Destination Controlled Anonymous Routing in Resource Constrained Multihop Wireless Sensor Networks

Alireza A. Nezhad[1], Dimitris Makrakis[1], Ali Miri[1]

[1] University of Ottawa, Ottawa, Ontario, Canada
{nezhad, dimitris, samiri}@site.uottawa.ca

Abstract. In this paper, a routing protocol is proposed that provides location privacy for the source and the destination as well as user anonymity and unlinkability in multihop wireless sensor networks. The sink is assumed to be computationally powerful and responsible for all routing decisions. It assigns incoming and outgoing labels to nodes in the uplink and downlink directions. Each node is only aware of its own labels and only forwards packets whose labels match either its downlink or uplink incoming label. Moreover, in order to prevent packet tracing by a global eavesdropper, layered cryptography is used in both directions to make a packet look randomly different on different links. However, due to the node capability limitations, only symmetric cryptography is used.

Keywords: Wireless Sensor Networks, Location Privacy, Anonymity, Routing

1 Introduction

Wireless sensors are characterized by limitations on batteries, computational capabilities (CPU power and memory), as well as communication capabilities such as bandwidth, transmission power and receiver sensitivity. In the early days of wireless sensor networks (WSN), the issue of network security was given second priority as the technology struggled to meet these strict and diverse constraints. It is only recently that a flurry of activities has been seen in some areas of wireless sensor networking including lightweight cryptography, secure routing and intrusion detection. However, *anonymous routing* that covers areas such as *node anonymity, node location privacy, untraceability and unlinkability* is yet to be adequately explored. These issues are of utmost interest in military, homeland security, and law enforcement but are also becoming progressively essential to many civilian applications.

Fig.1 demonstrates the importance of location privacy for the source and the destination in wireless communications. In this example, a criminal is interested in finding a valuable object (source) by using the beacon signals that it emits (similar to asset monitoring applications). It is also interested in identifying the locations of police patrols (destination, also called *sink* in WSN) in order to avoid or even eliminate them.

An anonymous routing protocol for WSN must prevent an adversary from finding the locations of the source and the sink. The adversary may exploit information in the packet headers or perform packet tracing. The former problem can be addressed

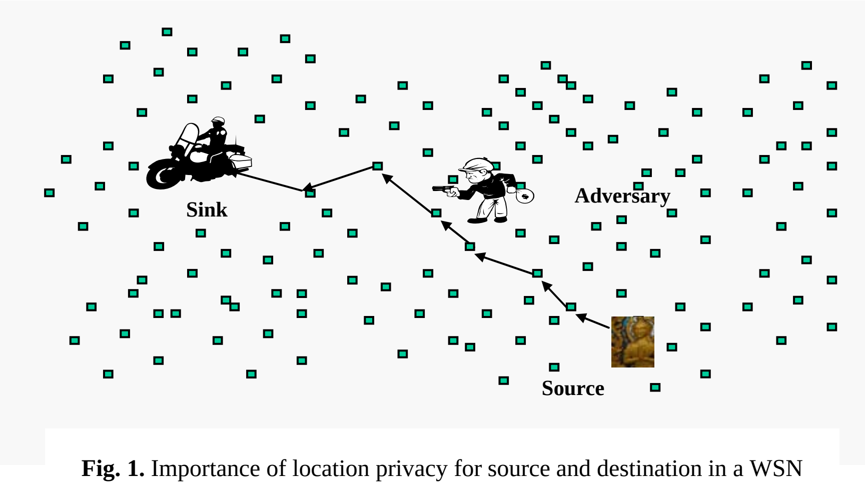

Fig. 1. Importance of location privacy for source and destination in a WSN

using encrypted packet headers or identity-free routing, while the latter must be addressed by untraceable routing schemes. According to Pfitzmann and Kohntopp [11], anonymity is the state of being un-identifiable within a set of objects; the *anonymity set*. Untraceability refers to the inability of an adversary in tracing individual data flows back to their origins or destinations. Unlinkability means preventing an adversary from learning the identities of the source and the destination at the same time.

Regular routing protocols designed for *multihop* wireless sensor networks [7] are vulnerable to *location privacy* violations. In order to achieve the highest possible efficiency, these protocols use methods that unfortunately offer a lot of help to an attacker. These methods are usually based on disclosing network topology, especially the location of the *sink*. Usually, the sink emits periodic beacon signals that are flooded in the network to let each sensor calculate the best (e.g. the shortest) path to the sink. Moreover, during data transmission, an adversary can trace data packets back to their source and/or their destination over consecutive links. This can be done using signal detection techniques such as triangulation, trilateration, angle of arrival, signal strength and so on in each locality to find the immediate sender of a packet.

Encryption is undoubtedly one of the most effective means of providing security services. However, encryption alone is not enough for ensuring anonymous communications. Adversaries can use *traffic analysis* techniques such as content analysis and timing analysis to obtain valuable information including the locations of the source and the destination of each packet as well as the identities of the nodes involved. Traditionally, anonymous communication schemes such as Mixes [12] have been used to protect networks against such attacks. Unfortunately, these solutions, even those versions developed for MANET [8], are not applicable to resource-constrained sensor networks. Lately, several anonymous routing protocols [1]-[6] have been introduced for wireless ad hoc sensor networks, some of which attempt to defend against this kind of attack. In the next section, we briefly overview some of these proposals and point out their shortcomings.

In this paper, we introduce a novel routing scheme for sensor networks based on label switching that supports traffic untraceability in order to hide the locations of a source sensor and the sink. It is also an identity-free routing protocol and as such offers anonymity to the source and the sink. The identity-free routing and untraceability ensure unlinkability even when multiple sinks use the same sensor network.

The rest of this paper is organized as follows: in the next section, we review some related works. In section 3, we describe our network model. In section 4, we explain our threat model and privacy objectives. In section 5, we present our routing protocol. Finally, we will conclude with a summary.

2 Related Work

Phantom routing [1] is one of the earliest works that clearly addresses the problem of source location privacy in wireless sensor networks. It considers that movements of an object are continuously monitored by a network of stationary sensors and that an adversary tries to locate the object by tracing the stream of packets flowing from its point of presence towards the sink. In order to mislead the adversary, phantom routing first sends each packet from the source randomly to an intermediary node called a *phantom source* located a number of hops away from the real source. The phantom source then either floods the received packet in the network or unicasts it to the sink. Obviously, the flooding version of this protocol does not reveal the location of the destination but it has the disadvantages of flooding, including excessive energy and bandwidth consumption. On the other hand, its single-path version cannot hide the location of the sink as it uses regular single-path routing schemes that require all sensors to know where in the network the sink resides.

Hong et al. [2] introduce two algorithms to protect sensor networks against timing analysis attacks. Timing analysis is a kind of traffic analysis that exploits timing correlation of transmissions from neighboring nodes to correlate packets on successive links and therefore uncover end-to-end traffic flows between the source and the destination of those packets. Most of the traditional countermeasures against this kind of attack, such as packet reordering and decoy traffic, are not suitable for the resource-constrained sensor networks. The authors base their algorithms on adding random delays to packet retransmissions at each forwarding node, in an attempt to obfuscate the temporal relationship among packet transmissions in a neighborhood. This paper is not concerned with the actual routing of packets. In other words, it does not specify how a path for a packet is found towards the sink.

Another kind of traffic analysis attack tries to gain sensitive information about end-to-end data flows by monitoring changes in link-level traffic patterns. The heuristic algorithm proposed in [3] tries to prevent this type of attack by routing end-to-end data flows in a way that keeps the global view of link-level traffic patterns on all the links across the network as invariable as possible. In fact, when the real traffic pattern in the network changes, this algorithm reroutes traffic flows in a manner that the overall network traffic patterns remain unchanged. However, this paper does not discuss implementation issues of this algorithm and their privacy implications. This algorithm requires complete knowledge of link-level and end-to-end flows throughout the network. In a centralized implementation of this algorithm for sensor networks, the sink would be the sensible choice as the entity performing the algorithm. In that case, protocols are needed for conveying necessary information from the sensors to the sink, as well as procedures for setting up the right link-level connections by the sink, all in a secure and private manner. A distributed implementation is more challenging, especially from a security standpoint. It would mean that global knowledge of data flows and network topology must be available to all sensors.

The focus of the work reported in [4] is protection against intrusion attacks and traffic analysis attacks that aim at isolating or locating the sink. It employs two main techniques to prevent such attacks. First, it increases tolerance against sink isolation by using redundant sinks and develops secure multipath–capable path setup mechanisms, so that sensors can report their data to multiple sinks. However, this path setup mechanism is based on sink-originated beacon signals, which reveal the location and identity of the sink. It also needs all nodes to identify their neighbors, which may also be a privacy concern. The second countermeasure offered in this paper is the use of anti-timing analysis techniques that prevent an attacker from tracking packets back to the sink by monitoring the transmission times of a sensor and its parent(s). It achieves this goal by unifying the transmission rates of all nodes, using delays and dummy packets, when necessary. This technique suffers from waste of energy and bandwidth, data loss due to buffer overflows and latency due to random delays.

Deng et al. [5] use four traffic randomization techniques to protect against traffic analysis attacks that aim at locating the sink in a WSN. They try to increase randomness in traffic patterns, in order to confuse an attacker who exploits pronounced traffic patterns due to fixed-path routing protocols. Traffic analysis attacks of the “rate monitoring” and “time correlation” types have been considered in this paper. The first technique proposed in this work is a per packet random multiple-path forwarding scheme used at each node. This scheme is based on the common topology discovery method of propagating beacons from the sink, which as mentioned before, reveals the location of the sink. The second technique is a controlled random walk that uses a probabilistic forwarding scheme at each node to determine the next hop according to a uniform probability distribution, in order to protect against rate monitoring attacks. The last two techniques use fake paths and fake hotspots that are based on decoy traffic, which as mentioned before, consumes energy and bandwidth as well as degrades performance in terms of packet delivery success ratio and latency due to increased rate of collisions.

Olariu et al. [6] hide the identities and locations of the source and the destination in a wireless sensor network, as well as provide traffic untraceability by imposing an anonymous virtual infrastructure over the physical infrastructure. However, their simple routing protocol is vulnerable to packet tracing. As mentioned before, several anonymity protocols have been developed for MANET, such as ANODR[8] and MASK[9], but they are not suitable for sensor networks due to their resource limitations, especially in terms of processing power. Mist Routing [10] can also provide source and destination location privacy, but it assumes a fixed infrastructure of special routers with a pre-determined logical hierarchy overlaying the physical network. Such assumptions are not typically applicable to wireless sensor networks mainly because sensors are usually deployed in an ad hoc manner.

3 Network Model

We envision a network of many (hundreds) small wireless sensors that are deployed in a large geographical area. We also assume that a single, considerably more powerful node, called the *sink*, exists in the network that controls the sensors and collects sensory data from them. Because of their limited transmission range, sensors send their data to the sink using multihop communication. Sensors generate data independently and relay packets received from their neighbors without deterministic knowledge of their arrival times. In other words, a node can always detect packet arrivals from its neighbors[1] and is ready to retransmit them according to a certain policy. Data packets generated by sensors always travel upstream (uplink) towards the sink and are never destined for any other node. Control packets such as routing updates from sensors also travel upstream, destined for the sink, while control packets from the sink such as routing instructions, cryptographic assignments, acknowledgments and so on, travel downstream (downlink).

Our approach to ensuring anonymity and location privacy for the destination in a network is to avoid disclosing information about the destination identity and its location as much as possible. To this end, we have made the sink entirely in charge of how packet routing should be done. That is why we describe our proposed routing protocol as being destination-controlled. Through a topology discovery process, described later, the sink obtains a global view of the network topology, but no other entity knows which node is the sink. We will also explain how the sink can securely and anonymously send instructions to each node in a manner that ultimately all packets can find their ways to the sink, without the privy of any internal or external entity other than the sink.

4 Threat Model and Privacy Objectives

We intend to resist an omni-present adversary sometimes called a *global eavesdropper*. This kind of adversary can monitor all transmissions happening anywhere in the network. It can theoretically follow one single packet to its destination. It can also detect the source of a packet as soon as it is generated. Therefore, in order to have an acceptable degree of protection against this kind of adversary, we need to have a sufficient number of active sources in the network, making the process of distinguishing a specific flow among many flows difficult. The greater the number of data flows (anonymity set) the higher the degree of anonymity will be.[2] This condition is intrinsically satisfied in many types of sensor networks such as inventory tracking, habitat monitoring and environment monitoring applications. Also, given a specific packet and the entire transmission history of the network, we would like to make the task of identifying the sender of the packet as difficult as possible for an eavesdropper. Finally, and most importantly, we wish to hide the location and the identity of the sink. Our

[1] Some energy-efficient MAC protocols allow a node to detect packets even in the idle mode.

[2] As an extreme case, consider that only one data flow exists in the network. Regardless of what routing mechanism is used, a global adversary can monitor all packet transmissions. In fact, it can detect the source as soon as it generates a packet.

protocol works on the network layer but it does not preclude the use of node level mechanisms such as anti-timing analysis techniques whenever available.

Later, we will explain how packets belonging to different flows can become indistinguishable. Thus, we assume that as far as the eavesdropper is concerned, a packet transmission by a node may be for an original packet or just a relayed packet. Sensor nodes are usually vulnerable to capture and intrusion. An adversary is assumed to gain complete control of a compromised node including its routing information, collected data and all its cryptographic material. Therefore, it is important that no node is aware of the locations and identities of the sink or other data sources. In this paper we are only concerned with eavesdropping actions (passive attacks) of an adversary who is trying to locate a target node. Such an adversary usually prefers to be hidden, thus it refrains from executing active attacks e.g. denial of service, impersonation, jamming and so on, which may be discovered by the network operator using intrusion detection techniques.

5 Proposed Routing Scheme

In this section, we describe our Destination-Controlled Anonymous Routing Protocol for Sensors (DCARPS).

Part 1: Initialization

Before deploying a node *i* (including the sink), it is assigned a unique network identifier S_i. The sink and a sensor S_i are pre-programmed with a unique shared secret key K_i. During the lifetime of a sensor, its key may be updated by the sink. Due to computational and energy limitations of sensors, we only use symmetric cryptography[3]. Before deployment, the sink is also programmed to share a value with each sensor, denoted by DI_i (for Downstream Incoming), whose use in downlink communications will be explained later.

Part 2: Topology Discovery

Upon network activation and also periodically thereafter, a subset of sensors (or all of them in the worst case) broadcast route discovery messages. A sensor that originates a route discovery message includes its identity (ID) and a globally unique sequence number[4] in this message. All of the forwarding sensors append their IDs to this message. Each sensor repeats a route discovery message with a certain sequence number only once. All of these messages eventually, and usually in multiple copies, reach the sink allowing it to obtain a global view of the network topology. Please see Fig.2. A node that has already forwarded a route discovery message does not generate one of its own but will continue to relay such messages for other nodes.

This design is contrary to the common method of topology discovery in sensor networks based on broadcasting beacons from the sink. As explained before, that method violates location privacy of the sink. By making all the nodes, including the sink, behave in the same manner during topology discovery, we have effectively hid-

[3] Using light-weight cryptography, asymmetric cryptography on sensors is now possible as well.

[4] IETF RFC 4122 defines a namespace for globally unique identifiers.

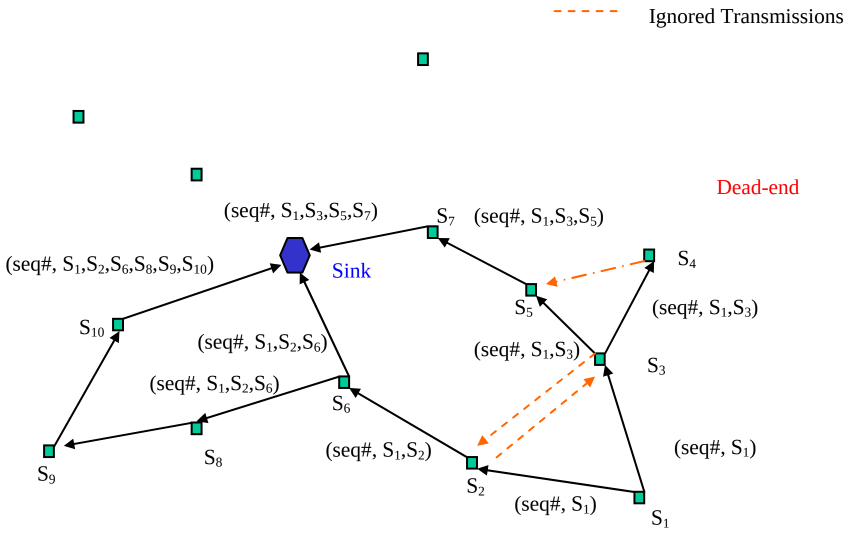

Fig. 2. Topology Discovery; S_1 broadcasts a route discovery message

den the sink. The details of our topology discovery protocol are published in another paper [13]. Currently, we are studying various techniques such as clustering and MultiPoint Relays in order to enhance the performance of this protocol.

Part 3: Route Calculation

Once the sink has acquired the network topology, it calculates routes for all the sensors. In the simplest case, it calculates only one route per sensor according to a prespecified policy, such as “the shortest path”. All the possible end-to-end routes for each sensor resemble multiple streams originating from the same point i.e. that sensor. Although each stream may branch out somewhere on the way but different branches of the eventual tree for each sensor can only merge at the final destination (the sink).

The final set of shortest paths is a tree structure rooted at the sink. Each sensor may have multiple downstream links but only one upstream link towards the sink. In other words, a *main branch* (a link connecting the sink to one of its neighbors) emanating from the sink may split several times into smaller branches, finally ending at individual leaves. A *leaf* is a sensor that does not have a downstream link on the shortest-paths tree. The shortest path for a leaf sensor contains the shortest paths for all its upstream sensors.

Part4: Uplink Path Establishment

In our routing protocol, the sink is responsible for establishing upstream next hops for all sensors. In this phase, the sink assigns labels for uplink communications to each node, including itself. A *label* is part of a packet that dictates how it should be forwarded. The label in an incoming packet at a node is called an incoming label, while the label in an outgoing packet is called an outgoing label. A node is the recipient of a packet if it has been assigned an incoming label that matches the label of the incoming packet. To forward a packet, a node swaps the label in the packet with its own outgoing label. Further into the manuscript, we will explain how the sink communicates these label assignments to sensors.

Due to the tree-like structure of the routes, each sensor will have only one incoming label (even though it may have more than one incoming link) and one outgoing label in the upstream direction, because all sensors send their data to the same destination, namely the sink, and each one of them has only one upstream link. The outgoing label of a node is always the same as the incoming label of its upstream node. The sink may have no outgoing label, while the leaf sensors have no incoming labels. The sink can assign the same outgoing label to all of its neighbors. However, this may result in a large concentration of packets with the same label in its vicinity, which may be an indication to adversaries that the sink resides in that area. Therefore, we recommend that these labels must be different. A label assignment example for a simple shortest-paths tree is shown in Fig.3.

The sink uses *path_setup* messages to inform each sensor of its assigned uplink incoming label and outgoing label. These messages travel downwards along the branches from the sink to each sensor, according to the tree structure calculated previously by the sink. The format of these messages and how we maintain their secrecy is explained below

In order to reduce overhead, we distribute labels to all the sensors connected to one main branch by using just one message. The sink assembles a cryptographic onion per each main branch and broadcasts it locally. Each layer of an onion normally contains the incoming label and the outgoing label for the sensor that can successfully decrypt that layer. Each layer is encrypted by the sink using the secret key that it shares with the intended recipient. The layers are in such an order that one of the immediate neighbors of the sink can peel off the outermost layer of an onion. Each sensor broadcasts the rest of the onion after peeling off the outer layer. At each step, only one sensor can decrypt the outer layer. However, where a branch splits up, the corresponding sensor must broadcast as many sub-onions as the number of its down-

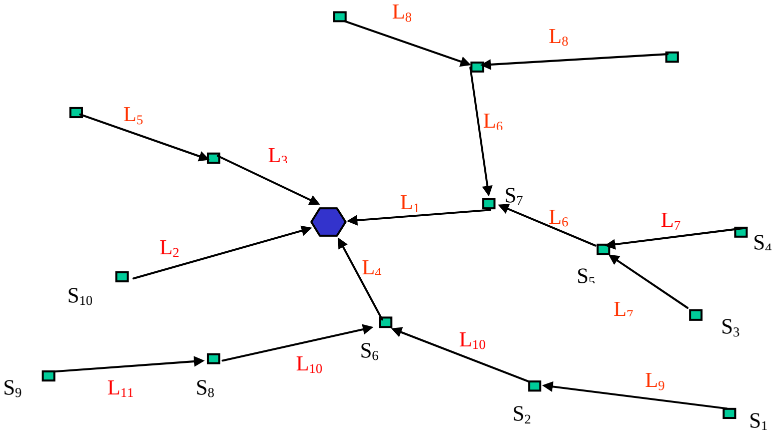

Fig. 3. Label assignments in a shortest-paths tree

stream sensors. These sub-onions are prepared by the sink in a way that the outer layer of each can only be decrypted by one of the sensors downstream from the branching sensor. As an example, the path_setup message broadcasted on the main branch labeled with L_4 in Fig.3 is:

$$K_6 (L_{10}, L_4, K_2 (L_9, L_{10}, K_1 (-, L_9)), K_8 (L_{11}, L_{10}, K_9 (-, L_{11})))$$

K_i (x) denotes encryption of x using the key K_i. At each layer, starting from the left, the first item is the incoming label and the next item is the outgoing label.

One possible algorithm for assembling the path_setup message for a main branch is the following: the sink starts at any leaf sensor and includes its outgoing label in the innermost layer of an onion. Then, it moves upstream and for each sensor adds a layer to the onion containing its incoming and outgoing labels. If it encounters a "branching sensor" (a sensor from where multiple branches start) it marks the onion so far built as a sub-onion. It then moves down on each of other branches, and for each one starts building a sub-onion beginning each at a leaf sensor. Once all the necessary sub-onions have been built, it puts them in a tandem and moves upstream from the branching sensor. This process continues until all the sensors in that main branch have been included in the onion. Please note that the sink applies this algorithm offline, to the routing information that it has collected. When all of the onions constructed in this way are broadcasted in the network, each node knows its incoming and outgoing labels.

Assignment of Unique Labels

As Fig.4 shows, when a node (B) broadcasts a packet, all nodes in its one-hop neighborhood receive that packet. Therefore, the nodes in the 2-hop neighborhood of the recipient (A) get the packet.

In order to avoid recipient ambiguity in a local broadcast, the incoming label of a node must be unique at least in its 2-hop neighborhood. In the label assignment phase, the sink must execute an algorithm to ensure that each node is allocated a unique incoming label. Below, we provide a heuristic algorithm for this task.

Denote the set of 2-hop neighbors of node X including itself with $N^2(X)$ and the yet unused labels in its 2-hop neighborhood with $L(X)$. At the start of the algorithm, the latter set is the same for all nodes and contains all the available labels. At every step

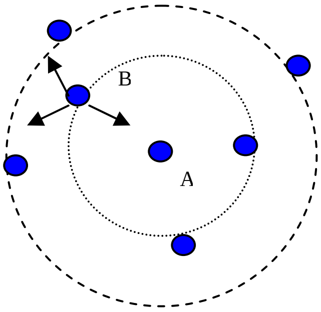

Fig. 4. A local broadcast

of the algorithm, the sink chooses a node (randomly or systematically) and assigns to it one of the labels still available in its label set. Then, it removes this label from the set of labels of that node and all its neighbors in a 2-hop area. One run of the algorithm ends when all the nodes have been allocated an incoming label. At the beginning of the next run, the set of unused labels for each node is re-initialized to contain all the available labels. In the following pseudocode, *N* is the set of all nodes and l_i is the incoming label assigned to node *i*.

```
for all $i \in N$
{
$l_i = x$;    //$x \in L(i)$
for all $j \in N^2(i)$
            $L(j) = L(j) - \{x\}$;
}
```

Part 5: Uplink Data Transmission

When a sensor has a packet, for example some sensory data to report, it encrypts it for the sink. Then, it appends its outgoing label to the packet. The upstream neighbor, with an incoming label matching the packet label, accepts the packet and switches its label to its own outgoing label. At this point, it can rebroadcast the packet. However, to prevent a global eavesdropper that applies content analysis to the payload in order to trace the packet back to the source and/or destination, we make the payload to appear different at each hop. To do this, the forwarding sensor encrypts the already encrypted payload for the sink. Every sensor in the path repeats the encryption process, effectively creating an onion. Therefore, on each hop, the packet contains an unencrypted label and a payload that has been encrypted several times. Upon receiving the packet, the sink performs recursive decryptions to recover the original data.

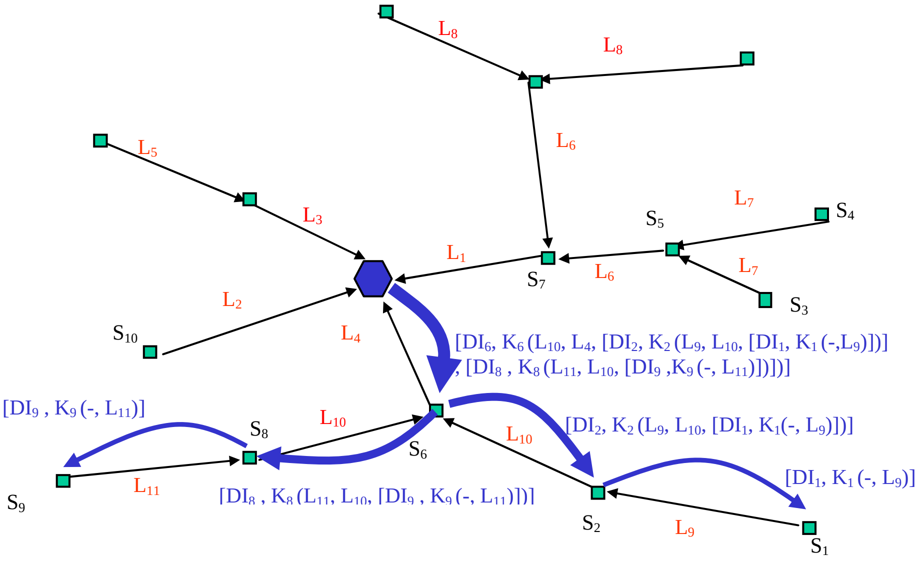
L_8
L_8
L_6
L_5
L_3
L_7
S_5
S_4
L_1
S_7
L_6
L_7
S_3
L_2
S_{10}
L_4
[DI₆, K₆ (L₁₀, L₄, [DI₂, K₂ (L₉, L₁₀, [DI₁, K₁ (-,L₉)])]
, [DI₈ , K₈ (L₁₁, L₁₀, [DI₉ ,K₉ (-, L₁₁)])])]
[DI₉ , K₉ (-, L₁₁)]
S_8
L_{10}
S_6
L_{10}
[DI₂, K₂ (L₉, L₁₀, [DI₁, K₁(-, L₉)])]
[DI₁, K₁ (-, L₉)]
S_9
L_{11}
[DI₈ , K₈ (L₁₁, L₁₀, [DI₉ , K₉ (-, L₁₁)])]
S_2
L_9
S_1

Fig. 5. Downlink data transmission using downlink labels

Part 6: Downlink Data Transmission

Downlink communications (from the sink to the sensors) is also based on layered cryptography and label switching. We explained before how the sink sends path_setup messages to all the sensors that belong to the same main branch in the shortest-paths tree. Other messages, such as acknowledgments and other control commands are sent in a similar manner. Remember that each node is initialized with a DI (Downstream Incoming) label. When the sink wishes to transmit a packet, it labels the outermost layer of an onion with the DI of one of its neighbors that is supposed to be the next hop for that packet according to the latest routing information available to the sink. However, inside that layer, which is encrypted for that neighbor, it precedes each sub-onion with the DI label of the next hop on the route.

As an example, the path setup process for the example in Fig.3 is illustrated in Fig. 5. For enhanced security, DI labels may be changed periodically.

Sink Anonymity in Downlink Communications

In order to prevent an eavesdropper from acquiring valuable information regarding the location of the sink, path_setup messages and other control packets sent by the sink in the downlink must be indistinguishable from data packets transmitted by sensors. Data packets consist of a label and an encrypted payload that is re-encrypted on every hop. Control packets look exactly like data packets since they have a label (DI) and an encrypted part that looks different on each hop because of layered cryptography. Moreover, these packets are only transmitted within a short setup phase, which gives the adversary little chance to find the sink while in other protocols any compromised node gives a clue about the sink's location.

Upon hearing a packet, a sensor performs the following algorithm:

```
If (label == my DI)
   then
       downlink packet, I am the immediate recipient.
       decrypt payload with key shared with sink.
       re-broadcast sub-onion(s).
else if (label exists in my routing table)
   then
       data packet, I am the immediate recipient.
       swap label, (re)encrypt payload and forward packet.
else
  discard packet.
```

6…Summary

We have used the concepts of onion routing and label switching to develop an anonymous untraceable routing protocol for wireless sensor networks. Our protocol ensures location privacy for the sink by putting the control of routing in its hands. Also, unlike other routing protocols for sensor networks, in order to hide the location of the sink, our topology discovery process is initiated by the sensors not by the sink. To thwart content analysis attacks aimed at tracing packets back to the source or destination, we have used layered cryptography to make packets appear randomly different on different hops along their paths. However, in order to conserve the energy of sensors, we have used labels to identify the next hop for each packet so that only the immediate recipient of a packet attempts to decrypt it. Moreover, a sensor performs a minimal amount of computation and has only one key, which is shared with the sink.

References

1. Pandurang Kamat, Yanyong Zhang, Wade Trappe, Celal Ozturk, Enhancing Source-Location Privacy in Sensor Network Routing, in the proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICSCS'05), Vol. 00, pp. 599 – 608.
2. Xiaoyan Hong, Pu Wang, Jiejun Kong, Qunwei Zheng, Jun Liu, Effective Probabilistic Approach Protecting Sensor Traffic, IEEE Military Communication Conference (Milcom) 2005, Atlantic City, NJ, Oct 2005.
3. Shu Jiang, Nitin H.Vaidya, Wei Zhao, Routing in Packet Radio Networks to Prevent Traffic Analysis, in the proceedings of the IEEE Information Assurance and Security Workshop, West Point, NY, July 2000.
4. Jing Deng, Richard Han, Shivakant Mishra, Intrusion Tolerance and Anti-Traffic Analysis Strategies For Wireless Sensor Networks, in the proceedings of the IEEE International Conference on Dependable Systems and Networks (DSN 2004), 28 June - 1 July 2004, Florence, Italy.
5. Jing Deng, Richard Han, Shivakant Mishra, Countermeasures Against Traffic Analysis Attacks in Wireless Sensor Networks, First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05), Athens, Greece, pp. 113-126.
6. Stephan Olariu, Mohamed Eltoweissy, Mohamed Younis, ANSWER: Autonomous Wireless Sensor Network, Q2SWinet'05, October 13, 2005, Montreal, Quebec, Canada.
7. Jamal N. Al-Karaki, Ahmed E. Kamal, Routing Techniques In Wireless Sensor Networks: A Survey, IEEE Wireless Communications, Vol. 11, No. 6. (2004), pp. 6-28.
8. J. Kong, Anonymous and Untraceable Communications in Mobile Wireless Networks, PhD thesis, University of California, Los Angeles, June 2004.
9. Y. Zhang, W. Liu, and W. Lou, Anonymous Communications in Mobile Ad Hoc Networks, IEEE INFOCOM, Miami, FL, March 2005.
10. J. Al-Muhtadi, R. Campbell, A. Kapadia, M. Dennis Mickunas, Seung Yi, Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments, in the proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02), July 2002, Vienna, Austria, p74.
11. A. Pfitzmann and M. Kohntopp. Anonymity, Unobservability and Pseudonymity -- A Proposal for Terminology, In Hannes Federath (Ed.), Designing Privacy Enhancing Technologies, Lecture Notes in Computer Science, LNCS 2009, pp. 1-9, Springer-Verlag, 2001.
12. D. L. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, Communications of the ACM, 24(2), pp. 84–88, 1981.
13. Alireza A. Nezhad, Dimitris Makrakis, Ali Miri, Anonymous Topology Discovery for Wireless Sensor Networks, submitted to the 3-rd ACM International Workshop on QoS and Security for Wireless and Mobile Networks, October 22 - 26, 2007, Chania, Crete Island, Greece.