An Adaptive DoS Attack Mitigation Measure for Field Networks in Smart Grids

Gunee Lee, Yun-Sam Kim and Jungmin Kang

Abstract The wireless mesh networks is one of the key technologies for field device networks to maximize the effect of smart grid. However, the wireless mesh networks is exposed to DoS attack based on routing misbehavior that cause the network does not work properly. If sensed data cannot be transferred from field network to server side system, we could not get the benefits of smart grid successfully. For protecting the field networks in smart grids from DoS attack based on routing misbehavior, we propose a revised monitoring method that improve the level of security of the wireless mesh networks for smart grid's field device. We also provide the result of experiments.

1 Introduction

Smart grid is an intelligent power grid equipped with information communication technologies. With smart grid, electricity utilities could estimate electricity demand based on customer electricity usage information collected from smart meters, and then they might control peak load situation based on the estimation. Before an electricity peak load occurs, electricity utility has customer reduce electricity usage or makes customer use electricity generated by distributed electricity resource (DER) in the customer premises, which are polar voltaic over the roof, electricity storage

Gunhee Lee
National Security Research Institute, 1559 Yuseong-daero, Yuseong-gu, Daejeon 34044, Korea, e-mail: icezzoco@nsr.re.kr

Yun-Sam Kim
National Security Research Institute, 1559 Yuseong-daero, Yuseong-gu, Daejeon 34044, Korea, e-mail: bijak@nsr.re.kr

Jungmin Kang
National Security Research Institute, 1559 Yuseong-daero, Yuseong-gu, Daejeon 34044, Korea, e-mail: jmkang@nsr.re.kr

L. Barolli et al. (eds.), *Advances on Broad-Band Wireless Computing, Communication and Applications*, Lecture Notes on Data Engineering and Communications Technologies 2, DOI 10.1007/978-3-319-49106-6_41

and electricity vehicle. Moreover, customer could delay or bringing forward electricity usage based on the information about the peak load time came from the utility.

Moreover, for maintaining the reliability of power grids, smart grids may have a capability of situation-awareness. With the capability, smart grids could determine the current status of the power grid. It can detect a failure point in power grid near real-time, so the power company can respond to the failure in time. In order to do that, the sensors and the actuators should be deployed over the whole smart grids.

Since the smart grid deployed over the wide and open area, wireless mesh network is one of the highlighted technologies for the field network of smart grids. For example, smart meters form a wireless mesh network in order to transfer their measured value to a utility. A data concentrator connected to a smart meter mesh network collects the measured values and sends the bunch of data to a AMI head-end in a utility.

To achieve the purpose of the smart grids, collecting information from sensors (i.e. smart meters and phasor measurement units) and issuing a command to actuator (i.e. controllers of DER, controllers of the electricity loads and IEDs on protection relays) are the most important tasks. Blocking, intercepting, and dropping the transmission of information and command would be one of the most serious threats against the smart grids. Unfortunately the wireless mesh networks, which is the most spotlighted technology, has high possibility to occur those kinds of threats by intentionally launching the routing misbehaviour attacks against the sensors and the actuators.

One or more compromised node could drop entire or some packets from other smart meters, and then the smart meter network could not provide the functionality of information delivery. The loss of the energy information delivery capability could be expected to have adverse effect on various applications in smart grid, such as demand response, load control, distribution management, and so on.

To mitigate the DoS (Denial of Service) attack based on routing misbehavior, we had introduced a effective method that monitors routing behaviour of neighbour nodes in a wireless mesh network[1]. The previous method, however, has limitations in efficiency and it could not handle a gray hole attack properly. Thus, in this paper, we propose a revised method in order to improve the efficiency of the method and mitigate a gray hole attack properly. To do so, we design a method for building neighbour list and propose an algorithm that adaptively determine a threshold of packet drop ratio. The threshold is the criteria that determine whether a node is compromised by attacker or not.

The rest part of this paper is organized as follows. Section 2 presents the overview of the previous method we introduced and explains its limitations. Section 3 describes a revised and proposed method and algorithm in detail. Section 4 provides the experimental results, and Section 5 draws conclusion.

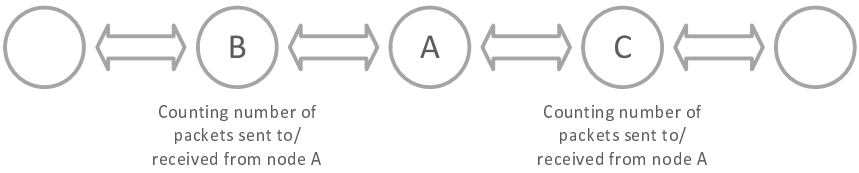

Fig. 1 Network traffic information monitoring method using cooperating nodes

2 The Previous Measure and its Limitations

Fig. 1 shows the schematic diagram of the previous method to monitor the wireless ad hoc networks and detect routing misbehavior. In this example, nodes in a wireless ad hoc network form a detection cluster. In order to monitor node *A*, neighbor nodes around node *A* collect necessary information continuously. That is, two nodes *B* and *C*, which have node *A* between them, count the number of packets that they have forwarded to node *A* and the number of packets that they have received from node *A*. Any node except node *A* in a cluster could be a watchdog node that monitors behavior of node *A*, and the watchdog node collects and analyzes the counter data from the other nodes in order to determine whether node *A* performs routing misbehavior.

In the previous method, since the watchdog node in the cluster should know its one-hop neighbors and two-hop neighbors, where the two-hop neighbor is all the nodes that could reach in two-hop routing via the monitored node. From the previous example, if the node *B* is a watchdog node, the node *C* is a two-hop neighbor of the node *B*. In the previous method, each node performs authentication process between its one-hop neighbors and two-hop neighbors in order to identify its two-hop neighbors. This process considers not only secure session key exchange between nodes but also mutual authentication between them. Thus when a new node is installed, there would be massive flood of message exchange between the new node and existing node.

In addition, it the previous method, the watchdog node calculate drop ratio of the monitored node, and it checks if the ratio is higher than a threshold. If so, the monitored node could be a attacker node. Thus the determining threshold is a key issue of the method. Higher threshold might increase false-negative ratio of the method and lower threshold might increase false-positive ratio of the method. Moreover, the threshold would be affected by the network performance as well. For example, in rainy day, the packet drop ratio could be increased, so a probability that the normal node would be identified as a attacker could be increased.

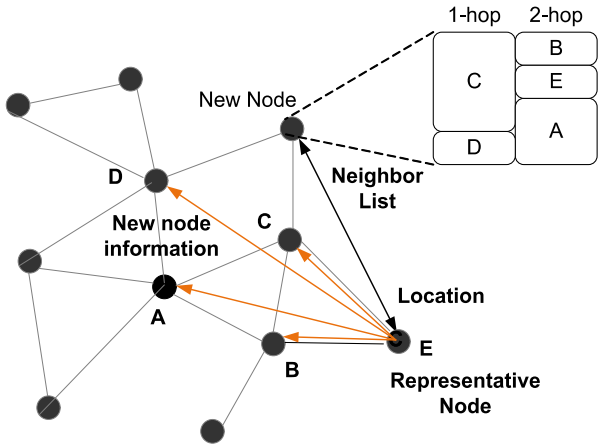

Fig. 2 The schematic view of the neighbor list distribution

3 The Proposed Method

3.1 Building a neighbor list

Fig. 2 shows the schematic view of the neighbor list distribution in the proposed method. When a new node joins a network, it should perform joining process with the representative node, and they should agree with a session key between them. The representative node is the only manager of a wireless mesh network. For example, in the advanced metering infrastructure, a data concentrator, which connects the smart meter networks and the AMI headend, can be a representative node for a smart meter network. After that, the new node should notice its transmission range and its location that is attained through the GPS or the location discovery algorithms [2, 3, 4, 5, 6]. The location is presented with a pair of X-coordination and Y-coordination. It should be encrypted with a session key, and it should be sent to the representative node.

As a new node is registered, the representative node should maintain the nodes information and the location information of the wireless mesh networks. The procedure is defined in the Algorithm 1. When the representative node receives the location information of a new node, it inserts the information of the new node into the *node information table*. The *node information table* is a list of nodes in a wireless mesh network. Each item in the list consists of 4-tuple including node's id, X-coordination of node's location, Y-coordination of node's location and transmission range. Then, it should modify the *location map* according to the information. The location map is a adjacency matrix for the wireless mesh networks. After that, with the location of the new node and the transmission range, the representative node looks up the *node information table* to find every one-hop neighbors of the new node. Based on the well known circle equation, the representative node is able to select every one-hop neighbor of the new node.

Algorithm 1 Modifying the location map of the representative node

```
1: N: a node information table
2: n: number of nodes
3: M: a location map
4: L: a list for the one-hop neighbors of the new node
5: Adj: an adjacency list for two-hop neighbors of the new node
6: r: transmission range of the new node
7: (a,b): location information on the new node
8: procedure ModifyLocationMap(a, b, r)
9:     N[n] ← (id,a,b,r)
10:    n ← n+1
11:    SelectOneHopNeighbors(a, b, r, L)
12:    for k = 0 → n do
13:        M[n][k] ← L[k]
14:        M[k][n] ← L[k]
15: procedure SelectOneHopNeighbors(a, b, r, L)
16:    for i = 0 → n−1 do
17:        if a−r < N[i].x < a+r then
18:            if b−r < N[i].y < b+r then
19:                L[i] ← 1
```

The algorithm that builds a *neighbor list* for the new node is given in Algorithm 2. The *neighbor list* is a adjacency list that includes every one-hop neighbors and two-hop neighbors of the new node. Each head node of the list is a one-hop node of the new node and the other nodes of the list are the two-hop nodes of the new node. The algorithm is easy to understand. For each one-hop neighbor of the new node, it is added as a header item of the list and the one-hop neighbors of the header are added at the next of the header. The neighbor list, which is encrypted with the session key, is delivered to the new node.

Algorithm 2 Building a neighbor list for a new node

```
1: N: a node information table
2: n: number of nodes
3: M: a location map
4: Adj: an adjacency list for two-hop neighbors of the new node
5: r: transmission range of the new node
6: (a,b): location information on the new node
7: procedure BuildNeighborList(id, a, b, r)
8:     l,k ← 0
9:     for i = 0 → n do
10:        if M[n][i] == 1 then
11:            Adj[l][0] ← N[i].id
12:            for j = 1 → n do
13:                if M[i][j] == 1 then
14:                    Adj[j][k] ← N[j].id
```

Algorithm 3 Sending a new node information to neighbors of the new node

```
1: N: a node information table
2: n: number of nodes
3: M: a location map
4: Adj: an adjacency list for two-hop neighbors of the new node
5: L1: a list for the one-hop neighbors of the new node
6: L2: a list for the two-hop neighbors of the new node

7: procedure SendToOneHopNeigbors
8:     for i = 0 → n − 1 do
9:         if M[n − 1][i] == 1 then
10:            k ← 1
11:            for j = 0 → n − 1 do
12:                if M[n − 1][j] == 1 ∧ j ≠ i then
13:                    L1[k] ← N[j].id
14:                    k ← k + 1
15:            L1[0] ← N[n − 1].id
16:            Send L1 to N[i].id

17: procedure SendToTwoHopNeigbors
18:     for i = 0 → n − 1 do
19:         if M[n − 1][i] == 1 then
20:             for j = 0 → n − 1 do
21:                 if M[i][j] == 1 then
22:                     L2[0] ← N[i].id
23:                     L2[1] ← N[n − 1].id
24:                     Send L2 to N[j].id
```

The Algorithm 3 provides the procedure that sends the partial neighbor list adding to the neighbor list of the new node's neighbors. According to the Algorithm 3, the representative nodes sends the new item of the adjacency list for one-hop neighbor of the new node, and it also sends the information of the relay nodes inbetween the two-hop nodes of the new node and the new node.

3.2 Monitoring and Detection of Routing Misbehavior

For detecting the misbehavior of compromised nodes, in the proposed method, we have two steps such as monitoring step and detection step as provided in the previous method[1]. In the monitoring step, each node continuously monitors their neighbors by counting the number of incoming/outgoing messages from/to monitoring node. For the detection step, any watchdog node selected as a detector should collect the counter from two-hop neighbors, which are one-hop neighbor of the monitoring node. Then the detector checks whether the ratio of the incoming and the outgoing packets is in a predefined threshold. If it is in the threshold, the monitoring node behaves normally. Otherwise, it might be compromised node.

The only difference between the previous method and the proposed method in this paer is the detector. As mentioned above, in the previous method, any watchdog node can be a detector. In the proposed method here, however, the only node that can be a detector is the representative node.

3.3 Adjustment of threshold for the packet drop ratio

The attackers can learn or estimate the threshold with his/her knowledge and practices. He/she could achieve the average drop ratio of nodes in the network, and it can select the threshold according to the average value. With the estimated threshold, the attacker can perform gray-hole attack or selective forwarding attack to the networks. Although the attack is launched, the representative node cannot notice that since the value of forwarding ratio does not exceed the threshold.

Thus, in the proposed method, we would employ an automatic threshold revision process. This process would revise the threshold H_rate after a pre-defined time duration. In a normal situation, as the number of revision would be increased, the new threshold value would be stable, so after some time the threshold value is stays at the same value. However, in a abnormal situation, the trend of threshold value's variation would be drifted from time to time. The abnormal situation includes not only attack situation but also accidental events that result in increasing the packet loss ratio.

Fig. 3 shows the schematic view of the revision algorithm. In the figure, there are two groups of forwarding ratio such as group Y and group X. Forwarding ratios of nodes in a local area are categorized into those two groups through any clustering algorithm such as K-means, Fuzzy C-means, Hierarchical clustering, and Mixture of Gaussians [7]. The proposed method will calculate a new threshold H'_{rate} in order to increase the accuracy of the proposed framework. For generating a new threshold, the representative node first calculates the median value of two values such as old threshold H_{rate} and the mean value of group Y as given in equation (1) and (2). Then, it calculates average of deviations that are differences between each value of group X and the median value M according to the equation (3). Finally the new threshold is the sum of the median value M and the average of deviations that are average of differences between each value of group X and M as shown in equation (4).

$$M = Median[H_{rate}, \overline{Y}] \tag{1}$$

$$Y = \{r_d(y) | y \in Y\}, \text{where } r_d(y) = \frac{M_f(y)}{M_r(y)} \tag{2}$$

$$V = \{r_d(x) - M | x \in X, r_d(x) > M\}, \text{where } r_d(x) = \frac{M_f(x)}{M_r(x)} \tag{3}$$

$$H'_{rate} = M + \overline{V} \tag{4}$$

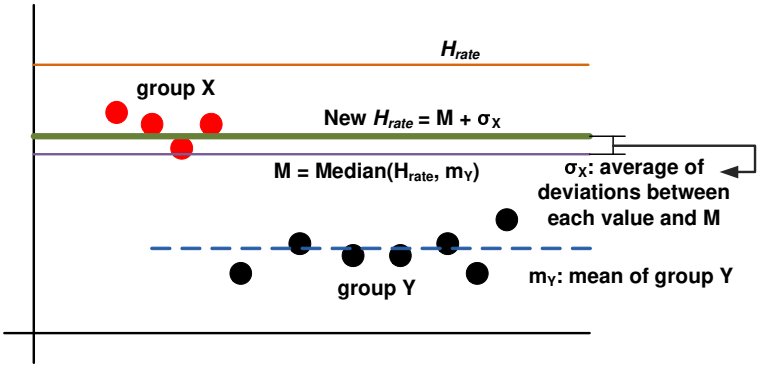

Fig. 3 Adjusting threshold value

4 Experiment results

In order to evaluate the proposed method, we conducted an experiment using network simulator NS-2. In the experiment, an attack that paralyzes network service by filtering out network packets at compromised nodes in the network was launched against the network without any protection method. The same attack was launched against the network employing the proposed method.

In the experiment, 50 smart meters were installed in an area of 1500m300m. Network routing was performed by the DSR (dynamic source routing) method. Twenty of the nodes were CBR sources sending 4 packets per second. In each session, 0 10 attackers were distributed in the network. When communication was active in the network, each attacker dropped a certain percentage of network packets so that network service would be discontinued. Details of the simulation parameters are given in Table 1.

Table 1 Parameters for the experiments

Parameters	Values
Number of nodes	50
Area size (m)	1500×300
Traffic model	CBR
Transmission rate (*packets/s*)	4
Maximum number of connections	20
Packet size (byte)	512
Duration (s)	900

Fig. 4 shows the successful intrusion detection ratio and the false intrusion detection ratio of the proposed system according to the number of attacker nodes. The detection ratio increased slightly with the increase in the number of attacker nodes, and was over 97% even when the number of attacker nodes was smallest. This also

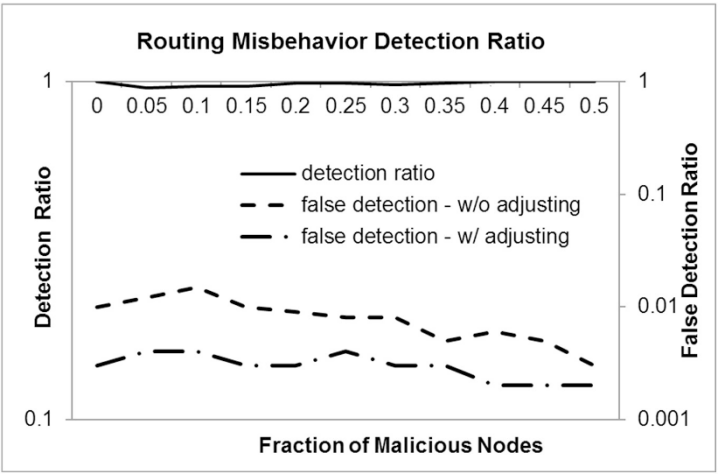

Fig. 4 Changes in the intrusion detection ratio

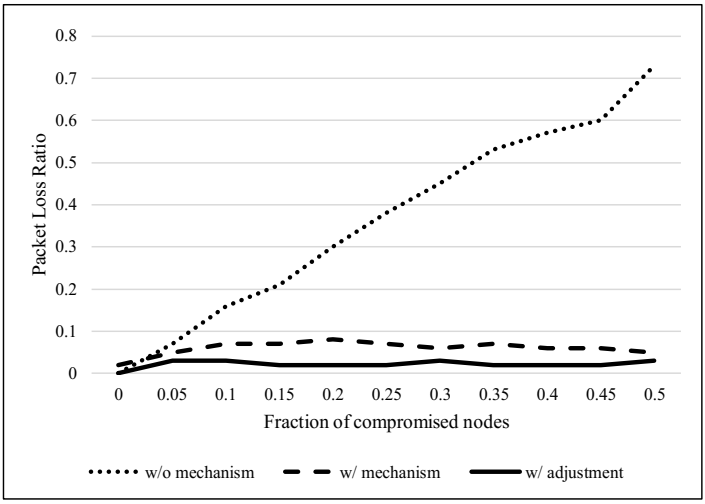

Fig. 5 Change in the packet loss ratio according to the number of attackers

means that false-negative is less than 3%. In the Figure 15, the false detection ratio means false positive ratio to total number of attack detection. When there was no attacker node, false detection ratio was 0.3%. Regardless of the number of attacker nodes, false detection ratio was less than 1%. This suggests that the possibility of false-negative or false-positive is low enough to apply the proposed method to the real environment.

Fig. 5 shows the packet loss ratio according to the change of the number of attacker nodes. When the proposed approach was not employed, the packet loss ratio went up to 73% with the increase in the number of attacker nodes. When the proposed detection method was applied, however, the packet loss ratio was stable

within 3%. This ratio includes packet loss caused by attackers and that caused by processing delay resulting from the execution of the routing misbehavior detection process.

5 Conclusion

In this paper, we proposed a revised monitoring method that can detect DoS attacks based on routing misbehavior. In contrast with the previous method we published before[1], the proposed method would remove a burden of authentication process. In addition, the proposed method could detect more intelligent routing misbehavior. The results of experiment in DSR-based network environment showed that the proposed method could prove the efficiency and effectiveness.

Acknowledgements This work was supported by the the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20141010501870)

References

1. B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, Global Positioning System: Theory and Practice, In the Fourth Edition, Springer-Verlag (1997)
2. L. Hu and D. Evans, Localization for Mobile Sensor Networks, In Proc. of ACM MOBICOM, pp. 45-57 (2004)
3. D. Liu, P. Ning, and W. Du, Detecting Malicious Beacon Nodes for Secure Location Discovery in Wireless Sensor Networks, In Proc. of the 25th International Conference on Distributed Computer Systems (ICDCS), pp. 609-619 (2005)
4. W. Du, L. Fang, P. Ning, LAD: Localization Anomaly Detection for Wireless Sensor Networks, In the Journal of Parallel and Distributed Computing (JPDC), Vol. 66, No. 7, pp. 874-886 (2006)
5. N. Sastry, U. Shankar, and D. Wagner, Secure verification of location claims, In Proc. of ACM Workshop on Wireless Security (WiSe), pp. 1-10 (2003)
6. C. H. Romesburg, Cluster Analysis for Researchers, (2004)