Protecting Sink Location Against Global Traffic Monitoring Attacker

Juan Chen
Dept. of Information Science and Technology
Dalian Maritime University,
Dalian, Liaoning, China
juanchencs@gmail.com

Zhengkui Lin
Dept. of Information Science and Technology
Dalian Maritime University,
Dalian, Liaoning, China
dalianjx@163.com

Xiaojiang Du
Dept. of Computer and Information Science
Temple University
Philadelphia, PA 19121, USA
dux@temple.edu

***Abstract*—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. Simulation results demonstrate that SRA has low communication cost and acceptable end-to-end latency.**

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

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 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 GTA The performance analysis of our scheme is given in Section IV. Section V presents the performance evaluation through simulations. Finally, we conclude the paper with future work in Section VI.

978-1-4673-8579-4/16/$31.00 ©2016 IEEE

II. SYSTEM MODEL

A. 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.

B. 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.

A. Network Initialization

In this phase, each node, say u initializes a list T_u including elements in the form of < *event type*, *number of packets*>, where T_u[*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, T_u[*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 L_v which is initialized to NULL. Correspondingly, the sink is also preloaded its sub-interval queue L_{sink} initialized to NULL. Once there is a new source, the sink constructs a packet sending rate variation queue $T_{ratevariation}$. $T_{ratevariation}$ records the packet sending rate adjustment caused by new source(s) appearance.

B. 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 $3R$, 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.

Theorem 1. *If there are S sources, for any sensor, say u, in order to forward the real packets immediately while achieving low communication cost, r_v should be set as $S*R$, where r_v presents the packet sending rate of v.*

Proof: Given that the packet sending rate of each source is R, if v must transmit real packets from m_v ($0 \le m_v \le m$) sources, v won't delay the transmission of any real packets on condition that $r_v \ge m_v*R$. Considering the worst case, if real packets from all sources must be transmitted by v, then we have $m_v=m$. Thus, only if r_v is set equal to or more than $m*R$, v can forward all real packets immediately. However, the communication cost increases as r_v increases. Therefore, if $r_v=m*R$, v guarantees immediate packet transmission with low communication cost.

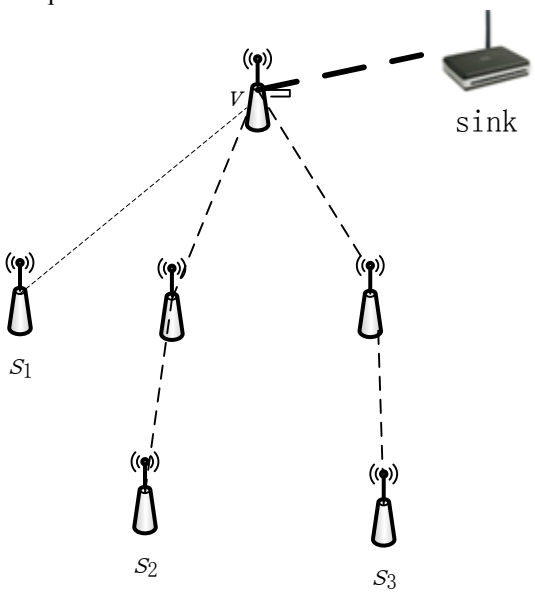


Fig. 1. Real packets transmission at node v.

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:

1) (t_s, t_e) Computation: 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}, δ 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.

$$t_s = t_{start} + \delta \quad (1)$$

$$t_e = t_s + T_{ime} \quad (2)$$

2) Sub-intervals Division The sink divides (t_s, t_e) 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 L_{sink} by the sink, where $L_{sink}=\{l_1,l_2\ldots\}$, $l_i=<t, a>$ and $l_{i+1}.t>l_i.t$. Element l_i indicates that there are $l_i.a$ sources since time $l_i.t$. Similarly, for node v, the sub-intervals are recorded in queue L_v. $L_v=\{l_{v,1}, l_{v,2},\ldots\}$ and for $\forall l_{v,i}\in L_v$, $l_{v,i}$ is in the form of $<t_v, a_v>$, where a_v is the number of sources since time t_v. Once new source u appears, there are four major relationships between the appearance duration of u and the divided sub-interval according to L_{sink}.

- If $\exists l_j$ which satisfies that $l_j.t==l_s.t$ as is shown in Fig 2.(a), then $l_j.a$++;
- If $\exists l_j$ which satisfies that $l_j.t==l_e.t$ as is shown in Fig 2.(b), then l_j doesn't change;
- If $\exists l_j$ which satisfies that $l_j.t==l_s.t$, then add l_s to L_{sink} and $T_{ratevariation}$. According to the value of s, following two conditions are considered.
- If s=1 as is shown in Fig 2.(c), then $l_s.a$=1;
- If s>1 as is shown in figure 2 (b), then $l_s.a=l_{(s-1)}.a$+1.
- If $\exists l_j$ 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.

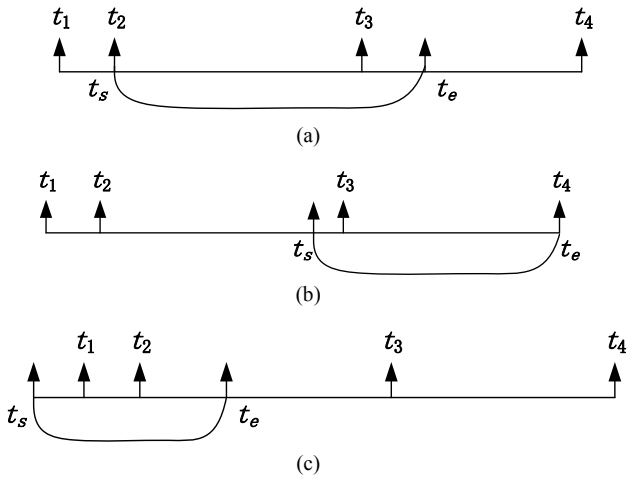


Fig. 2. Major relationships between the appearance duration of u and the divided sub-interval

So, according to the time relationships analyzed above, once a new source appears, the sink does the following three steps.

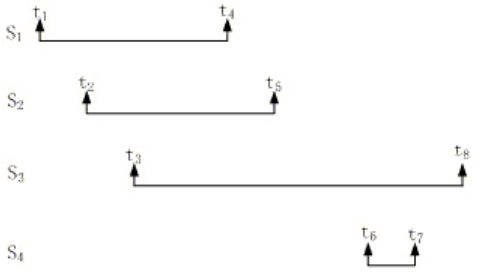
- For any element belonging to L_{sink}, say l_j, if $l_j.t\in[l_s.t, l_e.t)$, then the sink updates $l_j.a$ to $l_j.a$+1 and adds l_j to $T_{ratevariation}$. This is because since time $l_j.t$, one more source is added in the network due to u's appearance.
- If l_j which satisfies that $l_j.t==l_s.t$ does not exit, then add l_s to L_{sink} and $T_{ratevariation}$. If s=1, then $l_s.a$=1. And if s>1, then $l_s.a=l_{(s-1)}.a$+1.
- If l_j which satisfies that $l_j.t==l_e.t$ does not exit, then add l_e to L_{sink} and $T_{ratechange}$, where $l_e.a=l_{(e-1)}.a$-1.

3) 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.

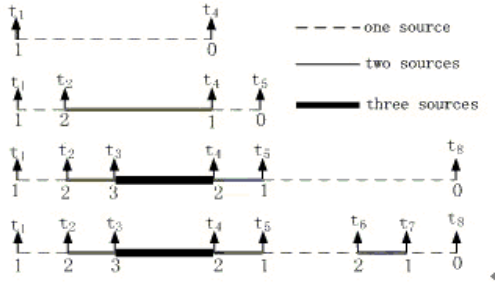
The sink broadcasts M_b (known as rate adjustment broadcast packet) which includes the packet sending rate variation queue $T_{ratevariation}$. Once, a sensor, say v receives M_b, v updates L_v according to $T_{ratevariation}$. Node v changes the packet sending rate to '*number of sources*'**R* at the '*rate change time*' according to L_v (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 s_1, s_2, s_3 and s_4 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, $2R$, $3R$, $2R$, R, $2R$, 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

Fig. 3. Packet sending rate adjustment

Algorithm 1 TPAN

Input: L_{sink}={<t1, a_1>, <t2, a_2>,…}, (t_s, t_e)
Output: $T_{ratevariation}$

```
T_ratevariation = ∅
for any element in L_sink    //Update T_ratevariation
  if l_j.t ∈ [l_s.t, l_e.t), then
    Update l_j.a to l_j.a+1;
    Add l_j to T_ratevariation;
  end if
  if l_j which satisfies that l_j.t==l_s.t does not exit, then
    Add l_s to L_sink and T_ratevariation;
    if s=1, then
      l_s.a=1;
    end if
    if s>1, then
      l_s.a=l_(s-1).a+1;
    end if
  end if
  if l_j which satisfies that l_j.t==l_e.t does not exit, then
    l_e.a=l_(e-1).a-1;
    Add l_e to L_sink and T_ratevariation;
  end if
end for
```

IV. Performance Analysis

The performance of SRA including security, communication cost and end-to-end latency will be analyzed in this section.

A. Security performance

SRA includes two phases: network initialization and the packet sending rate setting based on number of sources. Without loss of generality, we just analyze the security performance for the latter here because the former is one time and thus the sink is considered to be safe.

We only consider the packet transmission process because attackers trace the sink by traffic analysis. When there is a new source, say u, there are three packet transmission processes according to subsection III.

- The source broadcasts M_a first and it is obvious that this broadcast process won't reveal the sink location.
- The sink broadcasts M_b when the broadcast process of Ma may haven't been finished. Therefore, it is difficult for attackers to tell M_b from M_a and hence to infer the location of sink.
- At last, each node sends packets with the same rate. With evenly traffic distribution, the sink location is well preserved.

B. Communication cost

The communication cost of SRA is the total number of packets generated during three packet transmission processes as are stated above.

When a new source appears, the number of packets generated for the broadcast initialized by source or sink is N. For evenly traffic distribution, $T_{ime}*R$ packets (including fake and real packets) should be generated by one sensor. Hence, the communication cost for the N sensors is $T_{ime}*R*N$ for traffic balance. In all, the communication cost of SRA is $(2+T_{ime}*R)*N$.

C. End-to-end latency

We use the number of hops that a real packet takes from the source to the sink on average to measure the end-to-end latency. Different from routing protocols without sink location protection schemes against GTA, SRA brings about extra end-to-end latency which is $2h_{max}/T_u$[*event type*].*number of packets* due to the source and sink broadcasts, where h_{max} is the largest shortest hops that a node may have in WSN. Therefore, the end-to-end latency of SRA is $h_{u,sink}+2h_{max}/T_u$[*event type*].*number of packets*.

V. Simulation Results

We evaluate the performance of SRA in view of communication cost and end-to-end latency using a simulator written in C++ by us. Given that there are 1024 nodes and each node has 8 neighbors on average, we divide the square network area into grids with one node randomly placed in one grid. And the sink is placed randomly. At any moment, a node becomes

the source with some probability. For simplicity, only one kind of event is considered and the source will send 5 real packets to the sink at the rate of R=1. We compare our SRA with the 'Baseline' scheme. Baseline represents the simple solution by which each node sends packets at a pre-defined rate (Different from SRA, without obtaining the number of sources, the packet sending rate is usually set high to achieve better sink location protection without increasing extra end-to-end latency of real packets.).

Figure 4 shows the communication cost comparison between SRA and Baseline. We investigate the cases where one to five sources appear one after another. Here, the packet sending rate of Baseline is assigned with 5 in order to conceal the real traffic distribution when there are 5 sources. Note that in real cases, the packet sending rate of Baseline should be set to a higher number without knowing the current number of sources because the real packets the sink protection performance in Baseline improves as its packet sending rate increases. To capture the average trend, we repeat our experiments 50 times with different sink locations, and then take an average. As illustrated in figure 4, the communication cost of Baseline doesn't change with the increase of number of sources. However, the communication cost of SRA increases as the number of sources increases. Moreover, it is obvious that the communication cost of our SRA is always and far less than the Baseline. This is because without the number of sources, the packet sending rate in Baseline is usually set high to achieve better sink location protection and low end-to-end latency which results in higher communication cost.

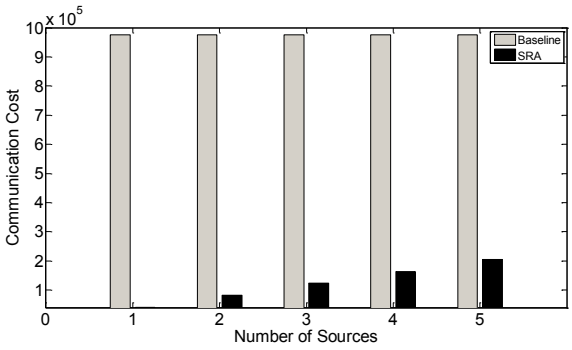


Fig. 4. Communication cost comparison between different schemes.

Figure 5 shows the end-to-end packet latency with the increasing number of average hops from the source to the sink under different schemes. As can be seen in Figure 5, the end-to-end packet latency is slightly higher (two hops on average) than that of Baseline. This is because in order to obtain the number of sources in different interval, both the new source and the sink need start a broadcast which results in extra end-to-end latency in SRA. Note that, though the end-to-end latency of SRA is a little more than that of Baseline, SRA achieves the perfect sink location protection and too much lower communication cost while Baseline doesn't.

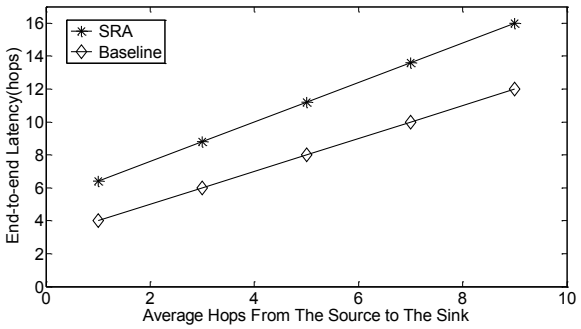


Fig. 5. End-to-end packet transmission latency under different schemes.

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. The effectiveness of SRA is validated by theory and experiment results. Future work will focus on sink location protection in mobile wireless sensor networks.

ACKNOWLEDGMENT

This research is supported in part by the Natural Science Foundation of China under grants No. 61300188, 61301131, 61301132 and 61203082; by the Fundamental Research Funds for the Central Universities No. 3132014209; by Liaoning Province Science and Technology Plan Program No. 2011402003 and National Key Technology R&D Program No. 2012BAF09B01; by Scientific Research Projects from Education Department in Liaoning Province No. L2015056.

REFERENCES

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

[2] 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.

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

[4] 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.

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

[6] 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.

[7] 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.

[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.