# “Preserving Source Location Security based on Fake Sources”

Ulugbek Adilbekov, Anar Adilova
Computer Science Department
Nazarbayev University
Kazakhstan, Astana
ulugbek.adilbekov@nu.edu.kz
anar.adilova@nu.edu.kz

ABSTRACT:

Wireless Sensor Network is a wireless network consisting of distributed sensor devices that can monitor physical and environmental conditions and pass them to each other. Adversary can follow the traffic and eventually determine location of the source. Therefore, source location privacy is an issue and plenty of good solutions have been proposed to deal with it. In this paper, we will assess Probabilistic Algorithm and propose improved version which works better in particular cases.

## INTRODUCTION:

Wireless Sensor Networks (WSN) are widely used in real-time event monitoring. Nodes collect data and continuously send packets to the sink node. Sink node is a node with greater computing power which analyses collected data. Source location privacy is one of the major security issues related to WSN. While content security can be achieved by using encryption algorithms, context security is harder to deal with due the fact that sensor devices have low-cost and low-power and the network itself has limited bandwidth. Context security scenario can be explained with the help of classical panda-hunter scenario. Suppose, there is a panda and a hunter. Nodes sense panda and send packets to sink node. At the same time, hunter wants to detect the location of panda by tracing packets from the sink back to source nodes. In order to protect panda from hunter we need to hide the location of source nodes.

Fake sources is one of the ways to achieve source location security. The basic idea is that fake sources send dummy messages to mislead the adversary. One of the source location oriented algorithms based on the idea of fake sources is Probabilistic Algorithm (PBA).

## PROBABILISTIC ALGORITHM:

In PBA, when a node has a data to transmit it can only send it at a time defined by its pseudo random number generator (PRNG), however when a node does not have data to transmit it should send a dummy packet with a probability p. Time between 2 sends is between some a and b. Each node is aware of PRNG seeds of its neighbors. When choosing a node to send a packet to, preference is given to nodes which are closer to sink and among them to nodes which are expected to send their packets earlier than others. Node can use PRNG seeds to compute send times for its neighbors.

This algorithm is very versatile because of 2 reasons:

1. By changing probability p, we can vary our security and radio overhead. The highest security is achieved when p = 1, but sometimes we might want to have lower radio overhead instead of superb security.
2. By changing the minimum and maximum bound for waiting times between sending messages we can vary between throughput of the network and power consumption.

In order to assess the algorithm’s effectiveness and strength, we will simulate it using Contiki with the help of Cooja simulator. Contiki is an operating system that connects low-cost, low-power microcontrollers. Cooja is a network simulator which is designed for Wireless Sensor Networks.

## ADVERSARY MODEL:

Here we give details of our adversary model. The attacker knows location of every node and has a global view of the network. The attacker can hear all on-going communication in the network. He has sufficient resources to know from a message who originated the message and who it is addressed to. The attacker can analyze any contextual information about collected data, but cannot understand data itself (we assume that data can be encrypted). It might know network is using PBA and all constants. Attack will be performed according to the following algorithm:

1) Collect data for some time T
2) Start from the sink node.

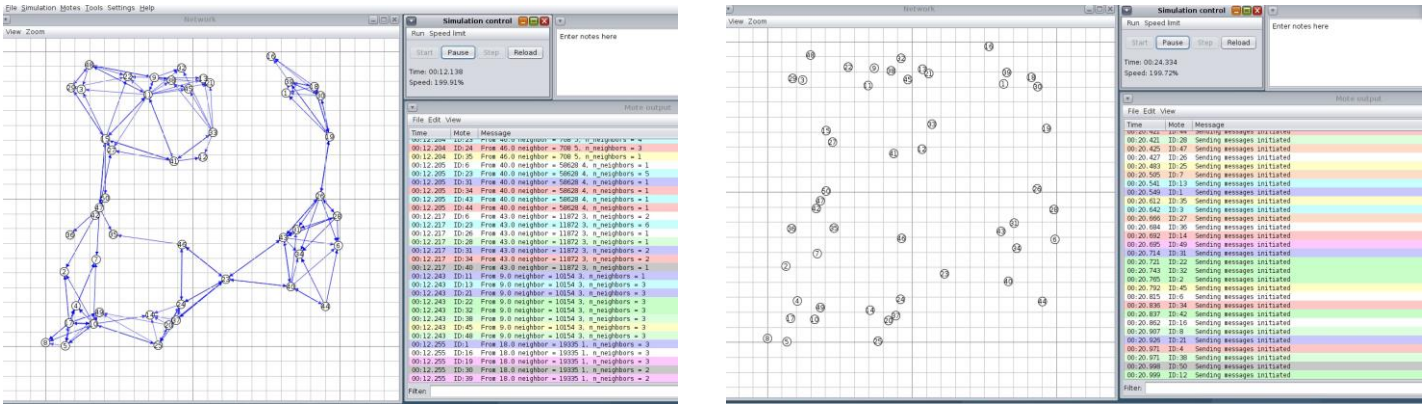

Figure 1. PBA Contiki Simulation

3) Analyze data and if there is a node which transmits more traffic than others move to it.
4) If all the nodes transmit almost equal amount of traffic then stop, this might be source.
5) Else go to step 2.

NETWORK MODEL:

Network model for the simulations is as follows. It consists of one sink node and a source node. The total number of nodes is 50. Intermediate nodes can receive and send data towards the sink node. Data is divided into dummy and real. All the nodes, except for sink, can generate dummy messages.

Figure 1 demonstrates sample simulations of PBA.

PROPOSED ALGORITHM:

We noticed that PBA has a use case where it is not as efficient as it could be. When adversary manages to steal small amount of data, it could be the case that for many nodes the following could be true: if there few cases when a node should decide whether to send a dummy message or not, then the node's distribution of "sends" and "not sends" would not satisfy probability p. While this is not dangerous if the number of "sends" is higher than it should be, it is when number of "not sends" is greater than it should be, because we rely on dummy messages to confuse adversary.

We could address this vulnerability by changing the mechanism of decision whether to send a node or not. p will no longer be a constant probability. We want any sequence of sequential decision a node makes during its lifecycle to have distribution of "sends" and "not sends" to be as close as possible to p, that is to converge to p.

In our experiments, we are going to use the following mechanism. Each node will have $cur\_p$ which is responsible for making upcoming decision. If node decides to send a dummy message, then $cur\_p$ should be decreased by $y_{dec}$, if it decides to not send a dummy message then $cur\_p$ should be increased by $y_{inc}$.

In our experiments we picked $y_{dec}$ and $y_{inc}$ in order to have certain value of variance. That can be described by the following formula

$$y_{dec} = variance * p + variance$$
$$y_{inc} = variance - y_{dec}$$

RESULTS

| Time(seconds) | PBA | PBA_IMPROVED |
|---|---|---|
| 100 | 118,42 | 123,3 |
| 500 | 576,93 | 608,99 |
| 1000 | 1146,66 | 1211,98 |
| 6000 | 6723,12 | 7267,91 |

Table 1. Number of messages per time interval

| Time(seconds) | PBA (%) | PBA_IMPROVED (%) |
|---|---|---|
| 100 | 3,66 | 4,33 |
| 500 | 16,75 | 8,25 |
| 1000 | 17,1 | 6,75 |
| 6000 | 30,2 | 28,45 |

Table 2. Source Detection Rate by Adversary

DISCUSSION:

The results of the experiment correspond with our expectations. We conducted the experiments only for 2 criteria: security and power consumption. Our proposition does not

change the way a node choses its next node on the way to sink, thus the delivery time from source to sink should not be different from original.

In terms of security, PBA with our proposition was more successful than original PBA on small time samples (T = {500, 1000}), exactly how we expected it to be. For T = 100, there was no difference, mostly because there was too small amount of messages flowing, so the adversary was mostly random in that case. For T = 6000, we also observed little difference, since the time range is already large enough to have original PBA's nodes' decisions converge to p.

We measure number of messages flowing in the network as a magnitude of power consumption. The results have shown that our proposition does not introduce significant overhead in number of messages. This can be explained by the fact that in both versions probability sending dummy message converges to p.

Overall, we advise anyone who is willing to use PBA to take into account our proposition to increase security of their WSN.

REFERENCES:


1) Conti, M., Willemsen, J., & Crispo, B. (2013). Providing Source Location Privacy in Wireless Sensor Networks: A Survey. IEEE Communications Surveys & Tutorials, 15(3), 1238-1280. http://dx.doi.org/10.1109/surv.2013.011413.00118

2) Ouyang, Y., Le, Z., Liu, D., Ford, J., & Makedon, F. (2008). Source Location Privacy against Laptop-Class Attacks in Sensor Networks. Securecomm 2008. Retrieved from http://delivery.acm.org/10.1145/1470000/1460884/a5-ouyang.pdf