Detecting compromised sensor node with subjective logic in wireless sensor network

Hongwei Zhou[1, a], Jinhui Yuan[1, b], Laishun Zhang[1, c] and Rui Xiao[1, d]

[1]Information Engineering University, Zhengzhou, 450001, China.

[a]hong_wei_zhou@126.com, [b]jcyjh@126.com, [c]zhang ls2002@163.com, [d]xd_xiaorui@163.com

Keywords: Sensor network, subjective logic, anomaly detection.

Abstract. To detect anomaly sensing data, in some existing methods, a sensor node first checks its sensing data. Only the node finds its data may be anomalous, it turns to other nodes for further detection. This reduces the energy consume, but it introduces the risk of cheat. Since sensor nodes are often physically captured, a compromised sensor node is able to stop checking its data. To cover it, in this paper, we propose the method that every neighbor node checks directly the node's sensing data and sink node fuses all neighbor node's detection to obtain the final conclusion. More importantly, neighbor node's detection is quantified as subjective logic opinion, and these opinions are fused with fusing operator of subjective logic. When only few nodes are compromised, our solution is capable of exposing compromised node.

Introduction

Sensor network plays an important role in our life[1,9]. For example, to warn potential fires in history museum, a lot of sensor nodes are deployed densely in the monitoring area. However, the sensor nodes are not always deployed in the safe region. So the sensor nodes are easy to be physically captured, and an attacker can send some false sensing data to disturb the entire sensor network. To overcome it, the sensor network have to find the anomalous sensing data and expose compromised nodes.

Some solutions are proposed to check anomalous sensing data[4,5]. In our opinion, they are two-phase methods. At first, every sensor node checks its sensing data. If its sensing data may be anomalous, the node turns to its neighbor nodes for further detection. So its neighbor nodes cooperate to check the data. However, this method is not always run smoothly. Due to the nature that the sensor nodes are usually deployed in insecurity area, it is easy to be physically captured. In this scenario, a compromised node may stop checking its data and send some false sensing data to disturb sensor network.

To cover it, in this paper, we propose one novel method. In sensor network, some sensor nodes do not send sensing data to save energy. Since transmitting sensing data takes most of the energy, one sensor node only intercept its neighbor node's message if it sends similar data. In this paper, we call the node that send sensing data as *active node*, and other node as *inactive node*. Noted that active nodes are our checking targets, and their neighbor nodes are always intercepting their sensing data for the detection. If one inactive node finds any active node sends a false data, it sends an alarm message to sink node. At this time, the sink node collects all messages on the same active node, and fuses the messages to obtain the final result.

Since some inactive nodes may send alarm messages, so we have to face one problem: these inactive nodes have to circle the compromised active node and find a new road to sink node. Once a routing is built, every node should have one road to sink node. In our method, only the active node may be found that it is compromised because all inactive nodes do not send sensing data. So if one active node is detected to be compromised, the routing is broken. It is possible that a neighbor node can't find a road to sink node because it has to circle the compromised node. When the sink node considers that the node is compromised, it has to build the routing for sensor network. However, in this paper, we do not discuss it for the limiting space. However we consider that our method are complementary to the existing route building method.

In this paper, we use subjective logic opinion to quantity the detection of neighbor node. As well known, temporal correlation[5] and spatial correlation[2,3] are the solid ground to find the anomalous data. However, it is a challenge to quantity them. In some existing work, neighbor node often considers that the checked data are absolutely anomalous or not[8,10]. In fact, it is very difficult to give the absolute conclusion, and the uncertainty usually disturbs the detection. To overcome it, we use subjective logic opinion to quantity the detection of neighbor node. In our solution, we try to compute the subjective logic opinion which is able to realistically quantity the view of neighbor nodes.

Our method has a shortcoming. If one node have too many compromised neighbor nodes, these compromised node may disturbs the detection. Suppose that one node has five neighbor nodes, but three neighbor nodes are compromised. So if these compromised nodes are controlled by one attacker, and they send the false alarm messages to sink node. In this time, sink node may draw a false conclusion. Moreover, it is difficult to find the attack with our method. So our method should be used in the network with few compromised nodes.

Background and Our Motivation

It is hard to protect sensor nodes because that they are deployed in insecure area. Limit to the cost, it is hard to improve sensor nodes to protect them. So an attacker is capable of physically capturing one sensor node, and inserting the false data to disturb the sensor network. To overcome it, some solutions are proposed to detect anomalous data with the cooperation of neighbor nodes. For example, as shown in the figure 1, node O has four neighbor nodes including node A, node B, node C and node D. Since they are in close proximity, their sensing data are usually similar, and this is called spatial correlation [2,3]. With spatial correlation, every neighbor node is able to expose the anomalous data sent by node O.

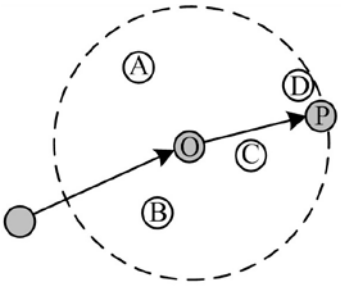

Fig. 1 An example of our method

There is a collection work that detect anomalous sensing data or compromised sensor node with spatial correlation. In the solution proposed by ref [4], every node first checks its data before cooperation-detection. Similar to it, our previous work[5], every node detects its data through temporal correlation. In this paper, we call these methods two-phase methods. With the above example, at the first phase, node O checks its sensing data. At the second phase, neighbor nodes of node O, including node A, node B, node C and node D, cooperate to check the data of node O further. As mention earlier, a sensor node is easy to be physically captured. So the sensor node may be compromised, and it is possible that the node does not check its data as our expectation. Thus, self-detection is not only improve the detection, but also disturb the detection of neighbor nodes.

Our solution is called one-phase solution. When the suspected sensor node is sending the data, its neighbor sensor nodes intercept and check the data. Thus, even the suspected sensor node does not send an alarm message, its neighbor nodes are still able to detect the anomalous data and expose the compromised node. Our method is able to avoid the intervention of compromised sensor node. To quantity the detection of neighbor node, we introduce subjective logic opinion into our solution. Moreover, those subjective logic opinions are fused with the fusing operator to give a final conclusion. With the above example, node A, node B, node C and node D are neighbor nodes for node

O. If node O is the compromised node and send a false data, these neighbor nodes all send the alarm messages to the sink node who fuse these opinions to get the final conclusion.

There is a routing problem for part of neighbor nodes. For example, node A finds node O may be compromised node, and decides to send a message to sink node. However, node O is one active node, and the messages of node A are first sent to node O allowing the current routing. This means that the compromised node is still possible to disturb the detection by refusing or rewriting the messages. So the node A should find another road to sink node. For example, it can ask for node B to transmit the message. However, it is possible that the neighbor node has no other choose to find a new road to the sink node. In our solution, these messages have to be abandoned for simplifying the solution and energy consume.

Our Solution

In this paper, we suppose that sensor nodes are densely deployed. It means that every sensor node should has several neighbor nodes. Otherwise, it is possible that a compromised node has no neighbor nodes to report the alarm message. So our solution fail to expose the compromised node. Before the detection, we suppose that sensor network had built the routing to collect the sensing data from all active nodes. The active nodes always send the messages, and its neighbor nodes may be inactive nodes or active nodes. In our solution, the active nodes are our detecting targets, and all nodes may be the voters for detecting the compromised nodes.

We first discuss the method to produce the neighbor node’s opinion with the example shown in the figure 1. Noted that the opinion is abbreviation of the subjective logic opinion. Since the opinion is a quadruple as denoted as {b,d,u,a}, we discuss how to compute b, d, u and a as following.

The first step is to estimate the future sensing data of routing node that is node O in the above example. To the end, every neighbor node has to store two sensing data of node O. We denote the sensing data as $x(O_{n+1})$, and it means that node O’s sensing data at time t_{n+1}. We suppose current time is t_n. Besides that, every neighbor node has to keep some sensing data for . For example, node A has to keep $x(O_{n-1})$, $x(O_{n-2})$, $x(A_{n-1})$, $x(A_{n-2})$. Now, node A estimates its expected sensing data at t_{n+1} as formula 1.

$$x'(A_{n+1}) = x(A_n) + \frac{x(A_n) - x(A_{n-1})}{t_n - t_{n-1}} \bullet (t_{n+1} - t_n) \quad (1)$$

Next, node A estimates the expected value of node O at t_{n+1} as formula 2.

$$x'(O_{n+1}) = x(O_n) + \frac{x(O_n) - x(O_{n-1})}{t_n - t_{n-1}} \bullet (t_{n+1} - t_n) \quad (2)$$

However, $x'(O_{n+1})$ is computed expected value, it may be inaccurate to real value. We can improve the value with node A’s real value at t_{n+1}. At t_{n+1}, node A has current sensing data denoted as $x(A_{n+1})$, and obtains the distance dis = $x(A_{n+1})$ -x'(An+1). With dis, node A can compute the expected value of node O at t_{n+1}: $x''(O_{n+1}) = x'(O_{n+1}) + dis$.

Now, node A begins to generate its opinion. Node A first computes $c = x''(O_{n+1}) - x(O_{n+1})$ that $x(O_{n+1})$ is intercepted by node A at t_{n+1}. Similar, Node A computes that $d = x'(A_{n+1}) - x(A_{n+1})$. With c and d, Node A obtains u as formula 3.

$$u = \frac{|c-d|}{|c|+|d|} \quad (3)$$

We believe that our method to obtain u is reasonable. In subjective logic, u means the uncertainty of opinion. So the difference between the sensing data of node A and node O can reflect the uncertainty. With u, node A can obtain b. In our opinion, we consider that d is zero, so b=1-u. In this paper, we set a as 0.5. Now, the neighbor node computed the opinion as the alarm message.

When a neighbor node find the active node may be compromised node, it needs to report its opinion to sink node. To this end, the neighbor node has to find a road to sink node which is not including the compromised node. Some neighbor nodes may find a new road to sink node. However, it is possible that some neighbor nodes fail to do that. We discuss this in detail.

Figure 2 shows the example of rebuilding routing for the neighbor nodes. When a neighbor node finds node O may be a compromised node, it begins to find a new road to sink node. In the example, node D may be the node that can send sensing data to sink node because it finds a new active node P to report the alarm message. Similar to node D, node C can find node P as its upstream node. Though node B can't send a alarm message to node P, it can send a message to node C. So node B take node C as its upstream node. Noted that, node C only transmit the alarm message, do not transmit sensing data. The sensing data are still transmit with the previous routing. Unfortunately, node A fails to find any node to report alarm message because node O is the only one upstream node.

To rebuild alarm routing, in sensor network, every node is set one ID. At the initial time, sink node floods one routing message, and these nodes that can receive this message are called 1-jmp nodes. After that, all 1-jmp nodes flood the similar routing message, and these nodes that can receive this message are called 2-jmp nodes. n-jmp is the ID of every node. When one i-jmp neighbor node wants to find a new upstream node, it floods one report message. Any i-1-jmp node that receives the message transmits the report message to the i-2-jmp nodes. When any active node receives the report message, it floods an answer message. The answer message is relayed to i-jmp neighbor node. Thus a new report routing is built. If the neighbor node fails to obtain the answer message in a long time, it considers that it fails to find a new road to sink node.

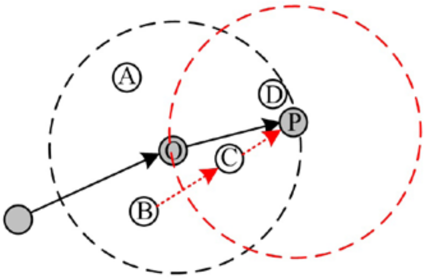

Fig. 2 An example of rebuild routing

There is a special scenery to rebuild report routing. For example, similar to figure 2, neighbor node A and node C are 2-jmp node, and node P is 1-jmp node, and node B is 3-jmp node. We suppose that node A fails to find any 1-jmp node to report the alarm message. However, node B is one node that has build a road to sink node. According to our above rebuild report routing method, node A fails to find node B as the upstream node. The improvement can cover it. However, it may takes more energy to do that. So these neighbor nodes are ignored in our solution. In fact, it is inevitable that some neighbor nodes fail to find a new road to the sink node.

After rebuilding alarm routing, the sink node can receive some alarm messages to expose some compromised node. In this paper, we use fusing operator [6,7] to fuse the neighbor node's opinion for the final conclusion. The fusing rule is shown as followings.

Suppose there are two opinion $w^A=\{b^A,0,u^A,0.5\}$ and $w^B=\{b^B,0,u^B,0.5\}$, the fused opinion is denoted as $w^{A\Diamond B}=\{b^{A\Diamond B},0,u^{A\Diamond B},0.5\}$, and the computed rules is as formula 4

case I. For $u^A\neq 0 \vee u^B\neq 0$:

$$\begin{cases} b^{A\Diamond B} = \dfrac{b^A u^B + b^B u^A}{u^A + u^B - u^A u^B} \\ u^{A\Diamond B} = \dfrac{u^A u^B}{u^A + u^B - u^A u^B} \end{cases}$$

case II. For $u^A=0 \wedge u^B=0$:

$$\begin{cases} b^{A\Diamond B} = b^A \gamma^A + b^B \gamma^B \\ u_{)}^{A\Diamond B} = 0 \end{cases} \quad where\ \gamma^A = \lim\left(\frac{u^B}{u^A+u^B}\right), \gamma^B = \lim\left(\frac{u^A}{u^A+u^B}\right) \tag{4}$$

With the fusing operator, these opinions are fused to one final opinion. On the final opinion, the sink node computes the final score of the checked node. The method is as following: e=b+a*u, where e is the final score. We need to define a threshold value k. If e is bigger than k, we consider that the node may be compromised. Otherwise, the sink node keeps checking the node until no alarm messages about the node are sent to it.

Noted that neighbor node sends the alarm message only when it finds the active node may be a compromised node. It means that not all neighbor nodes report the alarm messages because some nodes consider the node is a normal node. Moreover, as pointed before, some neighbor nodes fail to report the alarm message. So the sink node has to give up some alarm opinions for the detection. Due to the above causes, a sink node may has not enough opinions to draw the conclusion. Moreover, a compromised neighbor node may send a false alarm opinion to disturb the sink node. In our opinion, it is not reasonable to identify a compromised node only with one opinion because we fail to know the opinion is not forged. To overcome it, only more than q neighbor nodes send the alarm message, we identify the compromised node.

Discussion

Our solution overcomes the shortcoming that compromised active node would circle the detection by sending false sensing data. Our solution is one-phase method which the detection is done only with the opinions of neighbor nodes. Thus, even the active node is compromised, it fails to disturb its neighbor nodes and the sink node.

Other advantage of our solution is that our method use subjective logic to quantity the detection of the neighbor nodes. Though our method generating the subjective logic opinion is not optimal, we give one method that transmit the bi-value opinion to subjective logic opinion. In fact, some previous solution directly give the result on the number of yes or no. However, in most of time, one opinion of neighbor nodes usually hold uncertainty. On the other side, subjective logic is used to describe the opinion holding the uncertainty. So we use subjective logic opinion to express the detection.

Our solution is designed to find the compromised active node, but the compromised neighbor nodes may disturb our detection. As pointed out earlier, an active node works smoothly, but its neighbor node has been physically captured which is denoted as node X. Now, the attacker has a way to disturb the network by inserting some false alarm messages with node X. Thus sink node may consider the current active node is compromised, and remove the node from sensor network. To overcome it, a sink node does not consider that one node is compromised only with one alarm message. However, if the node has too many compromised neighbor nodes, our method usually fail to overcome it. So our method is effective that few nodes are compromised in sensor network.

Summary

In this paper, to overcome the shortcoming of two-phase method, we propose one-phase method. In our solution, the neighbor node intercepts the sensing data sent by the target, and generates the subjective logic opinion. To report the opinion, a neighbor node has to find a new road to sink node which circle the compromised node. After collecting enough messages, a sink node fuses these opinions with subjective logic fusing operator. With the fused opinion, a sink node can obtain the anomalous score. If the score is bigger than our preset threshold, we consider the node may be compromised.

Acknowledgement

Corresponding author Jinhui Yuan and her coauthors would like to thank the anonymous reviewers for their insightful comments that helped improve the presentation of this paper. The work is supported in part by the National Natural Science Foundation of China (61303074).

References

[1] J. M. Kahn, R. H. Katz, K. S. J. Pister. Next century challenges: mobile networking for smart dustProceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking. August, 1999.

[2] M. Ding, F. Liu, A. Thaeler, D. Chen, and X. Cheng, Fault- Tolerant Target Localization in Sensor Networks, EURASIP J. Wireless Comm. and Networking, vol. 2007, no. 1, pp. 1-9, Jan. 2007.

[3] Z. Merhi, M. Elgamel, and M. Bayoumi, A Lightweight Collaborative Fault Tolerant Target Localization System for Wireless Sensor Networks, EURASIP J. Wireless Comm. and Networking, vol. 8, no. 12, pp. 1690-1704, Dec. 2009.

[4] Y. Sun, H. Luo, S. K. Das. A Trust-Based Framework for Fault-Tolerant Data Aggregation in Wireless Multimedia Sensor Networks. IEEE Transactions on Dependable Secure Computing, 9(6):785-797, 2012.

[5] J. Yuan, H. Zhou, and H. Chen. Subjective Logic-Based Anomaly Detection Framework in Wireless Sensor Networks. International Journal of Distributed Sensor Networks. Volume 2012, 2012.

[6] A.JØsang. Probabilistic Logic Under Uncertainty. Proceedings of Computing: The Australian Theory Symposium. Ballarat: Australian Computer Society, 101-110, 2007.

[7] A. JØsang, S. Pope, S. Marsh. Exploring Different Types of Trust Propagation. Proceedings of the 4th International Conference on Trust Management (iTrust). Pisa: Springer Verlag, 179-192, 2006.

[8] B. Krishnamachari, S. Iyengar. Distributed Bayesian algorithms for fault tolerant event region detection in wireless sensor networks. IEEE Transactions on Computers. 2004.

[9] D. Cruller, D. Estrin, M. Sivastava. Overview of sensor networks Computer. 2004.

[10] M. Krasniewski, P. Varadharajan, B. Rabeler, S. Bagchi, and Y. C. Hu, TIBFIT: trust index based fault tolerance for arbitrary data faults in sensor networks, in Proceedings of the International Conference on Dependable Systems and Networks, pp. 672C681, Yokohama, Japan, July 2005.