ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Providing Source Location Privacy in WSN using Random Route Adoption Method

Abhilash N
PG Student, CNE, The National Institute of Engineering, Mysore, India

Abstract
Abundant applications of sensors made our world even smarter than it was. Wireless sensor network (WSN) is an assembly of self-organized motes which are deployed to examine various environmental conditions. This paper proposes a new approach in providing both content and context privacy of the source node in a sensor network. This scheme uses a cryptographic operation between two neighbours to achieve context privacy and a new two level random route algorithm to counter hop by hop traces and other adversarial activity.

Keywords
Wireless Sensor Networks, Source Node, Context Privacy, Mote, Random Route.

I. Introduction

Wireless sensor networks are attracting great interest in a number of application domains concerned with monitoring and control of physical phenomena, as they enable dense and untethered deployments at low cost and with unprecedented flexibility.
The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one or several sensors as shown in figure 1

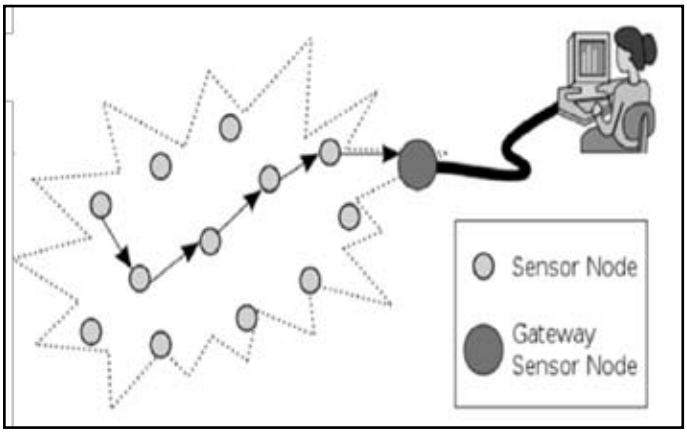

Fig. 1: Wireless Sensor Network

Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.
Talking about the connectivity part, WSNs usually prefer UDP over TCP. TCP is a connection-oriented protocol. However, in WSNs, the number of sensed data for event-based applications is usually very small. Even though TCP is one of the most important protocols of Internet Protocols suite and it is most widely used protocol for data transmission in communication network it's still not preferred, as the user can multicast or broadcast packets which can't be done in TCP and also UDP is faster than TCP.
Leaving the limitations behind we focus on some of the other features of UDP which are
It is used when acknowledgement of data does not hold any significance.

- It is good protocol for data flowing in one direction.
- It is simple and suitable for query based communications.
- It is not connection oriented.
- It does not provide congestion control mechanism.
- It does not guarantee ordered delivery of data.
- It is stateless.
- It is suitable protocol for streaming applications such as VoIP, multimedia streaming.

In this paper, we are interested in tracking and monitoring applications, such as wildlife tracking and monitoring the movement of doctors and patients in a hospital [1]. The architecture of a WSN in these applications is as follows: nodes monitor an area and look for the presence of a certain type of object of interest, which we call the subject.
A subject can be anything, depending on the application, such as a human, an animal or a vehicle. When a node senses the subject, it informs one or more sinks. A sink is a node that has more storage capacity, computation power, and a better power supply. It is able to do a lot of computational tasks that normal nodes are mostly not capable of. The sink collects all the data of the WSN and either sends this to a server, or it allows for the manual extraction of the data. The moment that a node senses the subject is called an event. Whenever an event occurs, the sensor senses the change in environment and collets the data and passes this data to the central base station.

II. Related Works

Many researches has been made on the interest of providing SLP and there is a broad spectrum of solutions available based on the techniques such as network coding, random walks, sending dummy packets, nodes acting as fake source and so on.
Most of the solution to provide SLP will either prefer RW or using fake packets or sometimes both. [2] Proposed a method which involves in 3 phases routing. In the first phase it uses pure random walk where the source uses completely random path to reach the sink. The next phase involves forwarding random walk

where the source chooses a node randomly from the forwarding list as its next hop which is less or equal cost to reach the sink. In its final phase credit based routing is used where the minimal cost from every sensor to reach the sink node is calculated and selects a node with the least cost.
Another solution which is little deviated from the above is [3] intends to provide SLP. Here multiple fake source is created to create anonymity for the advisories. Fake sources are created as soon as an event is detected around the neighbors and forwards fake packets. The advisories are drawn away from the actual source due to this extended path, thereby providing SLP.

III. Proposed System

A sensor node can communicate with its neighbours in the range of radio. Whenever a sensor receives a packet, it broadcasts the message or simply discards it. Adversaries can eavesdrop on communications between sensors. It is assumed that an adversary has the same eavesdropping range as the radio communication range of sensor nodes [4].
Although an adversary cannot obtain the exact content of the messages intercepted, the direct sender of the messages can be determined using traffic analysis or RF localization techniques. Adversaries overhear at the base station at the very beginning. When intercepting a message, it moves to the location where the message came from. Then, it eavesdrops on the communications between the current node and its neighbouring nodes until backtracking to the source hop-by-hop.
In general, adversaries have much larger storage than sensor nodes. So, we assume that they would record every location they have been to [5]. Only in this way can the adversary avoid getting into circulations generated by fake sources that probably exist in the network. Circulations make it difficult for adversaries to track the source. In order to cope with this situation, a sophisticated adversary will check the historical locations after determining where a message comes from.
Only if the message comes from a completely new sensor, the adversary would move to that node. Otherwise, it would ignore the message and keep listening at the current location. It is possible that a patient adversary cannot overhear anything for a long time. In this case, the adversary may roll back to the latest one among the recorded locations. Then this location will be removed from the record of historical locations. As depicted in Fig. 2.

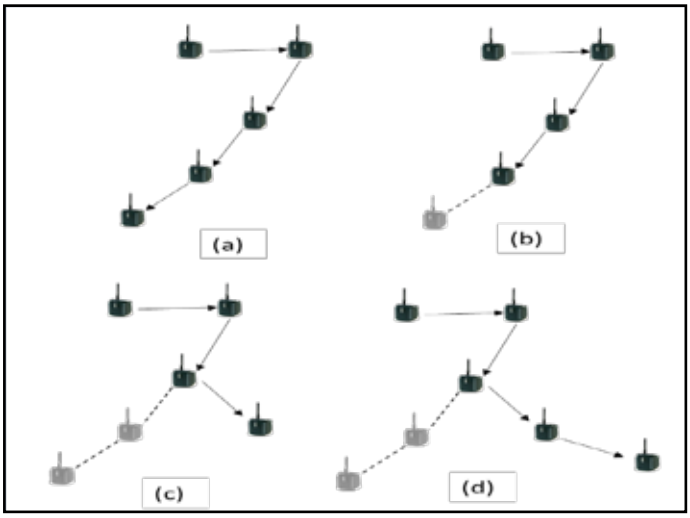

Fig 2: Attacker Module

An attacker sits on a sensor node and listens to data event , when he listens an event transfer in his neighboring node, ne moves to that node and listens for further event and reches the source node, if no event occurs, he will back track to his previous position and moves to other neighbor where the event is happening. The algorithm for a smart historical location attack is given below

Algorithm 1. Strategy used by sophisticated adversaries
1: An adversary moves from node A to node B after it overhears a message;
2: The adversary starts the timer and sets the timeout interval to be δ_;
3: **while** (keep listening at node B) **do**
4: **if** (overhear a message) **then**
5: determine which node it is from, say C;
6: **if** (C=A *or* C=B *or* historicalLocations.find(C)) **then**
7: drop the message;
8: **else**
9: historicalLocations.push(C);
10: move to node C;
11: **break**;
12: **end if**
13: **else if** (timer timeout) **then**
14: node X = historicalLocations.top();
15: historicalLocations.pop();
16: roll back to node X;
17: **break**;
18: **else**
19: do nothing;
20: **end if**
21: **end while**

The entire process starts when BS broadcasts hello packet. The neighboring node receives this packets and updates their routing table and forwards down the lane. Visited IP list is maintained to avoid loop backs. When event occurs, sensor collects data and has all possible routes to send this event data.
Base station sends s data encryption key to encrypt the sensed data. BS encrypts this key with the public key [6] of the sensor and sends back as an acknowledgment for the registration message sent by the sensors. The sensors decrypt this message using its own private key.
When a data has been sensed, the data has to be encrypted using data encryption key, Sensor picks a route randomly using two level random route algorithm. Before sending any data secure key sharing has to be done between two nodes. Node1 performs a modulus operation with its secret key and sends it to node2. Node2 performs the same modulus operation with its secret key and sends this to node1.
Now both node1 and node2 performs a final modulus operation with each other's private key. This results a shared secret key [7] where node1 uses to encrypt the sensed data and node2 decrypts the data. This entire process repeats all the way to the base station and finally base station decrypts the data by its data decryption key shared in the initial stage. Hence both content and context privacy is maintained.
The algorithm for the two level randomization algorithm is shown below,
Input: Hash table with routes and their IP's
Output: Random RouteIpList

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

1. **for each** KeyValueList in Sorted hashTable
2. **start**
3. **if** itteration == 0
4. store CurrentRandomIpList with the 1st KeyValueList
5. **else**
6. **start**
7. declare CurrentRepeatCounter, NewRepeatCounter
8. **foreach** items in PreviousRandomIpList
9. **start**
10. Store CurrentRepeatCounter with total no. of cluster repeat in CurrentRandomIpList
11. Store NewRepeatCounter with total no. of cluster repeat in KeyValueList
12. **end**
13. **if** NewRepeatCounter < CurrentRepeatCounter,
14. Replace CurrentRandomIpList with KeyValueList
15. **end**
16. CurrentRepeatCounter+=1;
17. PreviousRandomIpList = CurrentRandomIpList;
18. **end**

This algorithm tries to find a new random route by switching the cluster that has been used in the previous route as the first level randomization and it switches the node repeated in the same cluster as the second level of randomization. Hence an attacker will find an hard time to trace back to its original source node, which also provides a safety period for any entity to move away from the source node.

IV. Conclusion

Source-location privacy protection is a significant security property of sensor networks used to collect information about monitored objects in military or endangered species-monitoring applications. This paper combines cryptographic method and a random route adoption method in order to provide source location privacy. This method pre computes the routes from the random route adoption algorithm so that all the neighbours will be known in prior and they won't leak any data to the attackers. Since all the routes are equally used during the event data transmission the total life time of the network is improved. Since the forwarding probability of a node is unequal, this method promises to provide better source location privacy than pure random walk method.

References

[1] Mauro Conti ; University of Padua, IT ; Jeroen Willemsen; Bruno Crispo "Providing Source Location Privacy in Wireless Sensor Networks: A Survey", 1238 – 1280 ISSN : 1553-877X 28 January 2013

[2] Zongqing Lu ; Nanyang Technological University ; Yonggang Wen "Credit Routing for Source-location Privacy Protection in Wireless Sensor Networks," Page(s): 164 – 172 ISSN : 2155-6806 Las Vegas, NV.

[3] Kanthakumar Pongaliur and Li Xiao Computer Science and Engineering Michigan State University, East Lansing, Michigan, U.S.A. "Maintaining Source Privacy under Eavesdropping and Node Compromise Attacks" Page(s): 1656 – 1664 ISSN : 0743-166X, Shanghai.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," Comput. Netw., vol. 38, no. 4, pp. 393–422, Mar. 2002.

[5] Wei Tan ; Department of Computer Science and Technology, Tsinghua University, Beijing, China ; Ke Xu ; Dan Wang "An Anti-Tracking Source-Location Privacy Protection Protocol in WSNs Based on Path Extension", Page(s): 461 – 471 ISSN : 2327-466212, August 2014.

[6] Sourabh Chandra ; Department of Computer Science & Engineering, Calcutta Institute of Technology, Kolkata, India ; Smita Paira ; Sk Safikul Alam ; Goutam Sanyal "A comparative survey of Symmetric and Asymmetric Key Cryptography", Page(s): 83 – 93, Conference Location : Hosur.

[7] Imre Csiszár ; A. Rényi Institute of Mathematics, Hungarian Academy of Sciences, H-1364 Budapest, Hungary; Prakash Narayan "Capacity of a shared secret key", Page(s): 2593 – 2596 ISSN : 2157-8095, Conference Location : Austin, TX