the Sybil takes more than 15% of the IDs.

4 Conclusions

In this work we have presented a novel approach for coping with the Sybil attack in wireless sensor networks. We have proposed unsupervised machine learning SOM

algorithm for detecting outliers in data and deploys a feature set that is more general than those presented by the solutions of the state-of-the-art. Furthermore, it does not depend on the presence (or non-presence) of anomalous data during the training.

The idea of confining the Sybil is based on assigning reputation values to the nodes according to the decision of SOM algorithm. In this way, malicious nodes become isolated from the network which will impede them to further propagate their malicious activity. Our experiments demonstrate that our system is capable of detecting and confining Sybil attack.

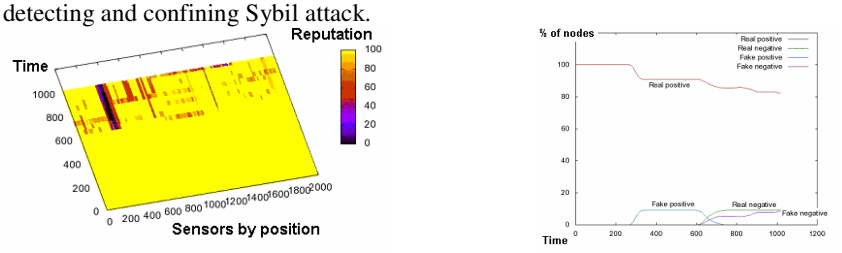

Fig. 4. Case 2 (a) Reputation Evolution (b) Detection Evolution

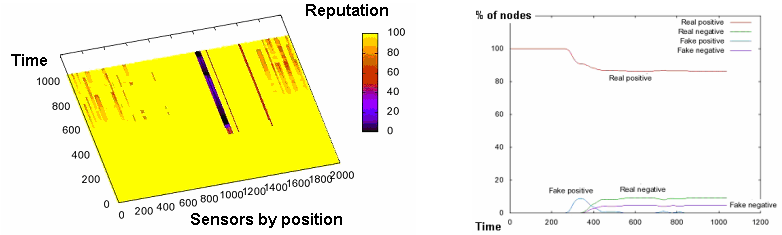

Fig. 5. Case 1 repeated (a) Reputation Evolution (b) Detection Evolution

References

1. Newsome, J.; Shi, E.; Song, D.; Perrig, A. The Sybil Attack in Sensor Networks: Analysis & Defenses. In *Proceedings of ACM/IEEE International Conference on Information Processing in Sensor Networks.* ACM: Berkeley, CA, USA, April 26-27, 2004; pp. 259-268.
2. Rieck, K.; Laskov, P. Linear Time Computation of Similarity for Sequential Data. *J. Mach. Learn. Res.* **2008,** *9*, 23-48.
3. Bar El, H. *Introduction to Side Channel Attacks*. Discretix Technologies Ltd., 2003.
4. Moya, J. M et al. Improving Security for SCADA Sensor Networks with Reputation Systems and Self-Organizing Maps. *Sensors* **2009,** *9*, 9380-9397.
5. Banković, Z.; Moya, Moya, J. M.; Araujo, A.; Fraga, D.; Vallejo, J.C.; de Goyeneche, J. M. Distributed Intrusion Detection System for WSNs based on a Reputation System coupled with Kernel Self-Organizing Maps. To be published in: *Int. Comp. Aided Design*
6. Muñoz, A.; Muruzábal. J. Self-Organizing Maps for Outlier Detection. *Neurocomputing 18(1-3),* **1998**, 33-60