Ann. Telecommun. (2009) 64:723–734
DOI 10.1007/s12243-009-0088-z

Energy conserving security mechanisms for wireless sensor networks

Md. Abdul Hamid · Choong Seon Hong

Received: 28 August 2008 / Accepted: 26 February 2009 / Published online: 27 March 2009

Abstract Since wireless sensor networks are emerging as innovative technologies for realizing a variety of functions through a number of compact sensor nodes, security must be justified and ensured prior to their deployment. An adversary may compromise sensor nodes, forcing them to generate undesired data, and propagation of these data packets through the network results in wasteful energy consumption. We develop a security mechanism to detect energy-consuming useless packets, assuming that a sensor node is able to generate multiple message authentication codes (MAC) using preshared secrets. The forwarding nodes along the path verify the validity of the packet by checking the authenticity of the attached MACs. This mechanism performs well when a malicious node does not have all the cryptographic keys. However, packets, generated by the malicious node having all the keys, would be considered as legitimate, and thus, the forwarding nodes become unable to detect and discard them. To deal with this problem, we devise another mechanism in which each forwarding node is capable of checking such suspicious nodes. We have quantified the security strength through analysis and simulations to show that the proposed mechanisms make the entire network energy conserving.

M. A. Hamid · C. S. Hong (✉)
Networking Lab, Department of Computer Engineering,
School of Electronics and Information,
Kyung Hee University, 1 Seocheon, Giheung,
Yongin, Gyeonggi 446-701, South Korea
e-mail: cshong@khu.ac.kr

M. A. Hamid
e-mail: hamid@networking.khu.ac.kr

Keywords Wireless sensor networks · Key distribution · Security · Legitimacy check · Energy conservation

1 Introduction

The expected achievement of a wireless sensor network (WSN) is to produce, over an extended period of time, global information from local data sensed by individual sensor nodes. The characteristics of sensor networks differ from traditional wireless networks in a way where energy conservation and self-configuration are the primary goals, while pernode fairness and latency are less important. Sensor networks usually consist of a large number of ultrasmall autonomous devices. Each device, called a sensor node, is battery-powered and equipped with integrated sensors, data processing capabilities, and short-range radio communications. Sensor networks are being deployed for a wide variety of applications [1], including military sensing and tracking, environment monitoring, patient monitoring and tracking, smart environments, etc. When sensor networks are deployed in a hostile environment, security becomes extremely important, as they are prone to different types of malicious attacks. For example, an adversary can easily listen to the traffic, impersonate one of the network nodes, or intentionally provide misleading information to other nodes.

The perceived usefulness of sensor network will be dangerously curtailed if misbehavior (by an adversary or a compromised node) occurs that threats the work of the network by perturbing the information produced, stopping production, or proliferating information. Implementing security mechanisms to restrict mass flow of illegitimate information can increase the lifetime of the entire network, thereby conserving the energy. In this paper, we present a security scheme to identify and restrict the illegitimate data packets to flow in the network. Intuitively, early detection and discard of those packets will allow the entire network to conserve energy, which is one of the primary goals in the design of resource-constrained sensor networks.

Networks may suffer in many ways due to the insertion of junk or misleading packets. First, it may cause congestion in the network and, therefore, the data acquisition point (base station) may loose its perceived goal from extracting information in a timely fashion. Second, attackers may intentionally insert wrong information in the data packets and the base station will proceed with this information. So, the main defense is to ensure that a route may serve the legitimate packets by detecting and preventing untrustworthy or replayed packets in the route. Thus, the overall lifetime of the network can be increased by reducing the transmission/reception power consumed by the large number of unexpected traffic.

We exploit the multiple complementary tree-based key predistribution protocol [2] to develop the detection technique in which each sensor is preassigned secret keys from complementary trees. When a sensor senses an event, it generates multiple message authentication codes (MAC) using distinct secret keys from distinct trees and appends these MACs to the sensed data. As the data packet traverses towards the base station, intermediate nodes verify the legitimacy of the data packet by checking the authenticity of the appended MACs. If the packet is detected to be illegitimate, the forwarding node discards the packet to save the wasteful energy that would have been consumed if the packet were traversing all the way to the base station. The security protocol performs well when an adversary does not have cryptographic keys from all the complementary trees. However, if an attacker node (compromised by the adversary) has all the keys, it can generate packets that would be considered as legitimate to the forwarding nodes and, hence, cannot be discarded. Such a malicious node may disrupt the normal operation of the network by continuously sending data to deplete the channel capacity in their vicinity and, hence, prevent other legitimate nodes from communicating. To deal with this problem, we devise a technique that runs in each node along with the aforementioned security mechanism to identify malicious nodes. To trace the suspicious node, each forwarding node monitors the traffic loads of its descendant nodes for a period of time and calculates the probability of a node being suspicious. Through analysis and simulations, we show the proposed scheme to be energy-conserving. A preliminary version of this paper can be found in [3].

The rest of the paper is organized as follows: In Section 2, we briefly explain related works. Network model and assumptions are outlined in Section 3. In Section 4, we present our security mechanisms in detail. The performance is evaluated analytically and through simulations to justify efficiency and practicability in Sections 5 and 6, respectively. Discussion and further issues are presented in Section 7. Section 8 concludes this paper.

2 Related works

Over the past few years, exhaustive research has been conducted on energy-conserving routing protocols for WSNs. In [4], the authors proposed a power-saving protocol that included the quantification of the trade-off between power conservation and quality of surveillance, the development of an efficient sleep–awake protocol, and the evaluation of soft deployment techniques. The sleep–awake protocol [4] provides better-quality surveillance while reducing power consumption. In [5], the authors proposed a geographic probabilistic flow-based spreading (PFS) routing protocol to extend the network lifetime. PFS spreads incoming traffic to eligible next-hop neighbors according to a probability distribution. The values of this distribution are set so as to balance the traffic load reported from all possible next-hop neighbors, and only next-hop neighbors with enough residual energy are eligible to receive packets for forwarding. Wie et al. in [6] proposed an energy-efficient, medium-access control protocol by periodically keeping the sensor nodes in listen and sleep modes. Their periodic listen and sleep scheme reduces energy consumption by minimizing radio transceivers' idle time.

In [7], Wu et al. proposed a distributed scheduling mechanism called lightweight deployment-aware scheduling (LDAS). This work assumes that sensor nodes are not equipped with GPS or other devices to obtain location information. LDAS can achieve a specific level of partial sensing coverage in a statistical sense. In LDAS, nodes are assumed to be randomly and uniformly distributed over the coverage area, and the protocol does not require accurate location information. Nodes have asynchronous sleeping schedules to balance energy consumption. In [8], authors proposed ASCENT, which uses sensors' local measurements to automatically configure network topology in a high-density sensor network. The goal is to maintain a certain data delivery ratio while allowing redundant sensors to stay asleep in order to conserve energy. Achieving this goal requires configuring the network to the right level of connectivity; it cannot be too low to hamper data delivery, but it cannot be too high either since neighboring nodes might interfere with each other, leading to a high collision rate. The approach adopted by ASCENT is to let sensors measure their connectivity as well as their data loss rate and activate their neighbors based on these local measurements.

Bandyopadhyay et al. [9, 10] considered a simple strategy to select cluster heads—they are chosen randomly with a probability p. There are two kinds of cluster heads: volunteer cluster heads and forced cluster heads. Each sensor can become a volunteer cluster head with probability p. A volunteer cluster head advertises itself to the neighboring sensors, which then forward the advertisement within k hops. Any noncluster-head sensor that receives such advertisements joins the cluster of the closest cluster head. Any sensor not associated with a cluster within t units of time becomes a forced cluster head. In [11], an energy-efficient protocol, TEEN, was proposed for reactive networks. The authors made a formal classification of sensor networks based on their modes of functioning as proactive and reactive networks.

In [12], Karlof et al. thoroughly discussed the problem of secure data transmission for different routing protocols, and they concluded that few of the many sensor network routing protocols have been designed with security as a goal. They suggested the security goals required for routing in sensor networks. A secure routing was proposed in [13], called security-aware ad hoc routing, that incorporates security attributes as parameters into ad hoc route discovery. Their goal is to characterize and explicitly represent the trust values and trust relationships associated with ad hoc nodes and use these values to make routing decisions. A new security threat, defined as rushing attack, was introduced in [14], and the authors showed that it is possible to secure such an attack, and a general design that uses this component may secure any on-demand route discovery mechanism against the rushing attack.

Random pairwise key distributions were discussed in [15] and [16] to make the sensor networks resilient to security threats. Kulkarni et al. [2] analyzed the problem of assigning initial secrets to users in ad-hoc sensor networks to ensure authentication and privacy during communications and pointed out possible ways of sharing the secrets. Particularly, they proposed two (tree and complementary tree) probabilistic protocols that maintain $O(\log N)$ secrets, where N is the number of nodes in the network. They showed that the probability of a security compromise between two users (nodes) is inversely proportional to the number of secrets they maintain.

A security mechanism is developed in [3] to thwart unauthorized data flow in the WSN by applying the complementary tree-based distribution of the cryptographic keys to the sensor nodes. The illegitimate packet is defended by checking the MACs attached to the data packet. Each node generates MACs with different cryptographic keys (from different complementary trees) and appends them with the data packet. The security mechanism works well when an adversary does not have cryptographic keys from all the complementary trees. However, if all the keys are compromised by the adversary, the security mechanism fails to detect the unauthorized data packets. To overcome this limitation, along with the security mechanism [3], we develop another mechanism to identify the malicious node when the security checking fails due to the compromise of all the cryptographic keys. In this mechanism, each forwarding node monitors the traffic loads of its descendant nodes for a period of time and calculates the probability of a node being suspicious when the traffic load exceeds the desired average value. The goal of both mechanisms is to detect and prevent malicious or misleading packet flow in the network and, thereby, to save wasteful energy consumption.

3 Network model and assumptions

We consider a uniformly distributed WSN that consists of N sensor nodes with equal capabilities and one data collection center called base station (BS) or sink. We assume that every sensor has a unique identifier (ID) $\mathrm{id}_{\mathrm{SN}}$ such that $1 \leq \mathrm{id}_{\mathrm{SN}} \leq N$. Once deployed, each node is assumed to be static. The BS is typically equipped with sufficient computation and storage capabilities, and it might have workstation- or laptop-class processor, memory, and storage [12]. However, sensor nodes are usually battery-powered, and the limited capacity of these batteries substantially limits the network lifetime [4]. Therefore, relaying by intermediate nodes needs to be performed so that the data can ultimately reach the BS.

We assume that the BS/sink is under the direct control of the network owner [12], and therefore, it is assumed that the adversary does not have the capability to attack the BS/sink because the powerful BS can protect any kind of malicious effort well. Note that using the mobile sink may create security problems since an adversary may find significant interest in compromising the mobile sink to easily bring down, or even take over, the sensor network. In such cases, security mechanisms that can tolerate mobile sink compromises are essential. We do not consider the use of a mobile sink in our work. However, our assumption on the network is that the attacker may know the basic approaches of the security mechanism and be able to compromise through radio communication channels. If the sensor node is compromised, all the information it holds will also be compromised (and, thus, the attacker knows all the cryptographic keys). Once the secret keys are known, a node can be used to generate/propagate sensed data that are illegitimate. Additionally, it may launch various other attacks, such as simply generating packets to congest the network (a selfish node may choose to wait for a smaller backoff interval, thereby increasing its chances of accessing the channel and, hence, reducing the throughput share received by other legitimate nodes) and recording and replaying older data packets, thereby consuming the network's overall energy.

4 Security scheme

In this section, we present our proposed security scheme in detail.

4.1 Complementary tree-based key predistribution

In this section, we describe the probabilistic protocol, the complementary tree protocol, for assigning the initial secret keys to the sensor nodes. We first describe the single complementary tree-based key distribution protocol and then describe the multiple complementary tree-based protocol. We organize the cryptographic keys in the tree of degree d, as shown in Fig. 1. In this paper, we use $d = 3$.

All nodes in the tree, except the root, are associated with a secret key. Each leaf of the tree is associated with a sensor node. Note that a leaf is associated with a sensor as well as a key, as shown in Fig. 1. The key distribution protocol is as follows: For each level (except level 1), the node gets keys associated with the siblings of its ancestors (including itself). Therefore, node s_1 gets keys k_2, k_3 (level 2), k_5, k_6 (level 3), k_{14}, and k_{15} (level 4). A node does not get the keys associated with its ancestors.

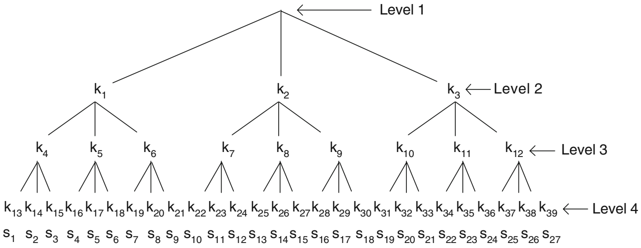
Level 1
k1
k2
k3
Level 2
k4
k5
k6
k7
k8
k9
k10
k11
k12
Level 3
k13 k14 k15 k16 k17 k18 k19 k20 k21 k22 k23 k24 k25 k26 k27 k28 k29 k30 k31 k32 k33 k34 k35 k36 k37 k38 k39
Level 4
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 s14 s15 s16 s17 s18 s19 s20 s21 s22 s23 s24 s25 s26 s27

Fig. 1 Single complementary tree-based cryptographic key distribution to the sensor nodes

When two nodes, say j and k, want to communicate, they first identify their least common ancestor. Let z be the least common ancestor of j and k. Let x denote the child of z that is an ancestor of j. Likewise, let y denote the child of z that is an ancestor of k. Now, to communicate, j and k use the keys associated with all children of z except x and y. For example, if s_1 and s_2 want to communicate, they use the key k_{15}. If nodes s_1 and s_9 want to communicate, they will use the key k_5. If s_1 and s_{15} want to communicate, they will use the secret key k_3.

For a single complementary tree protocol, each node gets $(d-1)\log_d(N)$ keys, and the probability to compromise a node is $\frac{2}{d+1}$, where N is the number of sensors. From Fig. 1, there are 27 sensor nodes, each node gets six keys, and the probability of compromise is 1/2 [2]. There are a total of $\sum_{j=2}^{level} d^{j-1} = 39$ keys in a single complementary tree, where $d = 3$.

It is possible to reduce the probability of compromise in the complementary tree protocol even further, if we maintain multiple trees, where each tree includes all the sensor nodes and secret keys. More specifically, if we maintain t trees, where there is no correlation between nodes' locations in different trees and $t \ll N$, then the probability of security compromise, $p_{\text{compromise}}$, is $\left(\frac{2}{(d+1)}\right)^t$ and each node gets $t(d-1)\log_d(N)$ keys [2]. As an example, for eight trees with degree $d = 3$, the probability of compromise is $(\frac{1}{2})^8 = 0.0039$. The authors refer readers to [2] for more details; however, for the completeness of the paper, we provide the derivation of the probability of compromise in the Appendix. With the initial secret keys assigned, the sensors are deployed in the desired area, and it is assumed that the sensor network is deployed by a single party and all the sensors are static after they are deployed in the area of interest.

4.2 Source data generation and forwarding

When an event occurs, the node that senses the signal will prepare the sensed data as a message to be sent to the BS through intermediate forwarding nodes. The message format is in the form of $(\text{id}_{\text{SN}}, \text{nonce}, \text{msg})$ where, msg is the sensed data, id_{SN} is the ID of the sensor node, and nonce is a special marker (e.g., a time stamp or a counter) intended to limit or prevent the unauthorized replay or reproduction of a message. Prior to forwarding the message to its upstream neighbor (i.e., next hop node towards the BS), the source node randomly selects f number of keys from f different complementary trees from its key-chain and generates f MACs and attaches them with the message msg. Source node computes f MACs, using those selected f keys, msg, id_{SN} and nonce according to Eq. 1:

$$\text{MAC}_i = (k_i, \text{id}_{\text{SN}}||\text{nonce}||\text{msg}), \tag{1}$$

where $i = 1, 2, \ldots, f$ and || represents stream concatenation. The MAC (which is of fixed length) is attached to the input and serves to prove integrity and authenticity of the input [17]. A MAC is also known as a cryptographic checksum [18]. Then, the source node combines the message, nonce, and MACs along with the key indices as a packet and sends to the forwarding node according to Eq. 2:

$$\text{packet} = (\text{id}_{\text{SN}}, \text{nonce}, \text{msg}, k_{\text{ID}_1}, \ldots, k_{\text{ID}_f}, \text{MAC}_1, \ldots, \text{MAC}_f). \tag{2}$$

The packet is said to be legitimate and allowed to be forwarded if it contains id_{SN}, nonce, msg, f MACs, and f key indices. Furthermore, keys selected by the source node must be different (from different trees) so that the same cryptographic key is not used more than once to generate the MACs.

4.3 Illegitimate packet detection

To check the legitimacy of a data packet, each forwarding node verifies the correctness of the MACs attached to the packet. If a malicious (compromised) node has only one key, it can generate one correct MAC. Since there are f distinct MACs (and f distinct key indices) that must be present in a legitimate packet, the attacker needs to forge $f - 1$ key indices (i.e., needs to know valid keys) and corresponding MACs. This is a difficult task for a compromised node (attacker) as the predistribution of keys is in such a way where finding the exact key that is shared between any pair of sensor nodes is difficult, as described in Section 4.1. However, if all the keys are correctly chosen from the total key pool, any attacker node can use f of its keys to generate multiple MACs, which would have been indistinguishable from those generated by f keys in the source (sending) node. In this case, an intermediate forwarding node cannot detect the illegitimate packet.

At the time when the forwarding node receives the packet, it looks at the key indices and the number of MACs. If this value is less than f or one key index is used more than once, the packet is considered to be illegitimate and, thus, is discarded. If the node has any of the key indices in common, it calculates the MAC using its own key and compares the result with the received MAC attached in the packet. The packet is discarded in case the attached one differs from the reproduced one. If it matches exactly or this forwarding node does not have any of the f keys in common (since key sharing is probabilistic), the node passes the packet to the next hop towards the BS/sink. This process continues at each forwarding node until the BS receives the packet.

4.4 Illegitimate sensor node detection

As stated in Section 4.3, if all the keys are correctly chosen from the total key pool, any attacker node can use f of its keys to generate multiple MACs (and, thus, legitimate packets) that are indistinguishable from those generated by f keys in the source (sending) node. Therefore, an intermediate forwarding node cannot detect the illegitimate packet. In this case, the sensor node itself is illegitimate but can continuously generate valid packets. This becomes a serious problem as the network energy may deplete, resulting in reduced network lifetime. To mitigate this problem, we present a simple and distributed detection algorithm to identify this kind of node (i.e., the illegitimate nodes). To identify a node, each forwarding node runs the algorithm to calculate the probability of a node to be illegitimate based on the number of packets it receives from its descendant nodes. The load at a forwarding node is the total number of data packets that arrive per unit time from its descendants. For any descendant node, the total number of packets sent to the forwarding node includes the packets from its descendants (if any) and the number of data packets generated by itself. We consider that each forwarding node considers only the generated packets from its descendants to identify whether the node is malicious or not. In what follows, we describe our proposed illegitimate sensor node detection algorithm.

Let $p^i_F(T)$ be the probability that node i is illegitimate calculated by the forwarding node F in a sampling interval T. This probability is a local signal calculated by the forwarding node and, thus, the protocol is fully localized and is run in a distributed manner. Let us consider a forwarding node F and n neighbor nodes (descendants) that forward their packets to F, as shown in Fig. 2. Let $x_i(T)$ be the number of packets received by F from a descendant i at sampling interval T. Then, the average number of packets, $F_{\text{avg}}(T)$, received by F can be given by Eq. 3:

$$F_{\text{avg}}(T) = \frac{\sum_{i=1}^{n} x_i(T)}{n}. \tag{3}$$

Let D be the number of descendant nodes for which $F_i(T) < F_{\text{avg}}(T)$, where $F_i(T)$ is the number of packets received from node i at forwarding node F at sampling

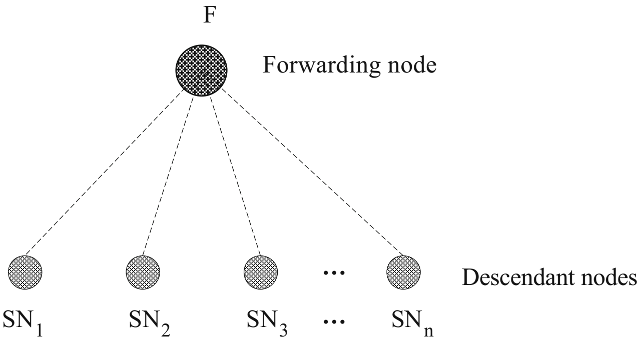

Fig. 2 Analytical model: *n* descendant nodes send data packet to the forwarding node *F*, which receives all the packets and redirects each packet towards the destination

interval T. So, $(n - D)$ is the number of descendant nodes for which $F_i(T) > F_{\mathrm{avg}}(T)$. The forwarding node F calculates $p_F^i(T)$ according to Eqs. 4 and 5 as follows:

$$p_F^i(T) = 0 \; \forall i, \text{ if } F_i(T) \leq F_{\mathrm{avg}}(T) \text{ and } i \in D, \tag{4}$$

$$p_F^i(T) = \frac{F_i(T) - F_{\mathrm{avg}}(T)}{F_i(T)}, \text{ if } F_i(T) > F_{\mathrm{avg}}(T) \text{ and } i \in n - D. \tag{5}$$

Equation 4 specifies that the probability of a node being suspicious remains zero when the expected number of packets received from individual nodes is less than or equal to the average calculated in Eq. 3. Equation 5 specifies the probability of a node being suspicious when the number of packets exceeds the desired average value. Based on this value, calculated in Eq. 5, each forwarding node F discards the packets from this particular suspicious node. Based on the output of the detection, the network operator may decide how to react to the attacker nodes. For example, the operator may revoke the attacker nodes and refresh the cryptographic material (i.e., rekeying).

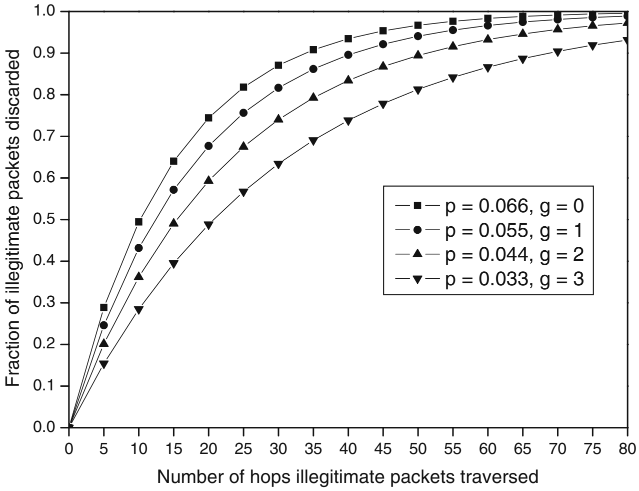

Fig. 3 Performance analysis. Fraction of illegitimate packets discarded by the intermediate forwarding nodes on the route towards the sink

To trace the suspicious node, the number of packets of the sending nodes is collected at the forwarding node for a period of time termed as the sampling interval. At the end of each interval, the detection mechanism is run at each forwarding node. It has been shown in [19] that the binary exponential backoff algorithm of IEEE 802.11 DCF is unfair in the short term. This would result in false positives if the sampling interval is short, even in the absence of malicious nodes. Therefore, the interval needs to be large enough to achieve long-term backoff fairness (we will specify the exact value of T in Section 6). In fact, taking into account the typical data rates, sampling interval may be short enough to prevent the illegitimate node from gaining large benefits before being detected. Detection efficiency depends on the typical network topology at hand and the number of malicious nodes and their behavior in the network. Through simulation, we will evaluate the performance of the proposed technique in terms of detection accuracy and energy conservation.

5 Performance analysis

In this section, we analyze two performance issues to evaluate the strength of our proposed security checking mechanism. First, we analyze the illegitimate packet detection efficiency, and secondly, we analyze the energy consumption with and without incorporating the security mechanism.

5.1 Illegitimate packet detection efficiency

Since the proposed security protocol deals with f MACs to send the packet to the upstream node (towards the destination), an adversary, that has compromised keys from f or all the complementary trees, can successfully generate legitimate packets. In this case, our proposed method cannot detect or discard such packets. Thus, the protocol is confined with its efficiency when an adversary has g number of compromised keys such that $0 \leq g < f$. So, if the attacker (node) wants to forward an illegitimate packet, he/she (it) has to compromise $f - g$ cryptographic keys and generate $f - g$ MACs. Now, if the attacker randomly chooses $f - g$ keys from distinct complementary trees, we compute the probability that a forwarding node has one of the $f - g$ keys and, thus, is able to detect an incorrect MAC and discard the packet. In this case, the probability that a forwarding sensor node has one of the $f-g$ keys, denoted by p, is given by Eq. 6

$$p=\frac{(f-g)}{t}\times\frac{t(d-1)\log_d(N)}{\sum_{j=2}^{\text{level}}d^{j-1}}. \tag{6}$$

The per-hop packet detection probability, denoted by $p_{\text{per-hop}}$, is given by Eq. 7:

$$p_{\text{per-hop}}=p\times(1-p_{\text{compromise}})\approx p. \tag{7}$$

As the probability of compromise, $p_{\text{compromise}}$, in Eq. 7 is very small (as described in Section 4.1), we consider this value to be negligible. Therefore, we take $p=p_{\text{per-hop}}$. Now, we can compute the expected fraction, p_H, of illegitimate packets being identified and discarded within H hops as (Eq. 8):

$$p_H=1-(1-p)^H. \tag{8}$$

As a natural corollary, we can compute the average number of hops, H_{avg}, an illegitimate packet passes the intermediate forwarding nodes according to Eq. 9

$$H_{\text{avg}}=\sum_{i=0}^{\infty}i\times(1-p)^{i-1}\times p=\frac{1}{p}. \tag{9}$$

The efficiency is shown in Fig. 3. The fraction of discarded packets increases as the number of hops grows. Here, we consider $N=729$ sensor nodes, a total of 1,092 cryptographic keys, the number of complementary trees, $t=8$, and each packet carries $f=6$ MACs. We have quantified the efficiency when an adversary (node) has keys from $g=0$, 1, 2, and 3 trees. Figure 3 shows that more than 64% of packets are discarded within 15 hops when an adversary has no compromised keys. Approximately 59% of illegitimate packets are discarded within 15 hops if the adversary has compromised keys from one complementary tree, and those packets pass only 18 hops on average. About 81% of packets are discarded within 50 hops when three MACs are incorrect (i.e., $g=3$), and they pass 30 hops on average. Clearly, the protocol performs well when the number of hops is large. This is because, the more an illegitimate packet travels, the greater the probability that this packet will be detected (and discarded) by one of the intermediate forwarding nodes is. For example, within 50 hops, almost 100% of packets are discarded if an attacker has only one valid MAC.

5.2 Energy conservation

In this section, we present the analysis to quantify the total energy consumption when the network operates without any security protocol and when the network incorporates the proposed security protocol. Total energy consumption in the sensor network results mainly from the energy consumed in transmission, reception, and computation. We ignore the energy consumption when a sensor keeps itself in active, sleep, or idle mode, since these will not make any difference in our analysis with and without the security protocol.

The proposed security protocol includes additional parameters of f key indices and f MACs. Let the byte length of the MAC and the key index be l_{MAC} and $l_{\text{key-index}}$, respectively. Let l_{msg} denote the length of the original message,(id_{SN}, nonce, msg). Then, the total length of the packet (with security parameters as in Eq. 2) becomes $f\times l_{\text{key-index}}+l_{\text{MAC}}+l_{\text{msg}}$.

Let H be the number of hops a packet flows from the source towards the destination, let $Q_{\text{illegitimate}}$ be the number of illegitimate packets, and let the number of legitimate packet be 1. All the packets traverse all the H hops when the security mechanism is not incorporated in the network. However, with the security mechanism enabled, an illegitimate packet will flow exactly H hops with the probability $(1-p)^{H-1}p$. Therefore, the amount of energy consumed for forwarding all the traffic without security, denoted by $E_{\text{no-sec}}$, and with security, denoted by E_{sec}, can be computed according to Eqs. 10 and 11

$$E_{\text{no-sec}}=l_{\text{msg}}(E_{\text{TX}}+E_{\text{RX}})(1+Q_{\text{illegitimate}})H, \tag{10}$$

$$\begin{aligned}E_{\text{sec}}=l_{\text{msg}}(E_{\text{TX}}+E_{\text{RX}})\big(1+f\times l_{\text{MAC}}/l_{\text{msg}}\\ +f\times l_{\text{key-index}}/l_{\text{msg}}\big)\\ \times\big(H+Q_{\text{illegitimate}}\times(1-(1-p)^H)/p\big),\end{aligned} \tag{11}$$

where, E_{TX} and E_{RX} denote the amounts of energy consumed in transmitting and receiving one byte, respectively. Additionally, the amount of energy consumption for the computation of security parameters, denoted by E_{comp}, can be approximated according to Eq. 12

$$\begin{aligned}E_{\text{comp}}=f\times E_{\text{MAC}}\big(1+H+Q_{\text{illegitimate}}\\ \times\big(1-(1-p)^H\big)/p\big),\end{aligned} \tag{12}$$

where, E_{MAC} is the amount of energy required for the MAC computation. So, the total amount of energy consumption with the security mechanism is $E_{\text{sec-total}}=E_{\text{sec}}+E_{\text{comp}}$.

Figure 4 shows the comparison of energy consumption with and without the security mechanism for different numbers of illegitimate packets, $Q_{\text{illegitimate}}$, when $H=80$, $l_{\text{msg}}=32$ bytes, $l_{\text{key-index}}=2$ bytes, $l_{\text{MAC}}=4$ bytes, and an attacker node has cryptographic keys from 1, 2, and 3 complementary trees. We take $E_{\text{TX}}=17$ and $E_{\text{RX}}=13$ μJ required [20] to transmit and

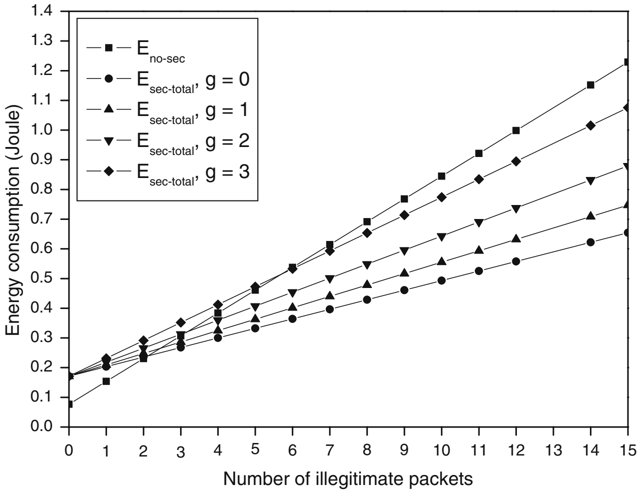

Fig. 4 Performance analysis. Comparison of energy consumption with and without the security mechanism as a function of the number of illegitimate packets

receive one byte of data, respectively, and $E_{\text{MAC}} = 16\ \mu\text{J}$ for MAC computation [18]. We observe (from Fig. 4) that $E_{\text{no-sec}}$ increases much faster than $E_{\text{sec-total}}$, and thus, the proposed security mechanism conserves the overall network energy. If $Q_{\text{illegitimate}}$ increases, the amount of energy that is saved gets higher and higher. For example, more than 35% of the energy is saved when 10 illegitimate packets are present in the network, and 40% energy can be saved with 15 illegitimate packets, where an attacker node has keys from only one tree. About 47% of the energy can be saved for 15 illegitimate packets when a node has no compromised key(s), as depicted in Fig. 4.

6 Performance evaluation

The effectiveness of the proposed security scheme is evaluated through simulations in NS-2 [21]. The simulation parameters are shown in Table 1. We have constructed a sink-rooted, tree-based routing, where identical sensors are uniformly distributed over the terrain. Sensor nodes and the sink are static after the deployment. The routing tree is constructed using Warshall's algorithm so that the sensed data could reach the sink with the shortest number of hops. It may be mentioned here that the choice of the downstream node(s) does not depend on any traditional parameters of sensor network routing (e.g., energy or delay).

Table 1 Simulation parameters

Parameter	Value
Deployment area	1,500 × 300 m
No. of sensor nodes, N	243
TX range of a sensor node	40 m
Channel capacity	19.2 kbps
Packet size	68 bytes
Sink location	[1,500, 150]
No. of complementary trees, t	8
No. of MAC, f	6
No. of forged MAC, $g (0 \leq g < f)$	1, 2, 3
Sampling period, T	13.5 s
Simulation time	135 s

We have implemented our security mechanisms to check the legitimacy of the data packets and nodes in the network. Our security module performs legitimacy checking on the data packets at each intermediate node (as described in Section 4.3), and each valid packet is forwarded along the shortest path towards the sink. Note that invalid packets are discarded if identified by the forwarding nodes. To evaluate the node detection efficiency, we simulate the proposed illegitimate node detection protocol described in Section 4.4. If a node is identified to be malicious by a forwarding node, all the packets are discarded by the forwarding node. First, we take a sensor node as a source that is 50 hops away from the sink. The source node randomly generates legitimate and illegitimate packets. With this setting, we quantify the strength of the proposed security checking mechanism when an attacker node is capable of generating $g = f - 1$ valid MACs. Second, we quantify the strength of the proposed node detection technique considering different percentages of nodes (out of 243 nodes) having all the cryptographic keys and being able to generate legitimate packets (i.e., $g = f$). The results are averaged over 10 simulation runs.

6.1 Simulation results

The analytical results of the proposed security mechanism are justified as can be observed from the simulated results depicted in Figs. 5 and 6. Figure 5 demonstrates the efficiency of discarding the illegitimate packets generated by the node as a function of the number of hops those packets traversed when $g = 1$, 2, and 3. When $g = 1$, more than 75% of the forged packets are dropped within 10 hops, and 68% and 57% of the packets are discarded with $g = 2$ and 3, respectively.

Figure 6 presents the comparison of energy consumptions with and without the proposed security mechanism as a function of the number of illegitimate packets. More than 50% of the energy is saved when an attacker has keys from one complementary tree and the number of forged packet is 10. When $g = 2$, about 44% of the energy is saved. Performance (i.e., energy conservation) is even better when the number of illegitimate packets is higher. For example, when $g = 1$,

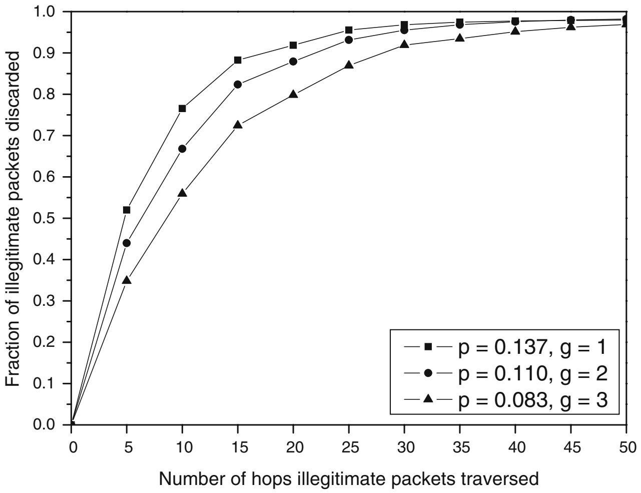

Fig. 5 Simulation results. Fraction of illegitimate packets discarded by the intermediate forwarding nodes on the route towards the sink

for 20 illegitimate packets, approximately 59% of the energy is saved. Hence, the proposed protocol performs better with a large number of malicious packets.

To evaluate the node detection efficiency, we simulate the proposed illegitimate node detection mechanism described in Section 4.4. We consider different percentages of nodes (out of 243 nodes) that have all the cryptographic keys and are able to generate legitimate packets (i.e., $g = f$). As stated earlier, sampling interval T is particularly important to increase the efficiency of the detection mechanism. To get the reference record of the traffic loads (i.e., number of packets of the descendant nodes) at the forwarding node, the value of T should be chosen carefully so that the usual variation of the short-term unfairness of the binary exponential backoff algorithm of IEEE 802.11 is taken into account. We derive the value of T to be 13.5 s from our simulation in NS-2 without the absence of any attacker node(s). The forwarding node computes $p^i_F(T)$ to identify the malicious node(s) when all the generated packets are indistinguishable by checking the attached MACs. Detection accuracy is expressed in terms of detection efficiency and false positives, and it is plotted in Fig. 7 as a function of $p^i_F(T)$ and percentage of the attacker nodes, respectively. Detection efficiency is calculated as the ratio of the number of attacker nodes detected correctly and the total number of attacker nodes in the network. False positive is calculated as the ratio of the number of legitimate nodes detected

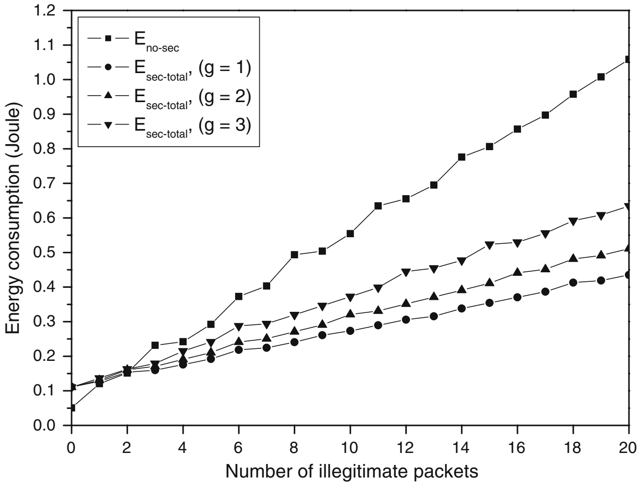

Fig. 6 Simulation results. Comparison of energy consumption with and without the security mechanism as a function of the number of illegitimate packets

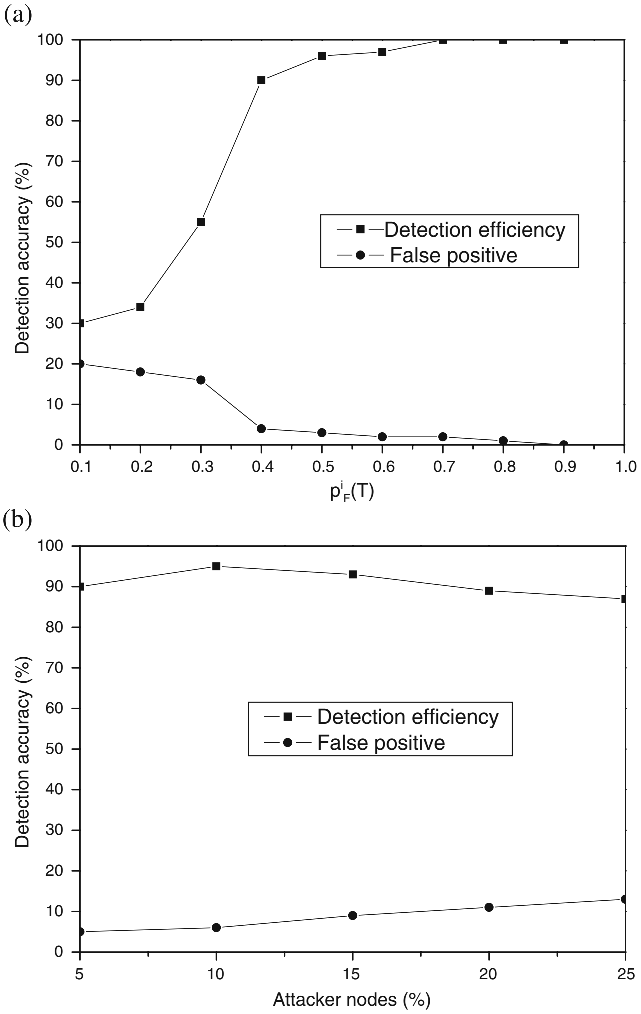

Fig. 7 Illegitimate node detection accuracy. **a** Detection efficiency and false positives with different $p^i_F(T)$. **b** Detection efficiency as a function of the percentage of attacker nodes (total 243 sensor nodes in the network)

as attacker nodes and the total number of nodes in the network.

Figure 7a shows that the detection efficiency gets better with low false positives when $p^i_F(T) \geq 0.4$. Figure 7b shows the efficiency and false positive rate with $p^i_F(T) = 0.4$, and the mechanism performs better when the percentage of malicious nodes is smaller. Figure 7a shows that around 90% detection efficiency is achieved with $p^i_F(T) = 0.4$ and almost 100% of the attacker nodes are detected with $p^i_F(T) \geq 0.5$, keeping the false positive rate bellow 4%. Figure 7b shows that approximately 90% detection efficiency is achieved with 9% false positive rate when 6.17% (15 out of 243 sensors) attacker nodes are present in the network.

Finally, we evaluate the energy conservation achieved through detecting the attacker node. We consider four source nodes that generate the data packets and send them to the forwarding node. One node (out of four) is considered to be the attacker having all the cryptographic keys and, thus, generates legitimate packets and sends packets continuously. The source nodes are placed 50 hops away from the sink. Figure 8 presents the comparison of the energy consumptions with and without the proposed node detection mechanism. We have normalized the number of legitimate packets to be one, and accordingly, the number of illegitimate packets is calculated, which we used for measuring the energy consumption, as shown in Fig. 8. As the attacker node has all the (valid) cryptographic keys, all the packets are transmitted towards the destination without being detected by checking the attached MACs (i.e., all the packets traverse all the hops). However, when the attacker node is detected by the forwarding node, all the packets generated by the attacker node will be discarded by the forwarding node. In Fig. 8, $E_{\text{not-detected}}$ denotes the amount of energy consumption when the node is not detected and E_{detected} denotes the amount of energy consumption when the node is detected by the forwarding node. Figure 8 shows that the energy consumption is much less with the proposed detection mechanism. Moreover, energy conservation gets higher with higher numbers of illegitimate packets since all the packets are discarded once the attacker node is detected by the forwarding node.

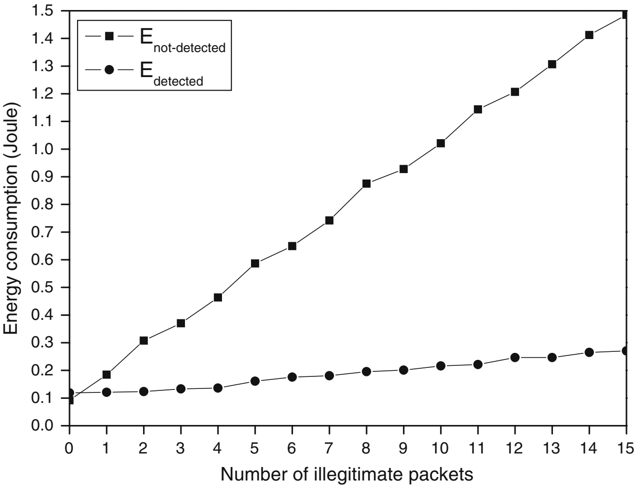

Fig. 8 Simulation results. Comparison of energy consumption with and without the proposed malicious node detection mechanism as a function of the number of illegitimate packets

7 Discussion and further issues

The security scheme is analyzed and simulated using the topology with a large number of hops (value of H is 80 in analysis and 50 in simulations) that may not seem practical for WSN applications. However, with the smaller value of H (e.g., 20 hops), it is evident from the results that the security scheme has good detection efficiency and, thus, saves overall network energy by discarding the illegitimate packets. So, the security mechanism can be applied for both small- and large-scale networks.

For the proposed security mechanism, each sensor node needs to store $t(d-1)\log_d(N)$ cryptographic keys to check the attached MACs to identify the illegitimate packets, where N is the number of sensors in the network. The computation overhead for each node includes the computation of f MACs. The communication overhead includes the transmission of f MACs, f key indices, and a nonce. For the proposed malicious node detection mechanism, each node keeps the records of the number of packets it receives from its n descendant (child) nodes during the sampling period T. So, the storage overhead is confined by the number of descendant nodes and the duration of the sampling period. Computation overhead includes the simple arithmetic calculation on the number of packets, average, and the probability $p^i_F(T)$. Thus, the implementation overhead is lightweight for the constrained sensor nodes.

The malicious node detection mechanism is proposed to deal with the situation when all the cryptographic keys of a node are compromised. An alternative technique may be developed with the assumption that each node is capable of using multiple distinct communication channels. In fact, with radio capabilities of MicaZ motes as specified in the 802.15.4 standard [22], nodes can communicate on multiple frequencies. Communication during the normal operation of the network is done on a single common channel, and the multichannel capability of the network is utilized only when a node suspects that its neighbor node is compromised. For example, when a node receives a large number of packets from one or more of its neighbor nodes and all the packets are legitimate (i.e., the receiving node cannot detect whether packets are illegitimate since the neighbor node(s) is (are) compromised), the receiving node may suspect that this is unusual behavior. The receiving node switches channels to communicate with other nodes and sends an alarm message to the BS so that the compromised node can be revoked. Latin square matrices [23] may be used to design such a switching schedule.

Finally, we believe that, in addition to the usual security concerns, it is also necessary to address selfish behavior (e.g., the protection of undesired data flow) that requires more attention and a more systematic approach. Design of energy-efficient techniques should consider sensors' capabilities, network structure, and deployment strategy. For example, the techniques developed in this paper mainly focus on how to save the overall network energy. A complementary mechanism can be designed to balance the energy consumption to maximize the network lifetime.

8 Conclusions

In this paper, we have developed two simple but efficient techniques to thwart unauthorized data flow in WSNs. A security protocol is devised to check the legitimacy of the packets (by the intermediate forwarding nodes), and a detection technique is devised to identify the malicious node (by the forwarding node) when the security checking fails due to compromise of all the cryptographic keys. The key features of both techniques are that they are: (1) simple and easy to integrate in the sensor node without interfering with its normal functioning (this is achieved by means of passive approaches based on legitimacy check and traffic monitoring) and (2) fully distributed and compatible with existing networks without requiring any modification of the standard communication protocols. Our analysis and simulations show the efficiency of the proposed techniques both in legitimacy check and energy conservation. We believe that our simple and distributed approaches that leverage on the sensor node characteristics can be effective in addressing security challenges for WSNs. Finally, discussions are made with potential research directions for further improvements.

Acknowledgements This research was supported by the MKE under the ITRC support program supervised by the IITA(IITA-2008-(C1090-0801-0016)). Dr. Choong Seon Hong is the corresponding author.

Appendix

Theorem *In the multiple complementary tree-based key distribution with t trees, where there is no correlation between nodes' locations in different trees and $t \ll N$, the probability of security compromise, $p_{\text{compromise}}$, is $\left(\frac{2}{(d+1)}\right)^t$.*

Proof Consider the single complementary tree-based key distribution in Fig. 1. Let l be the intruder that can observe the communication between sensor j and k. We want to identify the probability that l is aware of the secret(s) used by j and k. Now, consider different cases based on the shared secrets that j and k use during communication. Since no secrets are associated with the root, first consider the case where j and k use the secret(s) at level 2. Such a situation occurs if k is not a descendant of the level-2 ancestor of j. Thus, the probability of this case is $\frac{(d-1)}{d}$. Additionally, the probability that l is aware of all the secrets is $d/2$; l knows all the secrets used by j and k if and only if l is a descendant of the level-2 ancestor of j or l is a descendant of the level-2 ancestor of k. Next, we consider the probability that j and k use the secret at level 3 in the tree. Such a situation arises if k is a descendent of the level-2 ancestor of j and k is not a descendent of the level-3 ancestor of j. Thus, the probability of this case is $\frac{1}{d} \times \frac{(d-1)}{2}$. Moreover, l is aware of the shared secret(s) between j and k if and only if l is a descendant of the level-3 ancestor of j or l is a descendant of the level-3 ancestor of k. Thus, the probability of this case is $\frac{2}{d} \times \frac{1}{d}$. Continuing this way, the probability, $p_{\text{compromise}}$, that l is aware of the secret(s) used by j and k in a single complementary tree is

$$\begin{aligned} p_{\text{compromise}} &= \frac{(d-1)}{d}\frac{2}{d}\left(\sum_{j=0}^{h}(1/d)^{2j}\right) \\ &< \frac{(d-1)}{d}\frac{2}{d}\left(\sum_{j=0}^{\infty}(1/d)^{2j}\right) \\ &= \frac{(d-1)}{d}\frac{2}{d}\frac{1}{(1-1/d^2)} \\ &= \frac{2}{d+1}. \end{aligned}$$

With t complementary trees, $p_{\text{compromise}} = \left(\frac{2}{(d+1)}\right)^t$. □

References

1. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. Commun Mag IEEE 40(8):102–114. doi:10.1109/MCOM.2002.1024422
2. Kulkarni SS, Gouda MG, Arora A (2006) Secret instantiation in ad-hoc networks. Comput Commun 29(2):200–215
3. Hamid MA, Rahman M, Hong CS (2006) Energy conserving security mechanism for wireless sensor network. In: ICCSA (2), Glasgow, 8–11 May 2006, pp 866–875
4. Gui C, Mohapatra P (2004) Power conservation and quality of surveillance in target tracking sensor networks. In: MobiCom '04: Proceedings of the 10th annual international conference on mobile computing and networking. ACM, New York, pp 129–143. doi:http://doi.acm.org/10.1145/1023720.1023734
5. Wang N, Chang CH (2007) Performance evaluation of geographic probabilistic flow-based spreading routing in wireless sensor networks. In: PE-WASUN '07: Proceedings of the 4th ACM workshop on performance evaluation of wireless ad hoc, sensor,and ubiquitous networks. ACM, New York, pp 32–38. doi:http://doi.acm.org/10.1145/1298197.1298204
6. Ye W, Heidemann J, Estrin D (2002) An energy-efficient mac protocol for wireless sensor networks. INFOCOM 2002. In: Twenty-first annual joint conference of the IEEE computer and communications societies, vol 3. IEEE, Piscataway, pp 1567–1576. doi:10.1109/INFCOM.2002.1019408
7. Wu K, Gao Y, Li F, Xiao Y (2005) Lightweight deployment-aware scheduling for wireless sensor networks. Mob Netw Appl 10(6):837–852. doi:http://dx.doi.org/10.1007/s11036-005-4442-8
8. Cerpa A, Estrin D (2002) ASCENT: Adaptive self-configuring sensor networks topologies. In: INFOCOM 2002. Twenty-First annual joint conference of the IEEE computer and communications societies, vol 3. IEEE, Piscataway, pp 1278–1287. doi:10.1109/INFCOM.2002.1019378
9. Bandyopadhyay S, Coyle E (2003) An energy efficient hierarchical clustering algorithm for wireless sensor networks. INFOCOM 2003. Twenty-second annual joint conference of the IEEE computer and communications societies. IEEE 3:1713–1723
10. Bandyopadhyay S, Coyle EJ (2004) Minimizing communication costs in hierarchically-clustered networks of wireless sensors. Comput Netw 44(1):1–16
11. Manjeshwar A, Agrawal D (2001) TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. In: Parallel and distributed processing symposium. Proceedings 15th international, San Francisco, 23–27 April 2001, pp 2009–2015
12. Karlof C, Wagner D (2003) Secure routing in wireless sensor networks: attacks and countermeasures. Ad Hoc Netw 1(2–3):293–315
13. Yi S, Naldurg P, Kravets R (2001) Security-aware ad hoc routing for wireless networks. In: ACM international symposium on mobile ad hoc networking and computing. ACM, New York, pp 299–302
14. Hu YC, Perrig A, Johnson DB (2003) Rushing attacks and defense in wireless ad hoc network routing protocols. In: WiSe '03: proceedings of the 2nd ACM workshop on wireless security. ACM, New York, pp 30–40. doi:http://doi.acm.org/10.1145/941311.941317
15. Chan H, Perrig A, Song D (2003) Random key predistribution schemes for sensor networks. In: IEEE symposium on security and privacy, Berkeley, 11–14 May 2003, pp 197–213
16. Du W, Deng J, Han YS, Varshney PK, Katz J, Khalili A (2003) A pairwise key pre-distribution scheme for wireless sensor networks. In: ACM conference on computer and communications security (CCS). ACM, New York, pp 42–51
17. Menezes AJ, Vanstone SA, Oorschot PCV (1996) Handbook of applied cryptography. CRC, Boca Raton
18. Law YW, Doumen J, Hartel P (2006) Survey and benchmark of block ciphers for wireless sensor networks. ACM Trans Sen Netw 2(1):65–93. doi:http://doi.acm.org/10.1145/1138127.1138130
19. Barrett CL, Marathe MV, Engelhart DC, Sivasubramaniam A (2002) Analyzing the short-term fairness of ieee 802.11 in wireless multi-hop radio networks. In: MASCOTS '02: proceedings of the 10th IEEE international symposium on modeling, analysis, and simulation of computer and telecommunications systems (MASCOTS'02). IEEE Computer Society, Washington, DC, p 137
20. C. T. Inc. (2005) MPR400/410/420 mica2 mote. Datasheet 2005
21. The Network Simulator - ns-2 (2003) http://www.isi.edu/nsnam/ns/index.html
22. The Zigbee Alliance (2008) http://www.zigbee.org/en
23. Denes J, Keedwell AD (1974) Latin squares and their applications. Academic, New York