Proc. of Int. Conf. on Current Trends in Eng., Science and Technology, ICCTEST

Preserving Sink Location Against Global Traffic Monitoring Attacker for Wireless sensor Network

Nayana D.V
Dept. of Information Science and Technology, Don Bosco Institute of Technology
nayana0004@gmail.com

***Abstract*—In a WSN, sink-node gathers data from surrounding nodes and sends it to outside world via a gateway. Therefore, its location information is important to both attacker and network operator. The former can launch attacks on a sink-node to steal information or damage it, while the latter must hide its location to ensure data's safety, and physical protection. As the central point of failure, sink location protection is critical to the viability of the whole sensor network. However, existing work related to sink location protection only focuses on local traffic analysis attack. In this paper, we examine the sink location protection problem under a more powerful attack, the global traffic monitoring attack for the first time. In order to hide the sink location, a scheme based on packet sending rate adjustment (SRA) is proposed. By controlling the packet sending rate of each node according to the current number of source nodes, SRA conceals the real traffic volume generated by new source nodes and hence disguises the location of the sink. Theory analysis shows that SRA can protect the sink location against global traffic analysis attack effectively.**

***Keywords*—wireless sensor networks; location privacy preservation; sink location; global traffic attacker**

I. INTRODUCTION

Wireless sensor networks (WSNs) which feature information sensing, data processing and wireless communication have been widely used in military and civilization [1-2]. A typical WSN is composed of hundreds of sensor nodes and one sink. Once a sensor node detects the abnormal event, it becomes the source node (or source) and sends several event packets (known as real packets) periodically to the sink. Then, the sink collects these packets and sends them to the network manager. Such many-to-one communication pattern makes the sink the central point of failure [3]. Therefore, sink damage can cause the whole network useless. So, an attacker would like to destroy the sink physically after tracing and locating it and hence the sensor network will become paralyzed. Thus, it is of great importance to preserve the sink location.

Two kinds of sink location attacks (LTA) [4] including local traffic analysis attack and global traffic analysis attack (GTA) [5] have been proposed to determine the location of sinks. However, existing sink location protection protocols only consider the local traffic analysis attack which can further be classified into packet tracing attack[3], rate monitoring attack [4] and Zeroing-In attack [6]. Both packet tracing attack and rate monitoring attack use fake packet injection to deceive an adversary from tracking the sink [8]. Zeroing-In attack is effective in hiding the sink location information on condition that packets are transmitted by hop information in WSNs. However, none of the previous research focuses on the powerful attacker which has

Grenze ID: 02.ICCTEST.2017.1.143

the global view of the whole network communications.

Defending against the global traffic analysis attack is a challenging problem which hasn't been solved before. Schemes [4-8] under LTA do not help because these schemes cannot make the traffic distributed evenly across the whole network. Therefore attackers in GTA can deduce the location of the sink by monitoring the volume of transmissions caused by the appearance of a new source (or several new sources). A simple solution to defend against GTA is to control the packet sending rate of each node in such a way that every node sends packets at the same rate. However, if sensor nodes send packets at a low rate, the real packets must be delayed seriously. On the contrary, if sensor nodes send packets at a high rate, the communication cost is significantly high. To address these problems, in this paper, we propose a sink location protection scheme based on packet sending rate adjustment (SRA) under the GTA for the first time. SRA sets the packet sending rate of each node according to the current number of sources in WSNs. With uniform packet sending rate across the entire sensor network, SRA can defend against GTA effectively while incurring very low communication cost and the end-to-end latency (the propagation delay from the source to the sink) is acceptable as well.

The rest paper is organized as follows. We present our network and attack models in Section II. Section III proposes our new scheme SRA for sink location protection against GT

II. SYSTEM MODEL

Network Model

There are N evenly distributed sensor nodes and one sink in the whole network. We assume that both the sink and sensor nodes have the same appearance. The sink is assumed to construct the network topology (e.g. building broadcast-tree) by one-time broadcast over the entire network [4]. After that, sensor nodes can send packets hop by hop to the sink by broadcast-tree [4] or random routing based on parent nodes[14]. Furthermore, we assume clock synchronization of the nodes. At any time, there are m ($0 \le m \le N$) sources in the network and the real packet sending rate of each source is R ($R \ge 0$). N denotes the number of nodes in the WSN.

Attack Model

Different from sensor nodes, an attacker has faster computational ability, more storage space, and can communicate with others in a larger range. Several attackers are deployed in the network to launch collusion attack. Specifically, their attacking abilities are as follows:

- *Passive Traffic Monitoring.* The attacker is able to eavesdrop the packet transmissions in a range but unable to decipher packets.
- *Able to Collude*. Several attackers monitor their local traffic separately for a period of time and then move close to share their information. At last, they can infer and obtain the whole network traffic pattern.

III. PACKET SENDING RATE ADJUSTMENT SCHEME

In order to defend against the global traffic attacker, we propose an efficient sink location protection scheme based on packet sending rate adjustment (SRA). SRA firstly investigates the packet sending rate of each node so that low communication cost and low end-to-end latency can be achieved (e.g. In an extreme case, if all real packets are transmitted by one node, the node cannot transmit all these real packets immediately unless its packet sending rate is high enough); Then, SRA creates an uniform packet sending rate for all nodes. Thus, SRA can prevent the attackers with global monitoring ability from tracing the sink while achieving low communication cost and acceptable end-to-end latency. Specifically, SRA includes network initialization phase and packet sending rate adjustment phase.

Network Initialization

In this phase, each node, say *u* initializes a list *Tu* including elements in the form of < *event type*, *number of packets*>, where *Tu*[*event type*].*number of packets* presents the number of real packets must be sent from source to the sink once a node detects an event and becomes the source. As the source sends real packets periodically, *Tu*[*event type*].*number of packets* measures the duration from sending the first real packets to the last one by the source. For instance, temperature and humidity stand for different events. When *u* detects a sudden change of temperature or humidity, the number of packets sent from *u* to the sink is different. Any node, say *v* is also preloaded a sub-interval queue *Lv* which is initialized to NULL. Correspondingly, the sink is also preloaded its sub-interval queue *Lsink* initialized to NULL. Once there is a new source, the sink constructs a packet sending rate variation queue *Tratevariation. Tratevariation* records the packet sending rate adjustment caused by new source(s) appearance.

Packet Sending Rate Adjustment Based on Number of Sources

SRA protects the sink location against the global traffic analysis attack by creating uniform packet sending rate for all nodes. However, one question is how to set the value of the packet sending rate? A high or low packet sending rate can result in high communication cost or long packet end-to-end latency. As illustrated in figure 1, there are three sources including *s*1, *s*2 and *s*3. We can further observe that all real packets generated from these sources are transmitted by one node, say *v*. If the packet sending rate of each sensor is less than 3*R*, some real packets must be delayed at *v*, thereby increasing the end-to-end latency. Theorem 1 proves that given *m* sources in a network, if the packet sending rate is set to *m***R*, low communication cost and end-to-end latency can be guaranteed.

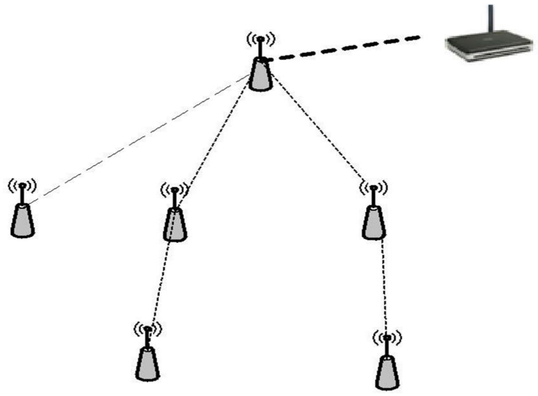


Figure 1. Real packets transmission at node *v*.

According to theorem 1, in order to set an appropriate packet sending rate, the sink must obtain the number of sources at any time. As source nodes appear and disappear randomly, once a node, say *u* becomes a source, the sink does the following three steps:

*(t_s, t_e) Comput*ation:

The sink computes the duration, say (t_s, t_e), of u remaining to be a source

Once the source *u* appears, *u* broadcasts message M_a to inform the whole network. As soon as receiving M_a, the sink computes the time duration, say (t_s, t_e), for *u* according to equation (1) and (2). Parameters including t_{start}, *ð*and T_{ime} stand for the time of receiving M_a at the sink, the time length that the node which is furthest from sink sends a packet to the sink takes and the duration that *u* keeps generating and sending the real packets (that is (T_u[*event type*].*number of packets*-1)/*R*) respectively. equation (1) shows that after all nodes receive M_a, *u* starts to send the first real packet to the sink. equation (2) means that the source is considered to be disappeared after it has sent its last real packet. And then, it becomes a normal sensor which only transmits real packets instead of generating real packets. Here, our “disappearance” is different from the conventional “non-exist”. Since a node may detect events occasionally, it may become a source again and again.

Therefore, it’s possible for it to go through the process from source appearance to source disappearance now and then.

- If *s*>1 as is shown in figure 2 (b), then $l_s.a=l_{(s-1)}.a+1$.
- If ∃ l_j which satisfies that $l_j.t==l_e.t$ as is shown in figure 2 (a) or figure 2 (c), then add l_e to L_{sink} and
  $T_{ratevariation}$, where $l_e.a=l_{(e-1)}.a-1$.

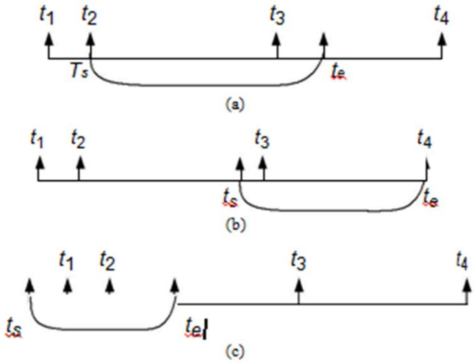


Fig. 2. Major relationships between the appearance duration of *u* and the divided sub-interval

Sub-intervals Division The sink divides (ts, te) into several sub-intervals by algorithm 1 to satisfy that in any sub-interval the number of sources is unchangable.

In order to adjust the packet sending rate of each node, we have to find the sub-interval in which there is the same number of sources. So, we propose an Interval Partition Algorithm Based on Number of Sources (IPAN), as described shown in algorithm 1.

In algorithm 1, the sub-intervals are recorded in queue Lsink by the sink, where Lsink={l1,l2…}, li=<t, a> and li+1.t>li.t. Element li indicates that there are li.a sources since time li.t. Similarly, for node v, the sub-intervals are recorded in queue Lv. Lv={lv,1, lv,2,…} and for □lv,i□Lv, lv,i is in the form of <tv, av>, where av is the number of sources since time tv. Once new source u appears, there are four major relationships between the appearance duration of u and the divided sub-interval according to Lsink.

- If ∃lj which satisfies that lj.t==ls.t as is shown in Fig 2.(a), then lj.a++;
- If ∃lj which satisfies that lj.t==le.t as is shown in Fig 2.(b), then lj doesn't change;
- If ∃lj which satisfies that lj.t==ls.t, then add ls to Lsink and Tratevariation. According to the value of s,following two conditions are considered.
- If s=1 as is shown in Fig 2.(c), then ls.a=1; So, according to the time relationships analyzed above,once a new source appears, the sink does the following three steps.
- For any element belonging to Lsink, say lj, if lj.t□[ls.t,le.t), then the sink updates lj.a to lj.a+1 and adds lj to Tratevariation. This is because since time lj.t, one more source is added in the network due to u's appearance.
- If lj which satisfies that lj.t==ls.t does not exit, then add ls to Lsink and Tratevariation. If s=1, then ls.a=1. And if

s>1, then ls.a=l(s-1).a+1.

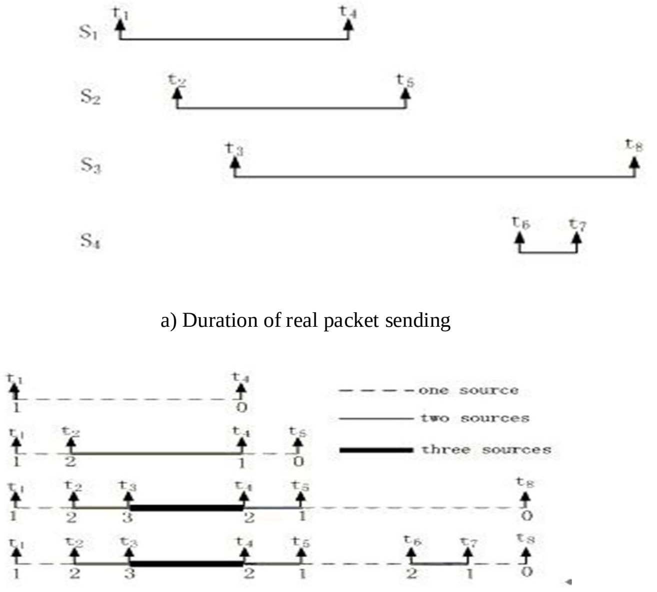
- If lj which satisfies that lj.t==le.t does not exit, then add le to Lsink and Tratechange,wherele.a=l(e-1).a-1.

Packet Sending Rate Setting After obtaining the sub-interval in 2), SRA sets the packet sending rate of each node according to the number of sources at each sub-interval. For example, if there are m' sources in a sub-interval, each node sends packets with the rate m'*R. Specifically, the process of packet sending rate adjustment is as follows.

The sink broadcasts Mb(known as rate adjustment broadcast packet) which includes the packet sending rate variation queue Tratevariation. Once, a sensor, say v receives Mb, v updates Lv according to Tratevariation. Node v changes the packet sending rate to'number of sources'*R atthe 'rate change time' according to Lv (Node v may send an amount of fake packets if there is not enough real packets to be transmitted, so that the packet sending rate can be achieved.).

For instance, figure 3 shows how SRA adjusts the packet sending rate of each node when four sources including s1, s2, s3 and s4 appear one after another. The duration in which each source appears is can be seen in figure 3 (a). Figure 3 (b) shows

the sub-interval division process by TPAN when four sources appear one by one. More specifically, when source *s*1 appears, there is only one source and hence one time interval (*t*1, *t*4) as can be seen in figure 3 (b). After that, *s*2 detects an event and becomes a source which sends real packets during (*t*2, *t*5). Then, the sink divides (*t*2, *t*5) into two sub-intervals: (*t*2, *t*4) and (*t*4, *t*5) according to the number of sources. Similarly, when *s*4 appears, seven sub-intervals have been obtained by algorithm TPAN as shown in figure 3 (b). As a result, the packet sending rate is set to *R*, 2*R*, 3*R*, 2*R*, *R*, 2*R*, *R* and 0 at *t*1, *t*2, *t*3, *t*4, *t*5, *t*6, *t*7 and *t*8.



a) Duration of real packet sending

(b) Sub-interval division

Figure 3. Packet sending rate adjustment

VI. Conclusion

In order to defend against the global traffic monitoring attack, we propose a sink location protection scheme based on packet sending rate adjustment (SRA). By controlling the packet sending rate of each node dynamically, SRA balances traffic over the entire network, conceals the real traffic pattern and hence hides the location of the sink..

REFERENCES

[1] R. El-Badry and M. Younis, “Providing Location Anonymity in a Multi-Base station Wireless Sensor Network,” in *Proceeding of the International Conference on Communications*, 2012.

[2] J. Chen, X. Du, B. Fang, “An Efficient Anonymous Communication Protocol for Wireless Sensor Networks,” *Journal of Wireless Communications and Mobile Computing*, 2011.

[3] K. Mehta, D. Liu, M. Wright, “Location privacy in sensor networks against a global eavesdropper,” in *Proceeding of the IEEE International Conference on Network Protocols*, 2007.

[4] J. Deng, R. Han, S. Mishra, “Decorrelating Wireless Sensor Network Traffic To Inhibit Traffic Analysis Attacks,” *Journal of Pervasive and Mobile Computing*, 2006.

[5] Y. Jian, S. Chen, Z. Zhang, and L. Zhang, “Protecting receiver-location privacy in wireless sensor networks,” in *Proceeding of the International Conference on Computer Communications*, 2007.

[6] U. Acharya and M. Younis, “Increasing base-station anonymity in wireless sensor networks,” *Journal of Ad Hoc Networks*, 2010.

[7] Z. H. Li and W. Y. Xu, “Zeroing-In on Network Metric Minima for Sink Location Determination,” in *Proceeding of the ACM conference on Wireless network security*, 2010.

[8] Y. Jian, S. Chen, Z. Zhang, and L. Zhang, “Protecting receiver-location privacy in wireless sensor networks,” in *Proceeding of the IEEE Conference on Computer Communications*, 2007.