WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2006; **6**:375–391
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/wcm.402

Secure cell relay routing protocol for sensor networks

Xiaojiang Du[1,*,†], Yang Xiao[2], Hsiao-Hwa Chen[3] and Qishi Wu[4]

[1]*Department of Computer Science, North Dakota State University, Fargo, ND 58105, U.S.A.*
[2]*Department of Computer Science, The University of Memphis, Memphis, TN 38152, U.S.A.*
[3]*Institute of Communications Engineering, National Sun Yat-Sen University, Taiwan*
[4]*Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, U.S.A.*

Summary

Past researches on sensor network routing have been focused on efficiency and effectiveness of data dissemination. Few of them consider security issues during the design time of routing protocols. Security is very important for many sensor network applications. Studies and experiences have shown that considering security during design time is the best way to provide security for sensor network routing. In this paper, we propose an efficient key management scheme and a novel secure routing protocol—Secure cell relay (SCR) for sensor networks. We also present an effective key setup scheme for sensor nodes deployed in the later stage. We analyze the security of SCR under various attacks and show that SCR is very effective in defending against several sophisticated attacks, including selective forwarding, sinkhole, wormhole, Sybil, hello flooding, and clone attacks. SCR is an energy-efficient routing protocol with acceptable security overhead. Our simulations demonstrate that with all the security primitives, SCR still has lower energy consumption and higher delivery ratio than a popular routing protocol—directed diffusion. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: secure routing; sensor networks

1. Introduction

Sensor networks have many application areas such as military, homeland security, environment, agriculture, manufacturing, and so on. Many routing protocols have been proposed for sensor networks, such as directed diffusion [1], two tier data dissemination (TTDD) [2], Mesh [7], LEACH [24] etc. However, most of these routing protocols did not consider security issues during the protocol design time. One can envision the deployment of large-scale sensor networks in the future where hundreds to thousands of small sensors form self-organizing wireless networks, and many sensor networks need security. However, providing security in sensor networks is not easy. Compared with conventional desktop computers, severe challenges exist since sensor nodes have limited processing capability, storage, bandwidth, and energy. Despite the challenges, security is important and even critical for many sensor network applications, such as military and homeland security.

*Correspondence to: Xiaojiang Du, Department of Computer Science, North Dakota State University, 258 IACC, A2 Fargo, ND 58105, U.S.A.
†E-mail: Xiaojiang.Du@ndsu.edu

Contract/grant sponsor: North Dakota NASA EPSCoR; contract/grant number: NCC5-582.

Several recent works [8–21] have addressed security problems in sensor networks. However, most works consider routing protocols and security schemes (like key management, authentication) in sensor networks separately. Few works consider security issues during the design time of a routing protocol. Since most existing routing protocols have not been designed with security as a goal, it is unsurprising that they are vulnerable to many attacks. However, this is non-trivial to fix the problem since it is unlikely that a sensor network routing protocol can be made secure by incorporating security mechanisms after the design has completed [3]. We believe that sensor network routing protocols must be designed with security in mind, and this is the only effective solution for secure routing in sensor networks.

In this paper, we present a novel secure routing protocol for sensor networks—secure cell relay (SCR) routing protocol. The main differences between SCR and other sensor network security protocols [8–21] are listed as follows.

(1) We use a three-way handshake protocol to establish neighborhood relationship between sensors, which can defend against Hello flood attack.
(2) We consider security issue during the protocol design of SCR. Several security features are incorporated in SCR to defend various attacks. Details are given in Section 3.
(3) We design an efficient key setup scheme for sensor nodes deployed after initialization phase. The detailed discussion of related works is given in Section 6.

Furthermore, SCR is an energy-efficient routing protocol. In SCR, the entire routing area is divided into several small, equal-size squares—cells. All these cells form a grid in the network, as illustrated in Figure 1. The grid structure is fixed even though the target or base station may move around.

In TTDD [2], Ye *et al*. also use grid structure to aid routing. However, the grid structure in TTDD is not fixed, and it depends on the locations of the source nodes. The idea of using grid also appears in GRID [6], a routing protocol designed for mobile ad hoc networks. There are several differences between SCR and GRID: (1) route discovery is needed in GRID to find a route between a source-destination pair. In SCR, once source knows the location of base station, no route discovery is needed. (2) In GRID, a gateway node is elected in each cell and used to forward packets. A gateway election algorithm is needed to elect and maintain gateway nodes, and it introduces significant overhead. SCR does not use gateway nodes.

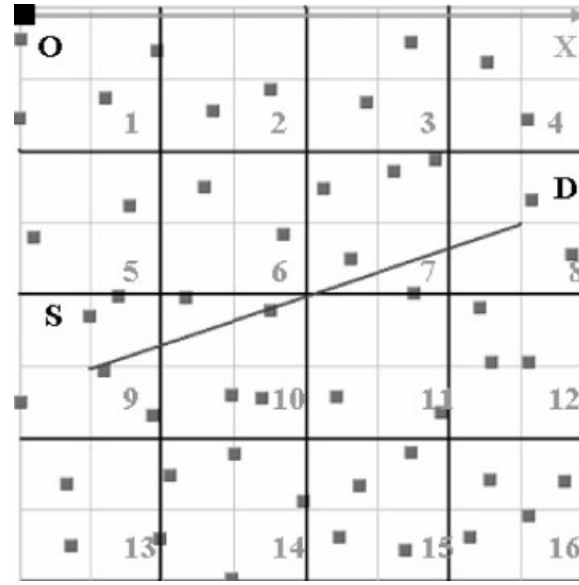


Fig. 1. The grid structure.

The SCR routing protocol is essentially a clustering-based algorithm. Several clustering-based routing protocols have been proposed for sensor networks, like LEACH [24], TTDD [2] etc. There are major differences between SCR and other clustering-based protocols. In SCR, nodes form cluster (cell) based on their locations. We utilize the fact that nodes in most sensor networks are location aware. While other clustering-based protocols use different schemes to form and manage clusters, and these schemes introduce non-trivial overhead. Another difference is that SCR does not need to elect cluster heads. In SCR, an active node becomes a relay node based on its remaining energy and a back-off algorithm. While other clustering-based routing protocols need algorithms to elect and maintain cluster heads. Furthermore, an important difference between SCR and the above routing protocols is that SCR provides secure routing and can defend against various attacks.

The rest of the paper is organized as follows. In Section 2, we describe system assumptions and attacks on routing. SCR is proposed in Section 3. We analyze the security features of SCR in Section 4. Routing performance is evaluated in Section 5. Related work is summarized in Section 6. Finally, we conclude the paper in Section 7.

2. System Assumptions and Attacks on Routing

In this section, we describe the system assumptions for the design of the SCR routing protocol. We consider a sensor network composed of a large number of small sensors. Due to cost constraints these sensors are not equipped with tamper-resistant hardware. We assume that if an adversary compromises a node, she can extract all key material, data, and code stored on that node.

We design SCR routing protocol based on the following assumptions: (1) each sensor is static and aware of its own location. Sensor nodes can use location services such as References [22,23] to estimate their locations, and no GPS receiver is required at each node; (2) base stations are trusted. Since a base station is the gateway for the nodes to communicate with the outside world, compromising the base station could render the entire sensor network useless. Thus we assume that the base stations are trusted.

The network is installed with a grid. An example of the grid structure with 16 cells is shown in Figure 1. Assume that the transmission range of a sensor node is R, and the side length of each cell can be set as $a = R/(2\sqrt{2})$. This value of a ensures that each sensor can directly communicate with sensors in neighbor cells, including the diagonal neighbor cell. Each cell has a unique id (e.g., the number in Figure 1). Given the position of a reference point (e.g., point O in Figure 1) in the grid and a direction (e.g., the X-axis in Figure 1), each node can determine the cell in which it locates, based on its own location and the cell size. The reference point and the direction are pre-stored in each sensor node.

Many sensor network routing protocols are quite simple, and for this reason are susceptible to several kinds of attacks. Attacks on sensor networks have been discussed in several papers [3,12–14]. Most network layer attacks against sensor networks fall into one of the following categories: manipulating routing information; selective forwarding; Sybil [16]; sinkhole; wormhole [15]; and Hello flooding (unidirectional) attacks [3]. We briefly describe those attacks on sensor networks as follows.

2.1. Manipulating Routing Information

The most direct attack against a routing protocol is to target the routing information exchanged among nodes. By spoofing, altering, or replaying routing information, adversaries may be able to create routing loops, attract or repel network traffic, extend or shorten source routes, generate false error messages, partition the network, increase end-to-end latency etc.

2.2. Selective Forwarding Attack

Many sensor network routing protocols are based on the assumption that participating nodes will faithfully forward received packets. In a selective forwarding attack, compromised or malicious nodes may selectively forward some of packets while dropping others. An adversary interested in suppressing or modifying packets originated from some selected nodes can reliably forward the remaining traffic and limit suspicion of her misbehaviors. Selective forwarding attacks are typically most effective when the attacker is explicitly included on the path of a data flow.

2.3. The Sybil Attack

In a Sybil attack [16], a single node presents multiple identities to other nodes in the network. The Sybil attack can significantly reduce the effectiveness of fault-tolerant schemes such as distributed storage, multi-path routing, and topology maintenance. Storage partitions or multi-path routes believed to be using disjoint nodes could actually be using a single adversary presenting multiple identities. Sybil attacks also pose a significant threat to geographic routing protocols. Location-aware routing often requires nodes to exchange coordinate information with their neighbors to efficiently route geographically addressed packets. It is only reasonable to expect a node to accept a single set of coordinates from each of its neighbors, but by using the Sybil attack, an adversary can 'be in more than one place at once.'

2.4. Sinkhole Attack

In a sinkhole attack, the adversary's goal is to lure nearly all the traffic from a particular area through a compromised node, creating a metaphorical sinkhole with the adversary at the center. Sinkhole attacks can enable many other attacks such as selective forwarding. Sinkhole attacks typically work by making a compromised node look especially attractive to surrounding nodes with respect to the routing algorithm. For instance, an adversary could spoof or replay an advertisement for an extremely high-quality route to a base station. Some protocols might actually try to verify the quality of route with end-to-end acknowledgements

containing reliability or latency information. In this case, a laptop-class adversary with a powerful transmitter can actually *provide* a high-quality route by transmitting with enough power to reach the base station in a single hop. Due to either the real or imagined high-quality route through the compromised node, it is likely each neighboring node of the adversary will forward packets destined for a base station through the adversary, and also propagate the attractiveness of the route to its neighbors.

2.5. Wormhole Attack

In the wormhole attack [15], an adversary tunnels messages received in one part of the network over a low-latency link and replays them in a different part. Wormhole attacks usually involve two distant malicious nodes colluding to understate their distance from each other by relaying packets along an out-of-bound channel available only to the attacker. An adversary situated close to a base station may be able to completely disrupt routing by creating a well-placed wormhole. An adversary could convince nodes who would normally be multiple hops from a base station that they are only one or two hops away via the wormhole. Wormhole attacks would likely be used in combination with selective forwarding or eavesdropping. Detection is potentially difficult when used in conjunction with the Sybil attack.

2.6. Hello Flood (Unidirectional) Attack

In Reference [3], the authors introduced a novel attack against sensor networks: the Hello flood attack. It is an attack by utilizing unidirectional connection between nodes. We also refer to this attack as unidirectional attack. Many protocols require nodes to broadcast Hello packets to announce themselves to their neighbors, and a node receiving such a packet may assume that it is within (normal) radio range of the sender. This assumption may be false because of the well-known unidirectional problem in ad hoc networks. For example, a powerful attacker (e.g., a laptop) broadcasting routing or other information with large enough transmission power could convince many sensor nodes in the network that the adversary is their neighbor. But those sensor nodes sufficiently far away from the adversary cannot send packets to the attacker directly, and they would send packets into oblivion. The network is left in a state of confusion. A node realizing the link to the adversary is false could be left with few options: all its neighbors might be attempting to forward packets to the adversary as well.

2.7. Clone Attack

In a clone attack, an attacker loads its own nodes with the keys of a compromised node, and then deploys these cloned nodes in different locations of the sensor network. These cloned nodes then try to establish pairwise keys with their neighbors. Once they are accepted by their neighbors, they can launch various insider attacks such as injecting false data packets. Consequently, an attacker might only need to compromise a few sensor nodes to bring down the entire network due to the unattended nature of a sensor network.

3. The Secure Cell Relay Routing Protocol

In this section, we present the SCR routing protocol. In this paper, we consider sensor networks with fixed base stations. Furthermore, for the simplicity of discussion, in the following we assume that there is only one base station in the sensor network. However, our SCR routing protocol can be applied to sensor networks with multiple base stations.

3.1. Key Management

Key management is an essential cryptographic primitive upon which other security primitives are built. Most security requirements, such as privacy, authenticity, and integrity, can be addressed by building upon a solid key management framework. In fact, a secure key management scheme is the prerequisite for the security of these primitives, and thus essential to achieving secure infrastructure in sensor networks. We design an efficient key management scheme for sensor networks as follows.

Let T_{init} denote the time of a newly deployed sensor to discover its one-hop neighbors. We do not assume that sensor nodes are equipped with tamper-resistant hardware due to cost constraints. Assume that if an adversary compromises a node, she can extract all key material, data, and code stored on that node. However, we assume that there exists a lower bound on the time (denoted as T_{min}) for an adversary to compromise a node and the time T_{init} is less than T_{min}. In practice, the T_{init} is in the order of several seconds, while locating and compromising a sensor usually take longer time, thus it is reasonable to assume

$T_{\min} > T_{\text{init}}$. We describe the details of our key management scheme in the following subsections.

3.1.1. Establishing individual keys

Every node has an individual key that is only shared with the base station (BS). This key is generated and pre-loaded into each node prior to its deployment. Each node has a unique ID, and the individual key K_u for a node u is generated as follows: $K_u = f_K(u)$, where f is a pseudo-random function and K is a master key known only to the BS. The BS only needs to store the master key K to save the storage, and the BS can compute the individual key of a sensor node based on the sensor's ID.

3.1.2. Establishing pairwise keys

A globally shared key K_G and the location of the BS are stored in each sensor node before sensor deployment. The BS also knows K_G, which is used only for a short period T_{init}—the initialization phase during which each sensor discover its neighbors and establish the shared pairwise keys. All sensor nodes are synchronized with the BS before deployment. The time synchronization is only needed for the short period T_{init}. Because of the limited processing and storage capabilities, symmetric encryption is used. To provide authentication and integrity of messages, we compute a message authentication code (MAC) for each encrypted message in SCR. The following notations are used to describe the security operations.

- A, B are principals such as communicating nodes.
- $\{m\}K$ denotes that message m is encrypted with key K.
- MAC(K, m) is the message authentication code of message m using key K.

In the following, we will use MAC(K, *) to denote the message authentication code of the corresponding message. After deployment, each sensor node sets a timer $T_{\min}$, and discovers its one-hop neighbors via a three-way handshake protocol. The three-way handshake protocol establishes shared pairwise keys between two neighbor nodes, and the detail is given below.

(1) First, each sensor node broadcasts a Discovery message to its neighbors. The Discovery message includes the following fields: {node_ID, time_stamp} $K_G + \text{MAC}(K_G, ^*)$. The Discovery message is encrypted with K_G. Suppose that node B broadcasts a Discovery message.
(2) When a neighbor sensor node A receives the Discovery message from B, it decrypts the message with K_G, and obtains the time_stamp. If the message is too old, it will be discarded. This avoids the Message Replay attack. If the time_stamp falls within a time window and the MAC passes the verification, node A considers this is a valid Discovery message, and sends a Challenge message to node B with the following fields: {node_ID, time_stamp, nonce N_0} $K_G + \text{MAC}(K_G, ^*)$, where nonce N_0 is a one-time random number generated by node A. Note that node_ID here is A's ID.
(3) When node B receives the Challenge message, it replies an Ack (acknowledge) message to node A with the following fields: {node_ID, location of B, time_stamp, K_{AB}, K_B^b, $N_0 + 1$}$K_G + \text{MAC}(K_G, ^*)$, where K_{AB} and K_B^b are keys generated by node B. Pairwise key K_{AB} will be use for the communication between A and B, and the broadcast key K_B^b will be used for the future broadcast from node B. All neighbors of node B record K_B^b, but only A records K_{AB}. It is assumed that no node has been compromised during the initialization phase, that is, only legitimate nodes know K_G at this time. Although other neighbors of node B can overhear the Ack message, they will not record K_{AB} since they are legitimate nodes and run the legal routine. If A obtains the correct $N_0 + 1$, then the neighbor relationship is successfully established. Node A records B as one of its neighbors (including B's ID, location information, and two keys: K_{AB} and K_B^b).

The three-way handshake is needed to avoid the unidirectional link problem/attack, for example, if B is a more powerful node such as a laptop with a larger transmission range than node A, then B can send the Hello message to A directly, but A cannot send a packet directly to B. The above Challenge-Ack mechanism avoids the unidirectional link problem/attack. If A cannot reach B directly, then B will not know the nonce N_0, and there is no way for B to send a valid Ack message back to A. When the timer $T_{\min}$ expires, each sensor node destroys the globally shared key K_G. Note that the same key K_{AB} will be used for the communications between A and B.

The broadcast key (e.g., K_B^b) provides support for in-network processing or data aggregation in sensor networks. Data generated by neighbor sensor nodes

can be sent to one node and the data can be combined together to remove the redundancy. Furthermore, sensor nodes that are not in a local area can set up shared group keys, based on pairwise keys between neighbor nodes, and in-network processing can be conducted in a large area. We will not discuss the details in this paper.

3.2. Establishing Keys for Newly Deployed Sensor Nodes

After initial deployment, some area of a sensor network may not be covered by any sensor node due to the randomness of sensor location, for example, sensors are distributed from an airplane. Furthermore, sensor nodes are unreliable devices with limited power supply, and they may fail or run out of power over time. These factors can cause severe coverage and connectivity problems in sensor networks, and significantly degrade network performance and shorten network lifetime. To solve these problems, one solution is to deploy new sensor nodes to the field after some operation time. However, the additional deployment of sensor nodes poses challenge on security schemes. Specifically, the newly deployed nodes need to establish security keys with existing nodes. This is a non-trivial problem because of the following reasons:

(1) An existing node could have been compromised;
(2) A newly deployed node could be an adversary or a legal node;
(3) There is no globally shared key (as during the initial sensor deployment) between new nodes and existing nodes.

Most existing sensor network key management protocols [8–11] did not consider this issue. In the following, we present an efficient scheme to verify newly deployed sensor nodes and establish pairwise keys between new nodes and existing nodes.

Since a newly deployed sensor node could be an adversary, we need to verify its identity. The idea is to utilize the BS to verify the new node. We use the example in Figure 2 to describe the scheme. For each new sensor node R, the BS calculates an individual key K_R based on the master key K and the node ID R. The individual key K_R is loaded in node R before its deployment. After deployed in the sensor network, node R will prove it is a legal node via the BS, and then the BS sets up pairwise keys for R and its neighbors. The detail is presented below.

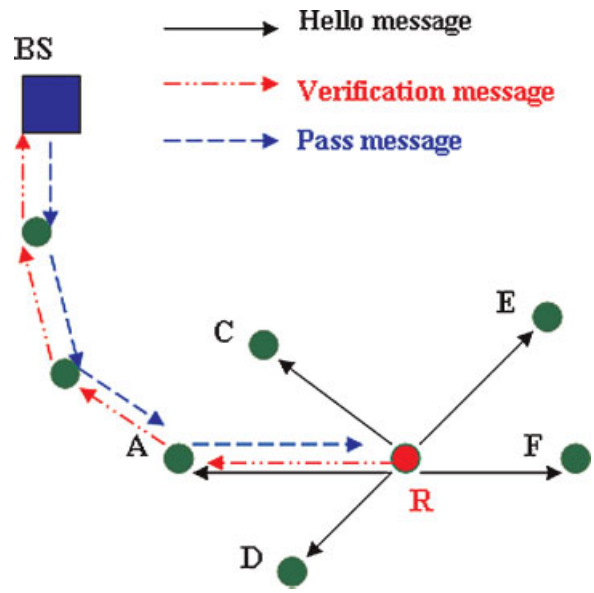


Fig. 2. Establishing keys for newly deployed sensor nodes.

(1) After deployment, node R broadcasts a Hello message (not encrypted) to all its neighbors, and R sets a timer T.
(2) When each one-hop neighbor of node R receives the Hello message, it will take a small random delay, and then sends its node ID (not encrypted) to node R. The small random delay is used to avoid the collision from simultaneous transmissions.
(3) When timer T expires, node R constructs a Verification message as following: ID list + MAC(K_R, ID list). The ID list includes the node IDs of R's neighbors. The timer T is set large enough such that all active neighbors can send their IDs to a new node. Then R randomly selects a neighbor (e.g., node A), and sends the Verification message via the neighbor to the BS. This reduces the probability that a compromised neighbor drops the Verification message.
(4) When the BS receives the Verification message, since the BS can compute K_R based on R and the master key K, the BS can verify if MAC(K_R, ID list) is correct. If yes, then R is proved as a legal node. Based on the ID list, the BS will generate pairwise keys between node R and each of its neighbors (e.g., $K_{RA}, K_{RC}, K_{RD}, \ldots$). The pairwise key is generated by the BS in the following way: for neighbor node A, the pairwise key $K_{RA} = f_{K_B}(A)$, where f is a pseudo-random function. Then the BS sends the pairwise key to each

neighbor of node R. For example, for neighbor node A,

$$\mathrm{BS} \rightarrow \mathrm{A} : \{K_{\mathrm{RA}}\}K_{\mathrm{A}} + \mathrm{MAC}(K_{\mathrm{A}}, ^{*})$$

where K_{A} is node A's individual key shared between A and the BS.

If the Verification message does not pass the check, the BS will notify neighbor nodes of R that R is an adversary.

(5) If R passes the verification, the BS will also send to node R a Pass message, which include the pairwise keys generated by the BS, that is, $\{K_{\mathrm{RA}}, K_{\mathrm{RC}}, K_{\mathrm{RD}}, \ldots\}K_{\mathrm{R}}$, encrypted by R's individual key. After receiving the Pass message, node R and its neighbors can communicate securely by using the pairwise keys.

(6) If R does not receive a Pass message within a timeout, R will randomly select another neighbor (e.g., node C), and send the Verification message via this neighbor to the BS. This scheme can defend against the attack where a compromised node (in the route from node A to the BS) drops the previous Verification message.

(7) After setting up pairwise keys with each one-hop neighbor, the new node R can construct a broadcast key $K_{\mathrm{R}}^{\mathrm{b}}$, and send $K_{\mathrm{R}}^{\mathrm{b}}$ to each one-hop neighbor by using the corresponding pairwise key, for example, R A : $\{K_{\mathrm{R}}^{\mathrm{b}}\}K_{\mathrm{RA}}$, and so on.

(8) In case the new node R is compromised and the individual key K_{R} is revealed, it will only affect the local neighbor area of node R, that is, all the pairwise keys with node R is compromised, but it will not affect nodes in other areas, since each new node has a different individual key. Thus, the impact of compromising a new node is localized and limited.

3.3. The Secure Data Dissemination Scheme

In this Section, we present the SCR data dissemination scheme for sensor networks. SCR adopts the key management scheme in Subsection 3.1. After handshake, sensor nodes can use localization algorithms such as References [22,23] to compute their locations. Based on the locations of a source sensor and the base station, a serial of cells that need to participate in routing are determined, these cells are in the direction from the source to the destination, and they are referred to as *routing cells*. Consider the network in Figure 1, suppose the source is node S and the destination is node D, a straight line L is drawn between the geographic centers of cell 9 and cell 8. The cells with which line L intercepts are the *routing cells*—10 and 7 in this example. Source node S knows the *routing cells* based on its location and the grid structure.

It is possible that some of the nodes in the *routing cells* are compromised by an adversary, or some of the *routing cells* are attacked by an adversary. The compromised nodes, or the adversary can initiate many attacks against SCR routing, like selective forwarding, Sybil attacks, jamming etc. To avoid or reduce the effect of such attacks, two or more backup paths are provided in SCR routing. The backup paths are determined by the source node, and the actual paths are selected with some randomness. This reduces the possibility that an adversary attacks both the *routing cells* and backup paths with a small investment. For example, one backup path from node S to node D in Figure 1 could be cell: 9, 10, 11, 12, and 8. When attacks on some *routing cells* are detected, one or more backup paths will be used to forward packets between sensor nodes and the base station.

Recall that we compute a MAC for each encrypted message in SCR to provide authentication and integrity of messages. For simplicity, in the following we do not explicitly list the MAC part of each encrypted message. The detail of SCR data dissemination scheme is presented below.

(1) Based on the locations of the source and the base station, the source sensor node S determines the *routing cells*. A line L is drawn between the cell centers of the source node S and the BS.

(2) The line L intercepts with several cells, and these *routing cells* are denoted as C0, C1, C2, . . . , Ck starting from the cell of source node S. Node S knows the *routing cells* based on its location and the grid structure. Then node S records the *routing cells* in a cell_list field, which is stored in the header of the data packet. The header contains the following fields: session_id, source_id, and cell_-list. session_id plus source_id uniquely determines a data transmission session.

(3) The data packet is first sent from the source node S to a node in cell C1. Contention-based mechanism is used in medium access control layer, for example, CSMA/CA or IEEE 802.11 with the request-to-send (RTS)/clear-to-send (CTS) mechanism enabled. First, a {RTS} K_{S} is broadcasted to neighbor nodes, and RTS is encrypted with S's broadcast key K_{S}. In addition to the

standard fields, there is an unencrypted field: next_cell in the RTS packet. The next_cell refers to the next cell to relay the data packet. For the RTS from the source node S, the next_cell is C1. Based on the next_cell field, only the nodes in cell C1 will response to this RTS packet. Nodes in cell C1 send a $\{\text{CTS}\}\ K_S$ packet to S with a delay of $t_d = \alpha(t)/E + t_r$, where t_d is the delay, E is the remaining energy of the sensor, $\alpha(t)$ is a system parameter representing a decreasing function of the local time t in the sensor, and t_r is a small random back-off time. The CTS is encrypted with S's broadcast key K_S. It is important to encrypt the RTS and CTS packets, otherwise an adversary can generate a false RTS or CTS packet, and interfere with the transmissions. The idea is to let one of the sensors with more remaining energy to response with a CTS packet and then becomes the node that relays the data packet. The delay $t_d = \alpha(t)/E + t_r$ includes two parts. The first part $\alpha(t)/E$ is to ensure that a node with more remaining energy in a cell participates in routing. Since sensor networks usually have a large number of nodes. There might be several nodes with similar remaining energy E in the same cell. In order to avoid the simultaneous transmissions of CTS, a small (compared to $\alpha(t)/E$) random back off time t_r is added to the delay. $\alpha(t)$ is chosen to be large enough so that the transmission of CTS from different nodes will not overlap. But $\alpha(t)$ should not be too large which will cause large routing delay. Since the remaining energy of sensor nodes decreases with time, $\alpha(t)$ is a decreasing function of time t. This avoids long delay when there is not much energy left in the nodes. Note that local time t does not need to be synchronized.

(4) Assume that node R1 is the first node that sends CTS back to node S. Other nodes in cell C1 will not send CTS when they overhear the transmission of the CTS. Then S sends the data packet: packet_ID + $\{\text{Data}\}\ K_{SR1}$ to node R1, where the data packet is encrypted with the shared secret key between S and R1, that is, K_{SR1}, and packet_ID (not encrypted) is a local ID assigned by node S. packet_ID is only used by S to monitor the packet transmission in the next step, that is, from R1 to the next node. Then R1 becomes the relay node in cell C1.

(5) Node R1 set the next_cell as C2 and proceeds the similar way as above, and sends the packet to a node R2 in cell C2. To guarantee the delivery, each relay node is responsible for confirming that its successor has successfully received the packet. This may be implemented by the transmitter monitoring the packet just sent out to the next node and overhearing if that node has passed it on within a time period using the packet_ID field. Of course, if a link level acknowledgement is supported by the medium access control layer protocol, for example, 802.11, the above passive acknowledgement scheme is unnecessary. If a link level acknowledgement is used, link level encryption is needed for the acknowledgement. The transmitted data packet has to be kept in the buffer before its receipt has been confirmed. The acknowledgement scheme reduces the impact of channel or node error and can detect selective forwarding attack.

(6) If R1 does not get any acknowledgement within a certain time period, R1 will re-transmit the data packet to a node in cell C2. At the next time, the relay node in cell C2 may be a different node other than the previous relay node. If the transmission to cell C2 fails again, a backup path will be used.

(7) The process continues until the data packet reaches the base station.

In the above SCR routing protocol, there is only one node in each *routing cell* that receives the data packet and transmits it to the next cell. Thus SCR is also an energy efficient routing protocol. To further save energy, nodes that do not forward packet may go to sleep for a while, and then wake up, like the scheme described in Reference [4]. We will not discuss this issue in this paper. Once a source node knows the location of the base station, no route discovery is needed in SCR. So SCR has small routing latency. Also, failure of some nodes will not affect SCR, since only an active node may become a relay node.

4. Security Analysis

In this Section, we analyze the security features of SCR. Due to the limited storage in sensor nodes, all cryptographic primitives, that is, encryption, message authentication code, random number generator, use a single block cipher for code reuse. In the following experiments, RC4 is used as the block cipher. The security configuration is discussed in the following.

- *Data Confidentiality* is provided by symmetric encryption (RC4).

- *Data Authentication* allows the receiver to verify that the data really was sent by the claimed sender. In SCR routing protocol, data authentication is achieved via a symmetric mechanism: each sensor node and the BS compute a message authentication code with the individual key of the sensor.
- *Data Integrity* ensures the receiver that the received data is not altered in transit by an adversary, and it is achieved by MAC.
- *Data Freshness* ensures each message is fresh, that is, the data is recent, but not a replay of old messages from an adversary. Weak freshness is provided in SCR, which provides partial message ordering. Weak freshness is provided by including session_id and source_id (in step 2 of Subsection 3.3).

The SCR routing protocol can defend against several attacks on sensor networks. In the following, we analyze the security of SCR defending against various attacks.

4.1. Defending Against Various Routing Attacks

4.1.1. Defense against the Sybil attack

In Sybil attack [16], a single node presents multiple identities to other nodes in the network.

4.1.1.1. Defense the Sybil attack as original sensor nodes. Authentication is used to ensure one node cannot pretend to be other nodes, that is, when a sensor node A sends a packet to another node B, A must present a message authentication code computed using the shared pairwise key K_{AB} between A and B. Since the pairwise key K_{AB} is only known by node A and node B, no adversary node can pretend to be node A. Thus, the Sybil attack does not work.

4.1.2. Defense against the wormhole and sinkhole attacks

In the wormhole attack [15], an adversary tunnels messages received in one part of the network over a low-latency link and replays them in a different part. Wormhole attacks usually involve two distant malicious nodes colluding to understate their distance from each other by relaying packets along an out-of-bound channel available only to the attacker. In a sinkhole attack, the adversary's goal is to lure nearly all the traffic from a particular area through a compromised node, creating a metaphorical sinkhole with the adversary at the center.

In SCR routing, given the cell structure and the locations of the source sensor node and the base station, a serial of cells is determined as the *routing cells* (or backup routing cells). The packet is forwarded only by nodes in the *routing cells*, and will not be forwarded by nodes elsewhere. Thus, the SCR routing protocol is resistant to wormhole attack and sinkhole attack. Let us consider several attack scenarios in the following.

(1) *Sinkhole Attack*: One scenario is that a powerful adversary (like a laptop) has a real, high-quality route to the base station, and it broadcasts this route to its neighbors. However, the neighbor sensors will not use the advertised route, since they will only route the packets via the *routing cells*. Thus, this attack does not work.
(2) *Sinkhole Attack*: An adversary broadcasts to its neighbors about an artificial link to the base station. This attack does not work for the same reason as above.
(3) *Wormhole Attack*: Two malicious nodes create an out-of-bound tunnel, and broadcast the route to the neighbors. This attack does not work for the same reason as in (1).

4.1.3. Defense against the selective forwarding attack

A selective forwarding attack on the routing protocol can be: a powerful adversary always serves as a relay node in a cell, and she can selectively forward some packets while dropping most packets. We propose the following schemes to defend this attack:

(1) Each node monitors the relay activities of its neighbor nodes. If one node serves as the relay node for more than M_1 times in a given time period T, neighbor nodes will send an alarm to the base station and other neighbor nodes.
(2) If one node serves as the relay node for more than M_2 times, the upstream node, that is, the sender, will send the packet to another node in the cell, encrypted with the corresponding shared pairwise key, where M_1, M_2, and T are system parameters, and the proper values can be determined by simulation experiments.

4.1.4. Defense against the Hello flood attack

Since sensor nodes use the three-way handshake protocol to establish neighborhood relationship, the

hello flood attack does not work. In other words, a powerful adversary node cannot lure nearby sensor nodes with the unidirectional link.

4.1.5. Defense against the clone attack

The adversary can compromise a node C and obtain the keys in the node, and then she can clone several nodes and load with the same keys, then distribute the cloned nodes at different locations of the sensor network. The cloned node could be either an existing sensor or a newly deployed sensor. We first discuss the case where the compromised node C is an existing sensor (i.e., initially deployed). However, a cloned node of C (say D) cannot establish pairwise keys with nodes that are not the neighbors of the compromised node C. D only has C's individual key, the pairwise keys with C's neighbors and the broadcast keys of C and C's neighbors. D cannot use the scheme in Subsection 3.2 to establish pairwise keys with its neighbors, since the BS can recognize that the individual key from D is actually node C's individual key. Thus, the cloned nodes could only have damage in the neighborhood of the compromised node C, and the security impact of a compromised node is localized.

In the second case, the compromised node C is a newly deployed sensor. Recall that each new sensor needs to contact the BS to pass the verification. When the BS finds out there are two or more new sensors that have the same individual key, the BS will know that node C has been cloned, and the BS can notice all sensors that any node with ID C has been compromised.

4.2. Security Performance

We design experiments to test the security performance of SCR routing protocol. One experiment is to measure the effectiveness of SCR against selective forwarding attack. We compare the performance of single-path (SP) routing and SCR. SP routing uses the pre-determined optimum route to forward data packets, if there is one (or more) malicious node on the route conducts selective forwarding attack, then all the packets dropped by her are lost without being detected. Of course, the packet dropping may be detected in other application layers, such as the application layer.

On the other hand, when SCR is used as the routing strategy, as discussed in Subsection 4.1.3 if one sensor node forwards more than M_1 packets in a certain time period T, an alarm will be sent to the base station. Also, an upstream node only forwards packets to the same downstream node for no more than M_2 times, where M_1, M_2, and T are system parameters. Thus, SCR can reduce the damage from selective forwarding attack.

Table I. Test result of the selective forwarding attack.

P_d	0.2	0.4	0.6	0.8	0.9
Drop rate in SP	0.2	0.4	0.6	0.8	0.9
Drop rate in SCR	0.07	0.14	0.21	0.26	0.32

For SP routing, one node on the route is set as the malicious node. For SCR, one node in a *routing cell* is set as the malicious node, and the malicious node always responses a CTS to a RTS it hears without any delay. In the test, a constant bit rate (CBR) traffic is set up between a sensor node S and the BS with 10-hop distance. The CBR traffic sends two packets per second, and it lasts for 30 s. So there are totally 60 packets in the session. The malicious node randomly drops packets according to a probability P_d. As one can imagine, the packet drop rate in SCR depends on the CBR traffic parameters and the routing parameters including M_1, M_2, and T. One of the test results is listed in Table I, where $M_1 = 20, M_2 = 10$, and $T = 30$ s.

We observe that the drop rate in SP is the same as the selective drop rate P_d, since in SP all packets pass the malicious node. Given the above parameters, one can calculate the drop rate in SCR. A malicious node can relay no more than 20 packets in the 30 min. So the drop rate in SCR is $20 \times P_d/60 = P_d/3$. Table I shows that SCR can significantly reduce the packet drop rate from the selective forwarding attack.

5. Evaluation of Routing Performance

In this section, we evaluate the efficiency and effectiveness of the SCR routing protocol through experiments. Typical sensor nodes have limited energy supply. A good secure routing protocol of sensor networks should be energy efficient. To evaluate the routing performance of SCR, we use QualNet to compare the performance of SCR with a popular sensor network routing protocol—directed diffusion (DD) [1]. The underlying medium access control protocol is 802.11 distributed coordination function (DCF). The default simulation testbed has 1 base station and 300 sensor nodes randomly distributed in a $300 \times 300\,m^2$ area, of which 20 nodes are sources.

Each simulation run lasts for 600 s, and each result is averaged over five random network topologies. A source generates one data packet per second. Each data packet is 64 bytes. The transmission range of each sensor node is 60 m. The side length of a cell is set as $a = R/2 = 30$ m. We studied the effect of different cell size on routing performance in our previous work [5]. One of the results is that $R/2$ is a good value for the cell size that tradeoffs the routing performance and the number of cells.

In the experiments, the energy consumption and routing overhead of SCR include those from all cryptographic primitives (encryption, MAC calculation, random number generator) in SCR. Note that DD does not use any cryptographic primitives and does not provide secure routing.

5.1. Routing Performance under Different Node Density

In this subsection, we compare the routing performance of SCR and DD when there is no attack placed on the sensor network. First we compared the delivery ratio and energy consumption for different node density. For the fixed $300 \times 300\,\text{m}^2$ routing area, we change the number of sensor nodes from 100 to 500 with an increment of 100. The delivery ratios under SCR and DD are plotted in Figure 3. As we can see that the delivery ratio of both SCR and DD increases as the node density increases. When the node density increases, there are more nodes in one cell, and there are more candidates to serve as the relay node in SCR. This is why the delivery ratio under SCR increases as node density increases. In DD there are more nodes involved routing when node density increases, thus the delivery ratio of DD also increases. Figure 3 shows that SCR routing has higher delivery ratio than DD. In SCR, only an active node with more remaining energy may become the relay node, and each relay node is responsible for confirming that its successor has successfully received the packet (step 5 in Subsection 3.3), thus the delivery ratio in SCR is higher than DD.

The total energy consumptions of SCR and DD are reported in Figure 4. The energy consumptions of both protocols grow with node density. In SCR, the main reason is that more power is dissipated for overhearing when every node has more neighbors, thus the energy consumed by SCR only increases a little bit when node density is high. However, the energy consumption of DD increases much faster than SCR, and it becomes very large when node density is high. This is because the more and more nodes are involved in disseminating 'interest' and 'gradient' of DD when node density increases.

5.2. Different Source–Base Station Distance

Figures 5 and 6 report the delivery ratio and energy consumption for different source–base station distances.

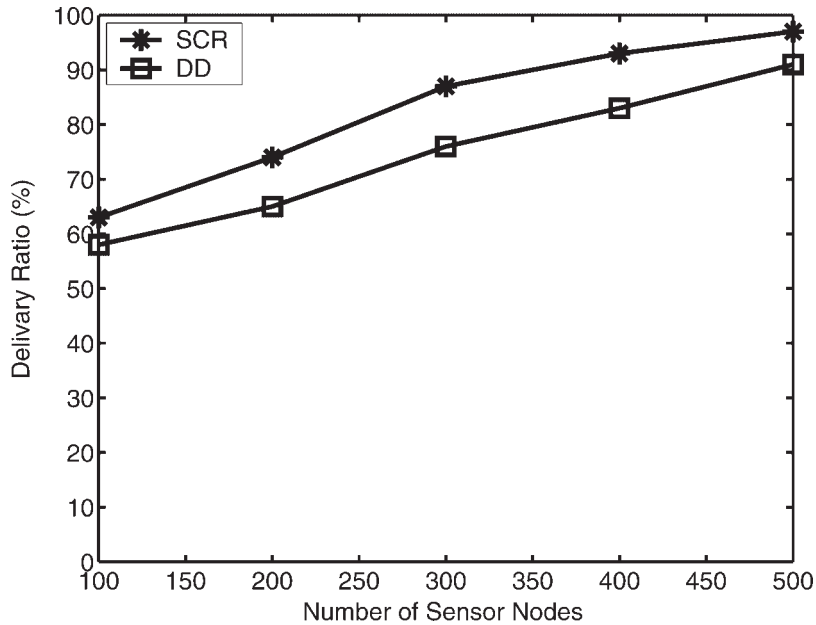


Fig. 3. Delivery ratio versus node density.

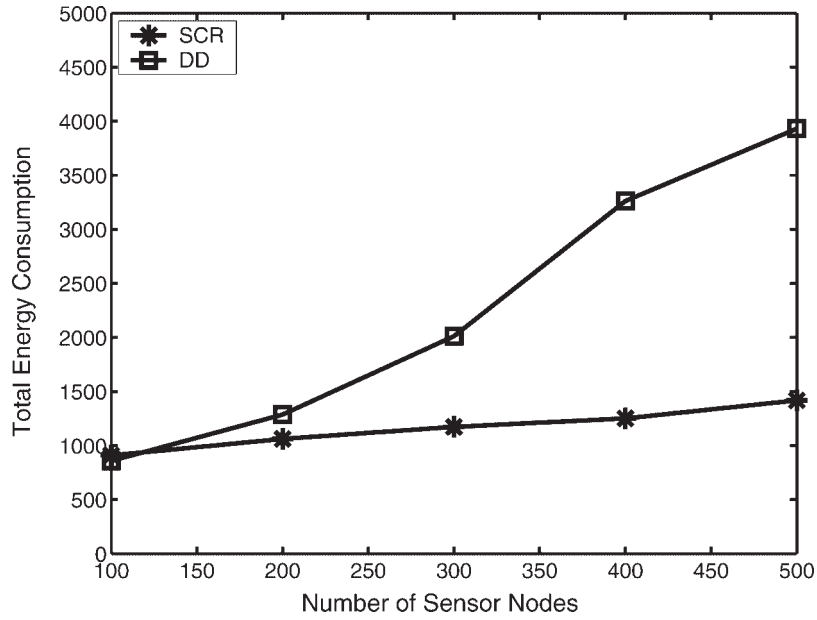


Fig. 4. Energy consumption versus node density.

The delivery ratio of SCR and DD slightly decreases for larger source–base station distance, and DD drops faster than SCR. For any source–base station distance, the delivery ratio of SCR is the higher than DD, and the reason is similar as above. The total energy consumed by SCR and DD increases as distance increases, as shown in Figure 6. However, the increase in DD is much faster than in SCR. As the source–base station distance becomes large, more and more nodes are involved in routing in DD, and much more energy is consumed. In SCR, the number of nodes that forward data is about the same as the number of

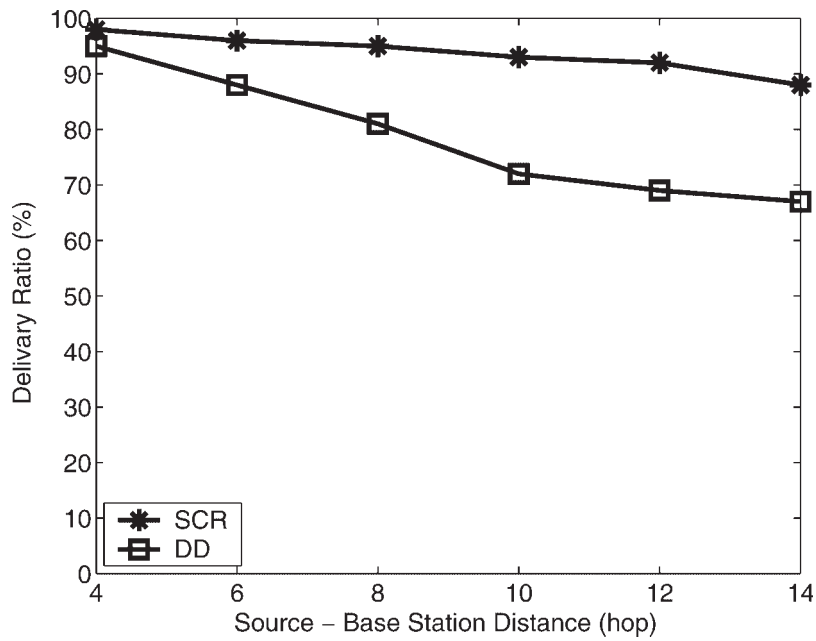


Fig. 5. Delivery ratio versus path length.

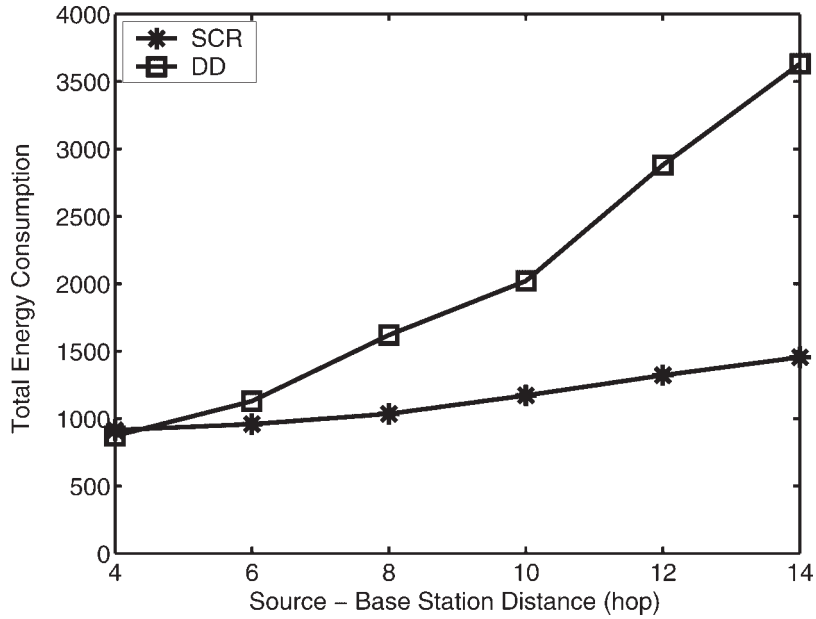


Fig. 6. Energy consumption versus path length.

hops, so the energy consumption increases slowly in SCR.

5.3. Different Node Failure Probability

Figures 7 and 8 show the change of delivery ratio and energy consumption for different sensor node failure probabilityP_n. The delivery ratios of both SCR and DD decrease as node failure probability increases. However, the decrease in SCR is much slower than DD. In SCR, each cell has several nodes that can serve as relay node and forward data packet, so the node failure has less impact on the route in SCR. Figure 7 shows that the delivery ratio of SCR is always higher than 90% when P_n is less than 20%. As we can see from Figure 8, the energy consumptions of both SCR

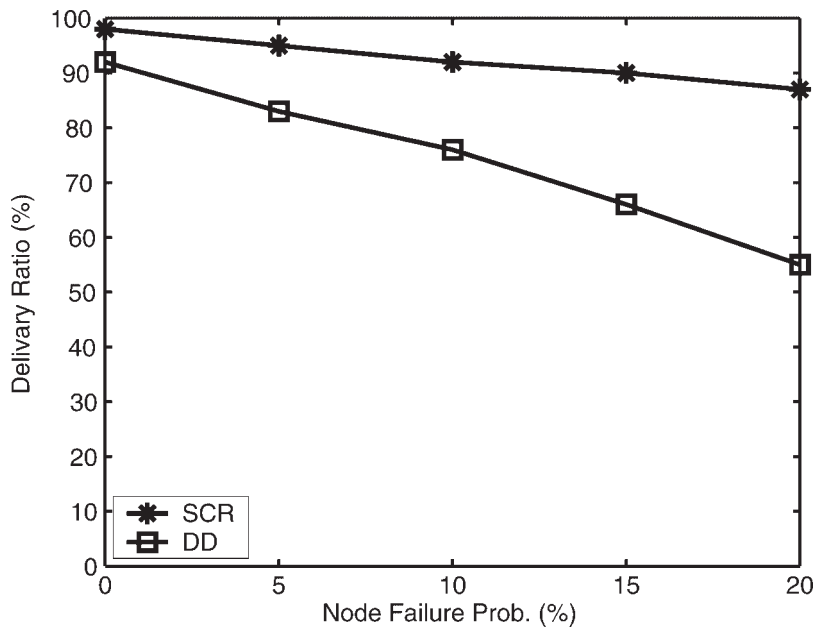


Fig. 7. Delivery ratio versus node failure probability.

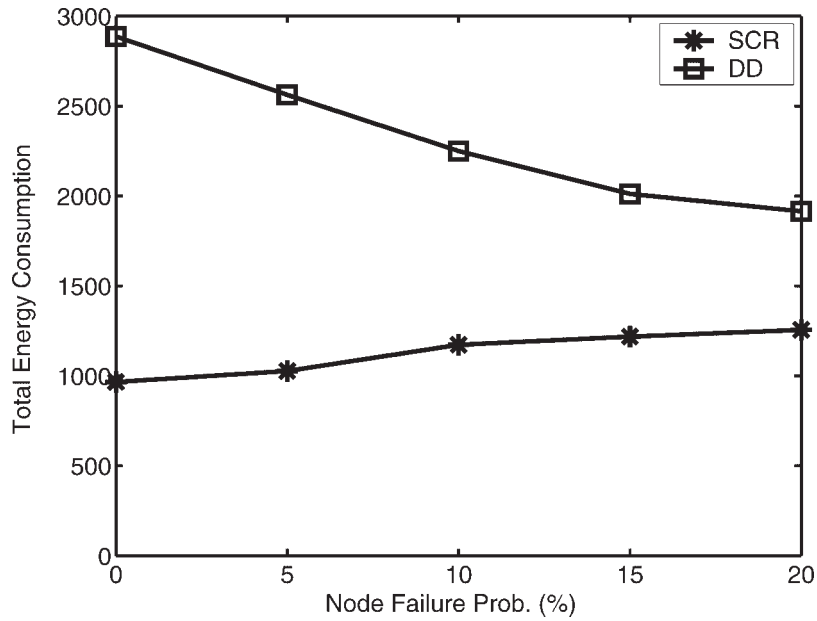


Fig. 8. Energy consumption versus node failure probability.

and DD decrease as P_n increases. The energy consumption of DD decreases faster than SCR, since larger P_n means more nodes may fail and less nodes are involved in routing in DD.

In summary, the simulation experiments show that SCR routing protocol has high delivery ratio and low energy consumption than directed diffusion, even though directed diffusion does use any security primitives.

6. Related Works

Several recent works [8–21] have addressed security problems in sensor networks. Eschenauer and Gligor [11] present a key management scheme for sensor networks based on probabilistic key pre-distribution. Chan *et al*. [9] extend this scheme and present three mechanisms for key establishment. Liu and Ning [8] propose a key management scheme based on the key pre-distribution approach to establish pairwise keys in sensor networks.

Perrig *et al*. [14] present two building block security protocols optimized for use in sensor networks, SNEP, and μTESLA. SNEP provides confidentiality, authentication, and freshness between nodes and the base station, and μTESLA provides authenticated broadcast. However, their scheme uses the base station to help establish a pairwise key between two nodes, which limits its scalability and leaves it subject to Sybil attacks. In contrast, in our key management scheme, pairwise keys are established in a distributed fashion without the involvement of the base station.

Karlof *et al*. [17] describe TinySec, a link layer security mechanism using a single pre-loaded fixed group key for both encryption and authentication, assuming no node compromises. They also discuss the impact of different keying mechanisms on the effectiveness of in-network processing in sensor networks. Deng *et al*. [20] discuss several security mechanisms for supporting in-network processing in hierarchical sensor networks. In Reference [13], Wood and Stankovic identify a number of Denial of Service attacks in sensor networks. In Reference [3], Karlof and Wagner describe several security attacks on routing protocols for sensor networks. In Reference [19], Ye *et al*. propose a statistical en-route scheme that detects false data (injected by compromised nodes) by the en-route nodes and base station with some probability. However, they did not address the secure routing of valid data in sensor networks.

Zhu *et al*. present a distributed key management protocol for sensor networks in Reference [10]. They also assume that there is a low bound on the time interval T_{min} that for an adversary to compromise a sensor node, and a globally shared key is used before T_{min} to establish pairwise keys between each sensor and its neighbors. Our key management scheme makes the same assumption.

The main difference between our security schemes and other sensor network security protocols [8–21] are: (1) we use a three-way handshake protocol to establish neighborhood relationship between sensors, which can defend against Hello flood attack; (2) we consider security issue during the protocol design of SCR. Several security features are incorporated in SCR to defend various attacks, for example, the RTS and CTS packets are encrypted to prevent medium access control interference from an adversary; to defend against selective forwarding attack, each node can only relay packet for at most M_1 times within a time period T; (3) we design an efficient key setup scheme for sensor nodes deployed after initialization phase.

Several secure routing protocols have been proposed for mobile ad hoc networks (MANETs). Some of the secure routing protocols are based on public key cryptography, such as References [25–27]. However, public key cryptography is too expensive for sensor nodes. Security protocols for sensors networks must rely exclusively on efficient symmetric key cryptography. There are several MANET secure routing protocols which are based on symmetric key cryptography, such as References [28–30]. However, these protocols are based on source routing or distance vector protocols and are unsuitable for sensor networks. They are too expensive in terms of node state and packet overhead and are designed to find and establish routes between *any* pair of nodes—which is different from the many-to-one traffic pattern dominant in sensor networks.

Routing protocols and security schemes (like key management, authentication) in sensor networks have been studied separately in the past. Although one may be able to add security features to the existing routing protocols, certain security schemes may not work efficiently or even correctly with certain routing protocols. Often the routing protocols and/or security schemes have to be modified to fit with each other, which can cause the degradation of routing performance. The best way to provide secure and efficient routing in sensor networks is to consider security during the design time of the routing protocol. We designed the SCR routing protocol in such a way.

7. Conclusions

In this paper, we have presented an efficient key management scheme and a novel secure routing protocol for sensor networks—SCR routing protocol. In addition, we have presented an efficient key setup scheme for sensors deployed after initialization phase. In SCR routing protocol, the routing area is divided into cells. Based on the locations of source and base station, packets are forwarded by *routing cells* along the direction from source to base station. The nature of SCR routing (cell relay via *routing cells*) makes it resistant to spoofed routing information, selective forwarding, and sinkhole and wormhole attacks. The three-way handshake can defend against Sybil attack and Hello flood attack. In SCR routing, only an active node with more remaining energy (than other nodes) in the *routing cells* forwards packet, thus the energy consumption is low and the traffic load is balanced among sensors. SCR effectively reduce the impact of node failure and channel error, and provides high-delivery ratio. The security analysis has shown that SCR can defend against several sophisticated attacks. Our QualNet simulations have demonstrated the efficiency and effectiveness of SCR, that is, SCR has higher delivery ratio and lower energy consumption than DD.

References

1. Intanagonwiwat C, Govindan R, Estrin D. Directed diffusion: a scalable and robust communication paradigm for sensor networks. *Proceedings of ACM MOBICOM'00*, Boston, MA, August 2000.
2. Ye F, Luo H, Cheng J, Lu S, Zhang L. A two tier data dissemination model for large-scale wireless sensor networks. *Proceedings of ACM MOBICOM'02*, Atlanta, GA, September 2002.
3. Karlof C, Wagner D. Secure routing in sensor networks: attacks and countermeasures. *IEEE First International Workshop on Sensor Network Protocols and Applications*, May 11, 2003.
4. Ye W, Heidemann J, Estrin D. An energy-efficient MAC protocol for wireless sensor networks. *Proceedings of IEEE INFOCOM 2002*.
5. Du X. QoS routing based on multi-class nodes for mobile ad hoc networks. *Elsevier Journal of Ad Hoc Networks* 2004; **2/3**: 241–254.
6. Liao WH, Tseng YC, Sheu JP. GRID: a fully location-aware routing protocol for mobile ad hoc networks. *Telecommunication Systems* 2001; **18**(1–3): 37–60.
7. Ye F, Lu S, Zhang L. Gradient broadcast: a robust, long lived sensor network. *Wireless Networks* 2005; **11**(3): 285–298.
8. Liu D, Ning P. Establishing pairwise keys in distributed sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*, 2006; 52–61.
9. Chan H, Perrig A, Song D. Random key predistribution schemes for sensor networks. *Proceedings of the IEEE Symposium on Security and Privacy*, May 2003; 197–213.
10. Zhu S, Setia S, Jajodia S. LEAP: efficient security mechanisms for large-scale distributed sensor networks. *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*, Washington D.C., October, 2003.

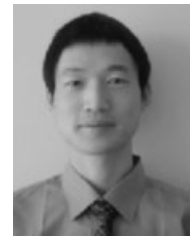
11. Eschenauer L, Gligor VD. A key management scheme for distributed sensor networks. *Proceedings of the 9th ACM Conference on Computer and Communication Security*, November 2002; 41–47.
12. Chan H, Perrig A. Security and privacy in sensor networks. *Computer* 2003; **36**(10): 103–105.
13. Wood AD, Stankovic JA. Denial of service in sensor networks. *Computer* 2002; **35**(10): 54–62.
14. Perrig A, Szewczyk R, Tygar JD, *et al*. SPINS: Security Protocols for Sensor Networks. *Proceedings of MOBICOM*, 2001.
15. Hu Y-C, Perrig A, Johnson DB. Wormhole detection in wireless ad hoc networks. Department of Computer Science, Rice University, Technical Report TR01-384.
16. Douceur JR. The Sybil Attack. *Proceedings of IPTPS'02*, March 2002.
17. Karlof C, Sastry N, Wagner D. TinySec: a link layer security architecture for wireless sensor networks. *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, November 2004.
18. Du W, Deng J, Han Y, Varshney P. A pairwise key predistribution scheme for wireless sensor networks. *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)* 2003; 42–51.
19. Ye F, Luo H, Lu S, Zhang L. Statistical en-route detection and filtering of injected false data in sensor networks. In *Proceedings of IEEE INFOCOM 2004*.
20. Deng J, Han R, Mishra S. Security support for in-network processing in wireless sensor networks. *Proceedings of the First ACM Workshop on the Security of Ad Hoc and Sensor Networks*, 2003.
21. Du W, Deng J, Han YS, *et al*. A key management scheme for wireless sensor networks using deployment knowledge. *Proceedings of IEEE INFOCOM 2004.*
22. Savvides A, Han C, Strivastava M. Dynamic fine-grained localization in ad-hoc networks of sensors. *Proceedings of ACM MOBICOM'01* 2001; 166–179.
23. Doherty L, Ghaoui L El, Pister KSJ. Convex position estimation in wireless sensor networks. *Proceedings of IEEE INFOCOM 2001*, Anchorage, AK, April 2001.
24. Heinzelman W, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocols for wireless microsensor networks. *Proceedings of Hawaiians Int'l Conference on Systems Science*, January 2000.
25. Kong J, Zerfos P, Luo H, Lu S, Zhang L. Providing robust and ubiquitous security support for mobile ad-hoc networks. *Proceedings of ICNP* 2001; 251–260.
26. Sanzgiri K, Dahill B, Levine BN, Shields C, Belding-Royer E. A secure routing protocol for ad hoc networks. *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, November 2002.
27. Kong J, Luo H, Xu K, Gu DL, Gerla M, Lu S. Adaptive security for multi-layer ad-hoc networks. *Special Issue of Wireless Communications and Mobile Computing, Wiley Interscience Press*, 2002.
28. Hu Y-C, Johnson DB, Perrig A. SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks. *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications* June 2002; 3–13.
29. Hu Y-C, Perrig A, Johnson DB. Ariadne: a secure on-demand routing protocol for ad hoc networks. *Proceedings of the 8th ACM International Conference on Mobile Computing and Networking*, September 2002.
30. Papadimitratos P, Haas Z. Secure routing for mobile ad hoc networks. *Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, January 2002.

Authors' Biographies

Xiaojiang (James) Du is an assistant professor in Department of Computer Science, North Dakota State University. Dr. Du earned his B.E. degree from Tsinghua University, Beijing, China in 1996, and his M.S. degree and Ph.D. from University of Maryland, College Park in 2002 and 2003, respectively, all in electrical engineering. His research interests are wireless sensor networks, mobile *ad hoc* networks, wireless networks, computer networks, network security, and network management. Dr. Du served as a NSF panelist. He is a guest editor for International Journal of Security and Networks (IJSN), Special Issue on Security Issues in Sensor Networks in 2005. Dr. Du is a TPC member for several international conferences (including IEEE ICC 06, Globecom 06 & 05, BroadNets 05, WirelessCom 05, IPCCC 06 and 05, and BroadWise 04), and he is the program chair of Computer and Network Security Symposium of IEEE International Wireless Communications and Mobile Computing Conference 2006.

Yang Xiao worked at Micro Linear as a medium access control (MAC) architect involving the IEEE 802.11 standard enhancement work before he joined the Department of Computer Science at The University of Memphis in 2002. Dr. Xiao was a voting member of the IEEE 802.11 Working Group from 2001 to 2004. He is a senior member of the IEEE. He currently serves as editor-in-chief for *International Journal of Security and Networks (IJSN)* and for *International Journal of Sensor Networks (IJSNet)*. He serves as an associate editor or on editorial boards for the following refereed journals: *International Journal of Communication Systems (Wiley)*, (*Wiley*) *Wireless Communications and Mobile Computing (WCMC)*, *EURASIP Journal on Wireless Communications and Networking (WCN)*, and *International Journal of Wireless and Mobile Computing (IJWMC)*. He serves as a (lead) guest editor for *IJSN* journal, Special Issue on Security Issues in Sensor Networks in 2005, as a (lead) guest editor for *EURASIP WCN*, Special Issue on 'Wireless Network Security' in 2005, as a (sole) guest editor for *(Elsevier) Computer Communications journal*, special Issue on 'Energy-Efficient Scheduling and MAC for Sensor Networks, WPANs, WLANs, and WMANs' in 2005, as a (lead) guest editor for *(Wiley) WCMC journal*, special Issue on 'Mobility, Paging and Quality of Service Management for Future Wireless Networks' in 2004, as a (lead) guest editor for *IJWMC journal*, special Issue on 'Medium Access Control for WLANs, WPANs, Ad Hoc Networks, and Sensor Networks' in 2004, and as an associate guest editor for *International Journal of High Performance Computing and Networking*, special issue on 'Parallel and Distributed Computing, Applications and Technologies' in 2003. He serves as a referee for many funding agencies, as well as a panelist for the US NSF. His research areas include wireless networks and network security.

Hsiao-Hwa Chen received B.Sc. and M.Sc. degrees from Zhejiang University, China, and Ph.D. from the University of Oulu, Finland, in 1982, 1985, and 1990, respectively, all in electrical engineering. He worked with Academy of Finland for the research on spread spectrum communications as a research associate during 1991–1993 and the National University of Singapore as a lecturer and then a senior lecturer from 1992 to 1997. He joined Department of Electrical Engineering, National Chung Hsing University, Taiwan, as an associate Professor in 1997 and was promoted to a full-Professor in 2000. In 2001, he moved to National Sun Yat-Sen University, Taiwan, as the founding director of the Institute of Communications Engineering of the University. Under his leadership the institute was ranked the 2nd place in the country in terms of SCI journal publications and National Science Council funding per faculty in 2004. He has been a visiting professor to Department of Electrical Engineering, University of Kaiserslautern, Germany, in 1999, the Institute of Applied Physics, Tsukuba University, Japan, in 2000, and Institute of Experimental Mathematics, University of Essen, Germany in 2002. He is a recipient of numerous Research and Teaching Awards from the National Science Council and Ministry of Education, Taiwan, from 1998 to 2001. He has authored or co-authored over 120 technical papers in major international journals and conferences, and three books and several book chapters in the areas of communications. He served as a TPC member and symposium chair of major international conferences, including IEEE VTC, IEEE ICC, and IEEE Globecom etc. He served or is serving as member of the editor and guest editor for *IEEE Communications Magazine*, *IEEE JSAC*, *Wireless Communications and Mobile Computing (WCMC)*, *Journal and International Journal of Communication Systems* etc. He has been a guest professor of Zhejiang University, China, since 2003.



Qishi Wu received the B.S. degree in remote sensing and GIS from Zhejiang University, People's Republic of China in 1995, the M.S. degree in geomatics from Purdue University in 2000, and the Ph.D. in computer science from Louisiana State University in 2003. He is currently a research fellow in the Computer Science and Mathematics Division at Oak Ridge National Laboratory. His research interests include computer networks, remote visualization, distributed sensor networks, high performance computing, algorithms, and artificial intelligence. He is a member of the IEEE and IEEE Computer Society.