

# INSENS: Intrusion-Tolerant Routing in Wireless Sensor Networks

Jing Deng, Richard Han, Shivakant Mishra

*Department of Computer Science*

*University of Colorado, Boulder, CO 80309-0430.*

*Email: {Jing.Deng, rhan, mishras}@cs.colorado.edu*

## Abstract

*This paper describes an INtrusion-tolerant routing protocol for wireless SENsor Networks (INSENS). INSENS constructs forwarding tables at each node to facilitate communication between sensor nodes and a base station. It minimizes computation, communication, storage, and bandwidth requirements at the sensor nodes at the expense of increased computation, communication, storage, and bandwidth requirements at the base station. INSENS does not rely on detecting intrusions, but rather tolerates intrusions by bypassing the malicious nodes. An important property of INSENS is that while a malicious node may be able to compromise a small number of nodes in its vicinity, it cannot cause widespread damage in the network.*

Wireless sensor networks (WSNs) are rapidly emerging as an important new area in the research community. Applications of WSNs are numerous and growing, and range from indoor deployment scenarios in the home and office to outdoor deployment scenarios in natural, military and embedded settings. In each of these scenarios, lives and livelihoods may depend on the timeliness and correctness of the sensor data obtained from dispersed sensor nodes. As a result, such WSNs must be secured to prevent an intruder from obstructing the delivery of correct sensor data and from forging sensor data.

The design and implementation of a secure routing in WSNs must simultaneously address several difficult research challenges. First, wireless communication among the sensor nodes increases the vulnerability of the network to eavesdropping, unauthorized access, spoofing, replay and denial-of-service (DOS) attacks. Second, the sensor nodes themselves are highly resource-constrained in terms of limited memory, CPU, communication bandwidth, and especially battery life. These resource constraints limit the degree of encryption, decryption, and authentication that can be implemented on individual sensor nodes, and call into question the suitability of traditional security mechanisms such as compute-intensive public-key cryptography. Third, WSNs face the added physical security risk of being deployed in the field, so that individual sensor nodes can be obtained and subject to

attacks from a potentially well-equipped intruder in order to compromise a single resource-poor node. Given these challenges, our approach for securing WSNs concedes that a well-equipped intruder can compromise individual sensor nodes, but that the overall design of our secure routing system should tolerate these intrusions such that the network as a whole remains functioning.

The INSENS secure routing system adheres to the following design principles. First, to prevent DOS-style flooding attacks, individual nodes are not allowed to broadcast to the entire network. Only the base station is allowed to broadcast. Authentication of the base station is achieved via one-way hashes, so that individual nodes cannot spoof the base station and thereby flood the network. Unicast packets must first traverse through the base station, thereby preventing DOS/DDOS broadcast attacks. Second, to prevent advertisement of false routing data, control routing information must be authenticated. A key consequence of this approach is that the base station always receives knowledge of the topology that is correct, though it may only represent a partial picture due to malicious packet dropping. Third, to address resource constraints, 1) symmetric key cryptography is chosen for confidentiality and authentication between the base station and each resource-constrained sensor nodes, since it is considerably less compute-intensive than public key cryptography, and 2) the resource-rich base station is chosen as the central point for computation and dissemination of the routing tables. Fourth, to address the notion of compromised nodes, redundant multipath routing is built into INSENS to achieve secure routing. The goal is to have disjoint paths so that even if an intruder takes down a single node or path, secondary paths will exist to forward the packet.

INSENS comprises of route discovery and data forwarding phases. Route discovery ascertains the topology of the sensor network and data forwarding deals with forwarding data from sensor nodes to the base station, and from base station to the sensor nodes. Route discovery is performed in three rounds. In the first round, the base station floods (limited flooding) a *request message* to all the reachable sensor nodes in the network. The base station broadcasts a

request message that is received by all its neighbors. A sensor node  $x$  that receives a request message for the first time first records the identity of the sender in its neighbor set, and then broadcasts a request message. This message includes a path from the base station to  $x$ . When a node receives another request message, the identity of the sender is added to its neighbor set, but the request is not rebroadcast. We use two mechanisms to counter malicious security attacks in this round. First, we leverage the concept of one-way sequences to identify a request message initiated by the base station and to restrict DOS-style flooding attacks. This allows a sensor node to verify that a request message it received indeed originated from the base station, and prevents a malicious node from flooding the network with out of date messages. Second, we use a keyed MAC algorithm. Each sensor node is configured with a separate secret key that is shared only with the base station. Before forwarding a request message, a node  $x$  generates a MAC by applying a keyed MAC algorithm. This MAC is applied to the complete path consisting of  $x$ 's identity appended to the path from the incoming request message.

The overall effect of these security mechanisms is that a malicious node can attack in the first round only by localized flooding, by not forwarding a request message, and by sending a fake path in the request which is later on detected in the second round. The latter two attacks will result in some nodes downstream from the malicious node not getting a request message, or not being able to forward their feedback message in the second round. Thus, a malicious node can compromise only the downstream nodes and a small number of nodes in its vicinity.

In the second round, sensor nodes send their (local) topology information using a *feedback message* to the base station. After a node has forwarded its request message, a node waits a certain timeout interval before generating a feedback message. The integrity of the topology data returned to the base station by each node in its feedback message is protected using a keyed MAC. This keyed MAC ensures that the base station will construct a correct topology, though it may be incomplete due to malicious nodes that may drop or tamper with feedback messages. The messages that reach the base station are guaranteed after verification to be correct and secure from tampering.

The overall effect of these security mechanisms is that a malicious node is limited in the damage it can inflict, whether attacking by DOS attack, by not forwarding a feedback messages, or by modifying the neighborhood information of nodes, which can be detected at the base station. These attacks will result in some of the nodes downstream from the malicious node not being able to

provide their correct connectivity information to the base station. Though a malicious node could launch a battery-drain attack by persistently sending spurious feedback messages at the rate-controlled limit, such an attack would still affect only a limited number of upstream nodes.

In the third round, the base station computes the forwarding tables for each sensor node based on the information received in the second round and sends them to the respective nodes using a *routing update* message. After sending the request message, the base station waits for a certain period of time to collect all the connectivity information received via feedback messages. From this connectivity information, the base station computes two independent paths for each node. These paths are chosen to be far apart to minimize the damage an intruder may cause. The base station then computes the forwarding tables of each node. These forwarding tables are propagated to the respective nodes in a breadth-first manner.

A node maintains a forwarding table that has several entries, one for each route to which the node belongs. Each entry is a 3-tuple: destination, source, and immediate sender. Destination is the node id of the destination node to which a data packet is sent, source is the node id of the node that created this data packet, and immediate sender is the node id of the node that just forwarded this packet. For example, given a route from node  $S$  to  $D$ :  $S \rightarrow a \rightarrow b \rightarrow c \rightarrow D$ , the forwarding table of node  $a$  will contain an entry  $\langle D, S, S \rangle$ , forwarding table of  $b$  will contain an entry  $\langle D, S, a \rangle$ , and the forwarding table of  $c$  will contain an entry  $\langle D, S, b \rangle$ . On receiving a data packet, a node searches for a matching entry (destination, source, immediate sender) in its forwarding table. If it finds a match, it forwards (broadcasts) the data packet.

We have simulated INSENS on ns2click, a network simulation tool that combines the ns-2 network simulator with the Click Modular Router. Performance measured from this simulation shows that INSENS tolerates DOS and other malicious attacks launched by intruder nodes, and functions correctly over a variety of sparse, dense, random and grid topologies despite intrusions.

We have also implemented INSENS on a physical testbed of wireless sensor network comprising of UC Berkeley MICA sensor motes. We have experimented with using RC5 and AES encryption standards on motes, an RC5-based scheme to generate message authentication codes (MACs) on motes, and an RC5-based generation of one-way sequence numbers on motes. This implementation demonstrates that the resource (memory, CPU, and bandwidth) requirements of INSENS can be easily met by currently available sensor nodes such as motes.