Ann. Telecommun. (2008) 63:529–541
DOI 10.1007/s12243-008-0042-5

SERP: secure energy-efficient routing protocol for densely deployed wireless sensor networks

Al-Sakib Khan Pathan · Choong Seon Hong

Received: 26 December 2007 / Accepted: 20 June 2008 / Published online: 19 July 2008

Abstract In this paper, we present secure energy-efficient routing protocol (SERP) for densely deployed wireless sensor networks which aims to achieve robust security for transmitted sensor readings with an energy-efficient network backbone. When the sensors with limited energy budgets are deployed in hazardous environment, ensuring energy efficiency and security of the sensor readings becomes a crucial task. Here, we address how to deal with such a deployment scenario. Our protocol ensures secure transmission of data from the source sensors to the base station in a way that it can best utilize the available amount of energy in the network. We use one-way hash chain and pre-stored shared secret keys for ensuring data transmission security. In SERP, first, a sink rooted tree structure is created as the backbone of the network. This energy-efficient network structure is used for authenticated and encrypted data delivery from the source sensors to the base station. To introduce data freshness, SERP includes an optional key refreshment mechanism which could be applied depending on the application at hand. Our analysis and simulation results show that SERP provides a good level of confidentiality and authenticity of data that are transmitted from the sensors to the base station. It also helps for energy-efficient structuring of the network so that the maximum lifetime of the network could be achieved.

A.-S. K. Pathan · C. S. Hong (✉)
Networking Laboratory, Department of Computer Engineering,
Kyung Hee University,
1 Seocheon,
Giheung, Yongin 449-701, South Korea
e-mail: cshong@khu.ac.kr

A.-S. K. Pathan
e-mail: spathan@networking.khu.ac.kr

Keywords Energy · Distance · One-way hash chain · Shared secret key

1 Introduction

Wireless sensor networks (WSN) are emerging as both an important new tier in the information technology ecosystem and a rich domain of active research involving hardware and system design, networking, distributed algorithms, programming models, data management, security, and social factors [1, 2]. The basic idea of sensor network is to disperse tiny sensing devices over a specific target area. These devices are capable of sensing certain changes of incidents/parameters and of communicating with other devices. WSNs could be very useful in providing support for some specific purposes like target tracking, surveillance, environmental monitoring, etc. Today's sensors can monitor temperature, pressure, humidity, soil makeup, vehicular movement, noise levels, lighting conditions, the presence or absence of certain kinds of objects or substances, mechanical stress levels on attached objects, and other properties. As such types of networks are composed of resource-constrained tiny sensor nodes, many research works have tried to focus on efficient use of the available resources of the sensors. Energy is in fact one of the most critical factors that play a great role to define the duration of an active and operable network. Energy efficiency is often very crucial in these sorts of networks, as the power sources of the inexpensive sensors are (in most of the cases) not replaceable after deployment. If any intermediate node between any two communicating nodes runs out of battery power, the link between the end-nodes is eventually broken. Therefore, any protocol should ensure a competent way of utilizing the energies of the sensors so that a fair

connectivity of the network could be ensured throughout its operation time. Energy efficiency is also very necessary to maximize the lifetime of the network.

Security, on the other hand, is another critical issue especially for ensuring the legitimacy of transmitted readings from the sensors to the base station [3, 4]. It is anticipated that in most application domains, sensor networks constitute an information source that is a mission critical system component and thus require commensurate security protection. If an adversary can thwart the work of the network by perturbing the information produced, stopping production, or pilfering information, then the usefulness of sensor networks is drastically curtailed. Thus, it should be made sure that the messages from the sensors *in action* are authentic and reach the base station without any fabrication or modification. As a strong property of security, authenticity of the messages is often considered as the most crucial.

The task of securing wireless sensor networks is however complicated, considering the fact that the sensors are mass-produced anonymous devices with a severely limited energy budget and initially with no knowledge of their locations in the deployment environment (in general cases). The architectural aspect of wireless sensor network could make the employment of a security scheme a little bit easier, as the base stations or the centralized entities could be used extensively in this case. Nevertheless, the major challenge is induced by the constraint of resources of the tiny sensors. In many cases, sensors are expected to be deployed arbitrarily in the enemy territory (especially in military reconnaissance scenario) or over dangerous or hazardous areas. Therefore, even if the base station (or sink) resides in the friendly and safe area, the sensor nodes need to be protected from being compromised. At least, it should be made sure that the reports that reach the base station are authentic and are not corrupted on the way of transmission.

In this paper, we deal with the challenge of energy efficiency and secure routing in wireless sensor networks in a highly dense deployment scenario. We propose secure energy-efficient routing protocol (SERP) [5], which aims at minimizing the wasteful energy consumption by energy-efficient structuring of the network and then secure the data transmissions from the sensors to the base station using one-way hash chain and shared secret keys. SERP selects a minimum number of forwarding nodes in the network. It provides a good level of confidentiality and authenticity of the reports sent from the source sensors to the base station.

The major contributions of this work are:

1. Energy- and distance-based efficient structuring of the network which helps for maximizing the lifetime of the network.
2. Providing data transmission security in wireless sensor networks. Here, we have mainly focused on data authenticity and confidentiality during their transmissions from the source sensors to the base station. There is also an optional key refreshment mechanism in our scheme, which could be applied based on the application at hand to provide data freshness.
3. Detailed analysis and simulation results of our proposed protocol.

The rest of the paper is organized as follows: Section 2 states the related works, Section 3 presents our assumptions and preliminaries, and Section 4 describes our protocol in detail. Simulation results and analysis are presented in Section 5, and Section 6 concludes the paper delineating the achievements from this work with future research directions.

2 Related works and motivation

Çam et al. [6] propose an energy-efficient security protocol for wireless sensor networks by using symmetric key cryptography and their non-blocking orthogonal variable spreading factor (NOVSF) code-hopping technique. They consider a hierarchical architecture of the network where data are routed from sensor nodes to the base station through cluster heads. The basic idea of their protocol is to implement two algorithms in the sensor nodes and in the base station which the sensor nodes and the base station would follow at the time of data transmission and reception. To ensure better level of security, they introduced NOVSF technique which basically scrambles the data blocks using a multiplexer in the system while transmitting data from the sensor nodes. Their scheme is secure and energy-efficient considering the fact that it increases the level of security during data transmission using NOVSF technique without utilizing any additional power. However, this scrambling technique increases the complexity of tasks for the base station, as it has to aggregate and reorder the incoming data blocks correctly. To address the issue of energy-efficient data aggregation with secure data transmission, energy-efficient secure pattern based data aggregation (ESPDA) protocol [7] is proposed. In contrast to the conventional data aggregation protocols, ESPDA avoids the transmission of redundant data from the sensor nodes to the cluster head. To make the data transmission and aggregation more secure, cluster head is not required to decrypt or encrypt the data received from the sensor nodes. On the whole, though [6] is an energy-efficient secure protocol, it increases the processing burden of the base station, and to support the associated ESPDA scheme, it requires more energy which literally ruins the gains of the original scheme.

Ye et al. [8] propose a statistical en-route filtering (SEF) scheme to detect and drop false reports during the forwarding process. In their scheme, a report is forwarded only if it contains the message authentication codes (MACs) generated by multiple nodes by using keys from different partitions in a global key pool. According to their findings, SEF can drop up to 70% of bogus reports injected by a compromised node within five hops and reduce energy consumption by 65% or more in many cases.

Zhu et al. [9] propose the interleaved hop-by-hop authentication scheme that detects false reports through interleaved authentication. Their scheme guarantees that the base station can detect a false report when no more than t nodes are compromised, where t is a security threshold. In addition, their scheme guarantees that t colluding compromised sensors can deceive at most B non-compromised nodes to forward false data they inject, where B is $O(t^2)$ in the worst case. They also propose a variant of this scheme which guarantees $B=0$ and which works for a small t.

Motivated by [9], Lee and Cho [10] propose an enhanced interleaved authentication scheme called the key inheritance-based filtering that prevents forwarding of false reports. In their scheme, the keys of each node used in the message authentication consist of the node's own key and the keys inherited from its upstream nodes. Every authenticated report contains the combination of the message authentication codes generated by using the keys of the consecutive nodes in a path from the base station to a terminal node. Other than these works, [11–13] focus only on energy efficiency in wireless sensor network, and the works like [3, 4, 14] deal with the security measures for routing in WSN.

After analyzing all these works, we design our protocol in which we create a tree structure in the network based on the energy levels and distances (from the base station) of the sensor nodes. Along with the energy-efficient structuring of the network, we initialize an efficient security scheme down the paths of the tree to ensure secure data transmission in the network. Security is in fact a vast area of research, but our focus of this work is to address secure data transmission from the source sensors to the base station along with energy-efficient structuring and operation of the network. We develop our protocol in a way that false injection of data cannot deceive the base station or, more specifically, cannot reach the base station. We emphasize on the authenticity of sensor readings so that before transmitting each packet, the forwarding nodes can detect the irregularities with a minimum effort and thus drop unnecessary or flawed packets. By stopping the false packets to travel a long distance along the created paths in the network, our mechanism helps for greater energy efficiency, as the intermediate nodes are thus saved from extra transmissions. For employing the entire protocol, we develop it in a way that before starting its operation for secure data transmission, the network is formed in an energy-efficient way. Periodic restructuring of the network is proposed to keep a balance among the nodes to dissipate energies in nearly equal proportion. Our goal here is to achieve maximum lifetime of the network with secure data transmission from any source sensor to the base station.

3 Assumptions and preliminaries

3.1 Sensor deployment and network model

We consider a wireless sensor network with densely deployed sensing devices. The deployment could be made by aerial or vehicular scattering or by physical installation. We assume that initially, all the nodes and the base station in the network have same transmission range (say r). Like μTESLA [15], our protocol requires that the base station and nodes are loosely time-synchronized, and each node knows an upper bound on the maximum synchronization error. The base station has enough energy to support the network's operations for its full lifetime. The sensors deployed in the network have the computational, memory, communication, and power resources like the current generation of sensor nodes (e.g., MICA2 motes [16]). Once the sensors are deployed over the target area, they remain relatively static in their respective positions. That means the nodes do not move with respect to their neighbor. The transmissions of each node are isotropic (i.e., in all directions) so that each message sent is a local broadcast within the transmission range of the node. The link between any pair of nodes in the network is bidirectional, that is; if a node n_i gets a node n_j within its transmission range (i.e., one hop), n_j also gets n_i as its one-hop neighbor.

3.2 Energy model and observations

An accurate model for the energy consumption per bit at the physical layer is given by:

$$E = E_{elec}^{trans} + \beta d^{\alpha} + E_{elec}^{recv} \tag{1}$$

where, E_{elec}^{trans} is the distance-independent amount of energy consumed by the transmitter electronics (PLLs, VCOs, bias currents) and digital processing, E_{elec}^{recv} is the energy utilized by receiver electronics, while βd^{α} accounts for the radiated power necessary to transmit over a distance d between source and destination. As in [17, 18], we assume that:

$$E_{elec}^{trans} = E_{elec}^{recv} = E_{elec} \tag{2}$$

Therefore, overall energy consumption between source and destination within one hop can be calculated using,

$$E = 2.E_{elec} + \beta d^{\alpha} \quad (3)$$

Broadly speaking, hierarchical routing protocols use control packets for topology construction phase. For a particular node i, control packet transmission cost can be calculated by,

$$C_i^{ctrl}(r) = [L_{ctrl} \times \beta r^{\alpha} + (nbr_i(r) + 1) \times L_{ctrl} \times L_E] \frac{1}{T} \quad (4)$$

where, α is the path loss exponent ($2<\alpha<5$), β is a constant [J/bit m^2], r is the transmission range, L_{ctrl} is the length of control packet in bits, nbr_i is the average number of neighbors of node i for range r, L_{E} is the energy needed by the transceiver circuitry to transmit or receive a packet, and T is the time period between two consecutive restructuring of the network.

For a particular path p, data communication cost from source i to the base station can be represented as,

$$C_i^{data}(p) = \left[\sum_{i=1,j=2}^{N} \left(nfrd_p(d_i) + 1\right) \times L_{data} \times \beta d_{i,j}^{\alpha} + \left(nbr_p(d_i) + 1\right) \times L_{data} \right] \times L_E. \quad (5)$$

Here, N is the total number of nodes in the network, i, $j \in 0,1,2,\ldots,N$ is the node index, p is the path associated for data transmission from source i to sink, d_i is the transmission range set by node i, $d_{i,j}$ is the distance between the nodes i and j, nfrd_p (d_i) indicates number of forwarding nodes for a path p and range d, nfrd_p (d_i) indicates the number of neighboring nodes for a path p and range d, L_{data} is the length of data packets in bits, and, finally, α and β are the same as in the previous equation. Total communication cost for sending a data packet from source i is:

$$C_i^{total}(p) = \sum_{i=1}^{N} \left[C_i^{ctrl}(r)\right] + C_i^{data}(p). \quad (6)$$

The observations from the above equation are:

- Wasteful (due to idle listening, overhearing, etc.) energy consumption increases as the number of redundant forwarder increases.
- Wasteful energy consumption increases as the number of idle nodes increases.
- Energy consumption increases exponentially as the distance between nodes increases.
- Frequency of control packet transmission is proportional to the energy consumption.

To reduce energy consumption, the following things could be done:

- Reducing the number of forwarding nodes (not hampering the level of connectivity and the reliability of the network);
- Putting certain portion of the nodes in sleep mode to reduce idle mode energy consumption;
- Employing adaptive transmission range according to the distance from the forwarder node to save energy; and
- Fixing the network restructuring frequency to ensure balanced energy consumption.

3.3 Basic terms and definitions

We consider three states of the nodes in our protocol during its operation:

Non-forwarding In this state, the nodes keep their radio transceivers 'Off' but continue to sense the events in their sensing ranges using the sensing circuitry. Sensing of any event turns on the radio of a non-forwarding node.

Forwarding Both the transceiver and sensing circuits remain 'On' in this state.

Active During the tree construction and one-way hash chain (OHC) initialization phase (later described in Section 4.1), all nodes remain in the active state. In active state, both the sensing and radio circuitries of the sensors remain 'On'. Basically, there is no major difference between forwarding and active state. We term these two states to differentiate the two phases in our protocol. That is, while constructing the tree in the network, we say the status of the nodes as they are in active status. After the first phase is over, we consider the nodes selected as forwarders are in forwarding state.

Active state time Let v be a node and $N_1(v)$ be the number of one-hop neighbors of v for a particular transmission range r (r is same for all nodes in the network including the

sink). Let, T_{rtt} be the round trip time for data propagation between the longest distant pair within one-hop neighbors. Then, the active state time for node v is given by the equation,

$$T_{active} = T_{rtt} \times N_1(v).$$

In our protocol, within the time T_{active}, a node could be able to determine whether it should participate in the tree as a forwarding node or not.

One-way hash chain To ensure security for data transmissions from the sensors to the base station, we use pre-stored shared secret keys and one-way hash chain. A one-way hash chain [19] is a sequence of numbers generated by one-way function F that has the property that for a given x, it is easy to compute $y=F(x)$. However, given F and y, it is computationally infeasible to determine x, such that $x=F^{-1}(y)$. An OHC is a sequence of numbers K_n, $K_{n-1},\ldots,K_0$, such that, $\forall i$: $0\leq i<n$, $K_i=F(K_{i+1})$. To generate an OHC, first, a random number K_{r} is selected as the seed, and then F is applied successively on K_{r} to generate other numbers in the sequence. In the next section, we describe in detail how the shared secret keys are used with OHC in our protocol to provide data transmission security.

It should be noted here that in this paper, we have used the terms 'base station' and 'sink' interchangeably.

3.4 Security assumptions and threat model

The base station could not be compromised in any way. We assume that no node could be compromised by any adversary while creating the tree structure in the network (i.e., the first phase of our scheme). This particular assumption is necessary to protect the network from being wrongly structured or to prevent the inclusion of any rouge entity in the network. In this case, we are mainly assuming that compromising a node with physical capture is not possible. In addition, some other attacks like jamming could hamper proper relaying of the control messages. We assume that at least in the tree structuring process, any physical capture or jamming attack is not done by any adversary. In fact, such types of initial attacks (for example Hello Flood attack [3]) could be another topic for research. In this paper, our focus is to secure the data transmissions from the source sensors to the base station and addressing jamming or physical capture are beyond the scope of this paper.

Initially, each node is equally trusted by the base station. Each node in the network has a unique shared secret key with the base station. These keys are pre-stored into the sensors' memories so that after deployment, the sensors could use the keys to encrypt data while sending to the base station. The base station keeps an index of the ids of the sensors and the corresponding shared secret keys.

Due to the use of wireless communications, the nodes in the network are vulnerable to various kinds of attacks. We assume that an adversary could try to eavesdrop on all traffic, inject false packets, and replay older packets. If in any case a node is compromised, it could be a full compromise where all the information stored in that particular sensor are exposed to the adversary or could be a partial compromise, that is, partial information is exposed.

3.5 Design goals

Our main goal is to shape the network in a way that it could ensure the delivery of authenticated and confidential data to the base station from any source node, which must be aware of the limited energy budget of the network. The design goals of our protocol could be noted as: firstly, the organization of the network should be energy-efficient so that the maximum lifetime of the network could be ensured. Secondly, the base station or the nodes should have the capability to detect falsely injected reports. Thirdly, the protocol should be considerably resilient and robust so that any node failure would not greatly hamper the network's operations.

4 Secure energy-efficient routing protocol

We describe our protocol with two interrelated phases. The base station initiates the first phase which creates a backbone in the network and initializes the OHC number. The second phase is the network operation and secure data transmission phase. Here, we present both of these phases in detail.

4.1 Tree construction and OHC initialization phase

We consider distances and residual energies of the nodes to construct a sink rooted tree (SRT) in the network. At the time of the tree construction, all nodes keep their radio transceivers 'On' to verify whether it should remain active as a forwarding node or not. A timer parameter is defined to ensure each node's active participation in this process for a specific period of time. Each node is prioritized for transmission according to its residual energy and distance from the sink.

Now, according to our assumption, all the sensors and the base station have shared secret keys that are pre-stored before deployment of the network. Therefore, when the sensors are deployed in the target area randomly, each sensor contains a shared secret key with the base station

which could be used to provide confidentiality of the reports. However, to provide authenticity of the transmitted data, all the intermediate nodes between any source node and the base station must be initialized with the basic one-way hash chain number. Let us suppose the initial OHC number is $I_{\mathrm{OHC}}=\mathrm{HS}_0$.

To initiate the first phase of network structuring and OHC number initialization, the base station B generates a control packet containing HS_0, a MAC for the control packet using the key K_i along with some other parameters. Here, K_i is the number in the key chain corresponding to time slot t_i. The format of the control packet is:

$$bcm: B|sid|ren|dist|fid|\mathrm{HS}_0|\mathrm{MAC}_{K_i}(B|sid|ren|dist|fid|\mathrm{HS}_0)$$

where sid indicates the sender's id, ren is the remaining energy of the sender, dist is the calculated cumulative distance to reach the sink using forwarding node(s), and fid is the id of the upstream node (i.e., immediate parent or immediate forwarding node) selected by the current node for forwarding data towards the sink. The sink node initiates bcm with sender id B, and the values of sid, ren, dist, and fid as -1, as according to our assumption, the base station has unlimited energy compared to the energies of the sensors in the network, and, in this case, no forwarding node is needed to reach itself.

When the base station transmits bcm, at first, its one-hop neighbors get the message. Receiving the message, each node in the one-hop neighborhood of the base station first calculates its distance (i.e., dist) from the base station based on the received signal strength, stores the value of HS_0, and sets B as its forwarder node (the ultimate destination is the base station). Now, each of these nodes transmits the message again within its own one-hop neighborhood (i.e., local broadcast). In this case, the sid is set to its own id, ren is its own residual energy, and the MAC part is kept the same as the base station message, bcm.

To ensure prioritization of the transmission of control messages, each node waits for a threshold time before each further transmission. Waiting time of a node before further transmission is defined by the following equation,

$$T_{\mathrm{wait}} = \{D_{\mathrm{s}}/E_{\mathrm{r}}\} \times R \tag{7}$$

where, D_{s} is the cumulative distance between the sink and the node, E_{r} is the residual energy of the node, and R is a constant that is needed to normalize the value of T_{wait}. As with the course of time, the sensors lose their energy levels and the value of the ratio of distance and residual energy increases; we need to normalize this value. In our case, R is the ratio of node's initial energy and transmission range.

Each node receiving the control messages from one or more upstream neighboring nodes first calculates the distance of each sender based on the received signal strength, then calculates the cumulative distances up to the sink via different possible forwarders (i.e., the upstream senders), stores the id and residual energy information of each sender, and stores HS_0 from the message sent by the first sender. To choose its forwarder node, it compares the values of the distance and energy ratios ($D_{\mathrm{s}}/E_{\mathrm{r}}$) of the neighboring upstream nodes and chooses the node with the least value of the ratio as its forwarder node.

It then senses the channel, and if the channel is idle, it waits for T_{wait} time and then re-transmits the message containing its own status information and with its chosen forwarder node id as its fid. As the selected upstream node could also get the message (as the link between any two nodes is bidirectional), it sets itself as a forwarding node for this transmitting node. This process continues and, eventually, a tree structure is created in the network where each node has a forwarder node on the way to reach to the base station and possibly one or more downstream nodes that can send data to it destined to the base station. Here, the value of T_{wait} depends mainly on the values of E_{r} and D_{s}. In fact, these values are used to set the priority of the nodes to be selected as forwarding nodes.

To authenticate HS_0, B releases the key K_i in time slot t_{i+d}. Here, d is the delay parameter for the time slot which could be set depending on the application at hand. It indicates after how many time slots the key for time slot i should be released. On receiving this key, a node can verify the integrity and source authentication of HS_0. Thus, along each path, the initial OHC number is initialized. It is to be noted that bcm will not bring any attack against the network even if the nodes on the other side of the network do not receive K_i at t_{i+d}. Since, the messages that are MACed by K_i are supposed to be sent out at time slot t, an adversary cannot launch any attacks with K_i when it gets K_i at $t_{\mathrm{i+}d}$. Within the time T_{active}, a node which does not get any message from any of its neighbors that, it should be a forwarding node, sets itself as a non-forwarding node.

Let us illustrate the first phase of our protocol with an example. Figure 1 shows a portion of a network where there are eight nodes. The black filled circle is the base station and the white circles are the sensors. The bold italic numbers beside each node indicate the residual energies of the nodes and the number in between any pair of nodes indicates the distance between those two nodes. For simplicity, here, we assume that R is 1. The base station initiates the first phase. The one-hop neighbors of the base station in Fig. 1, nodes 1, 2, and 3, get the message first. All of these nodes calculate their T_{wait} and T_{active} values. Here, the T_{wait} values for the nodes 1, 2, and 3 are 0.05, 0.057, and 0.03, respectively. In this dense deployment scenario, let us suppose that nodes 2 and 3 are within the transmission ranges of each other, but as node 2 has lower

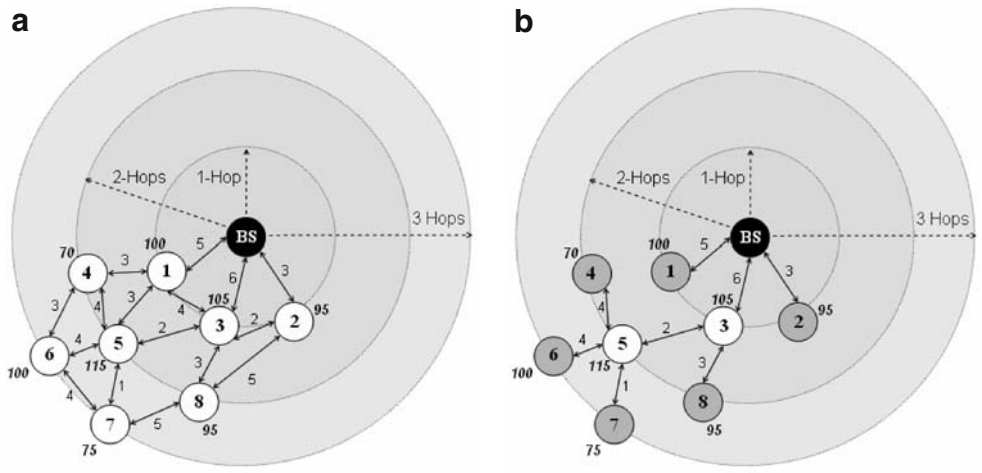

Fig. 1 A portion of an example network. **a** Before execution of the first phase. All the *white nodes* are in active status. **b** After execution of the first phase. The *gray nodes* are in non-forwarding status while the other two nodes are in forwarding status. It is evident from the figure that six nodes are in non-forwarding state which saves network-wide energy. In these figures, we have shown the N-hop (N=1, 2, 3…) neighbors of the sink on the circumference of the same circle regardless of their actual calculated distances from the sink

T_{wait} value, it transmits the message before node 3 does. Here, when node 2 transmits the message, nodes 3 and 8 get the message (as its one-hop neighbors), store the id, E_{r}, and D_{s} of 2 for future computations. In the case of node 1 and node 3, node 1 transmits before node 3 can do the transmission. The local broadcast of node 1 is heard by nodes 3, 4, and 5. Eventually, these nodes store the required information from this message. It should be noted here that as nodes 1, 2, and 3 get the control message directly from the base station, they set the base station as their fids.

After node 1 and node 2 have done the transmissions, let us take a sample case. Say, nodes 3 and 5 are neighbors of each other. Now, node 3 has T_{wait} value 0.057 and node 5 has T_{wait} value 0.069. Therefore, node 3 gets the chance of further transmission before node 5. Now node 8 is a neighbor of node 3. Therefore, node 3's transmission is heard by nodes 5 and 8. As a one-hop neighbor of the base station, node 3 does not need to select a forwarder, rather base station is its forwarder. When node 4's turn comes, it chooses node 5 as a forwarder node for itself. Node 5 knows this as it is a one-hop neighbor of node 4. This process continues, and the whole network is structured as a sink rooted tree where there are several paths from the base station to the leaf nodes. Along the path, the initial OHC number is also initialized. Figure 1b shows the resultant structure of the network after the first phase is executed. It could be noticed easily that there are several paths created in the tree like, 1→B, 4→5→3→B, 6→5→3→B, etc.

4.2 Network operation and secure data transmission phase

We construct the SRT based on the energy levels and distances of the nodes. After the tree is constructed within the network, all nodes are either in forwarding or non-forwarding states. Nodes with the non-forwarding state turn off their radio transceivers while keeping the sensing circuitry 'On'. On the other hand, forwarding nodes keep both radio and sensing circuitry 'On'. All nodes try to sense any change of parameters (like temperature, pressure, magnetism, etc. based on the duty assigned to the nodes) within their vicinities and upon detecting any event; the non-forwarding nodes turn their radios on and transmit data towards the base station via their selected forwarding nodes.

To send the data securely to the sink, each source node n_{s} maintains a unique one-way hash chain, HS: <HS_n, HS_{n-1},…,HS_1, HS_0>. When a source node, n_{s}, sends a report to the sink using the path created in the sink rooted tree (for example, a path is $n_{\text{s}}\rightarrow\ldots\rightarrow n_{m-1}\rightarrow n_m\rightarrow B$), it encrypts the packet with its shared secret key with the base station, includes its own id and an OHC sequence number from HS in the packet. It attaches HS_1 for the first packet, HS_2 for the second packet, and so on. To validate an OHC number, each intermediate node $n_1,\ldots,n_m$ maintains a verifier I_{ns} for each source node, n_{s}. Initially, I_{ns} for a particular source node is set to HS_0. When n_{s} sends the ith packet, it includes HS_{i} with the packet.

When any intermediate node n_k receives this packet, it verifies whether $I_{\text{ns}}=F(\text{HS}_{\text{i}})$ or not. If so, n_k validates the packet, it forwards it to the next intermediate node, and sets I_{ns} to HS_{i}. In general, n_k can choose to apply the verification test iteratively up to a fixed number w times, checking at each step whether $I_{\text{ns}}=F(F\ldots(F(\text{HS}_{\text{i}})))$. If the packet is not validated after the verification process has been performed w times, n_k simply drops the packet. By performing the verification process w times, up to a sequence of w packet losses can be tolerated, where the value of w depends on the average packet loss rate of the

network. Note that an intermediate node needs not to decrypt the packet; rather, it can check the authenticity of the packet before forwarding to its immediate forwarder. Figure 2 illustrates the one-way hash chain utilization procedure.

In Fig. 2a, the source node n_s sends the first packet to the base station with the OHC value HS_1. The content of the packet is encrypted with the secret key that it shares with the base station. Getting the packet, the base station performs the authenticity check by verifying the hash chain number and gets the report by decrypting it with the shared key for that particular source node. Figure 2b shows a scenario where the packet P_2 could not reach the base station for some reason. In spite of that, the OHC verification is not hampered as for the next packet, the third intermediate node performs the hash verification twice (Fig. 2c). Here, at the very first attempt, it cannot get the value of HS_1 in the verification process, but in the second iteration, it verifies it as a valid packet from the source n_s. In fact, in this case, the intermediate node can perform the hash number verification w times, where w is an application-dependent parameter. In Fig. 2d, an adversary tries to send a bogus packet with a false hash chain number and it is detected in the next upstream node. Eventually, such bogus packet fails to pass the authentication check and is dropped in the very next hop. This feature saves energy of the network as such falsely injected packets cannot travel through the network for more than one hop.

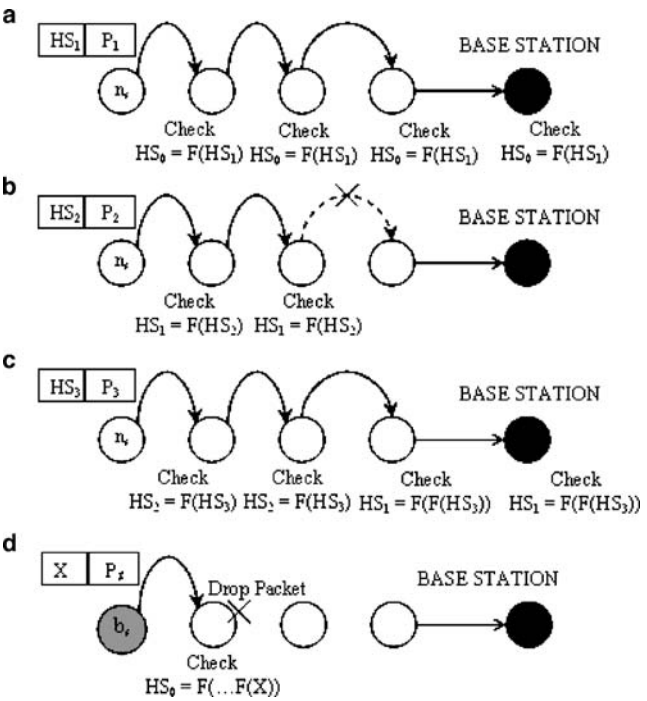

Fig. 2 **a** Authenticated packet delivery to the base station using the OHC numbers. **b** An example scenario where the packet could not reach the base station. **c** But it cannot affect the OHC verification technique. **d** A bogus packet with a false HS value is dropped by intermediate node

After the tree construction, at the time of data transmission, each node could dynamically set its transmission range according to the distance of the parent or immediate forwarding node. If the distance of the forwarding node is less than the initially used transmission range for tree construction, the node decreases the range by decreasing the transmission energy. This feature gives the flexibility in our protocol to dynamically set the transmission ranges, and thus, it helps for conserving network-wide energy.

The first phase is executed after every T time, where T is an application-dependent parameter. T depends on the event generation rate as well as on the load of the network. Each node participating in tree construction should have at least a certain level of energy. In our protocol, nodes with lower ratios of distance and residual energy are prioritized in the order of transmissions, which in fact ensures the choosing of the best forwarding node among the upstream neighbors of a node.

4.3 Optional key refreshment

To provide data freshness and to increase the level of security, our scheme has an optional key refreshment mechanism. In this case, the base station periodically broadcasts a new session key to the sensors in the network. The format for this message is:

$$B|K_s|MAC_{K_j}(B|K_s)$$

Where, K_j is the number in the key chain number corresponding to time slot t_j. To authenticate K_s, like the OHC initialization phase, B releases the key K_j in time slot t_{j+d}. On receiving this key, the nodes can verify the integrity and source authentication of K_s. Then, each node gets the new key by performing an exclusive OR (X-OR) operation with its old shared key. This method could also be utilized for refreshing the keys of a specific number of nodes. In that case, the base station could simply send the K_s to the specific node by encrypting it with its previous shared secret key. Upon receiving the new key, the node can perform the X-OR operation and could use the newly derived key for subsequent data transmissions.

Changing encryption keys time-to-time has an advantage, as it guarantees data freshness in the network. Moreover, it helps to maintain confidentiality of the transmitted data by preventing the use of the same secret key at all the times.

4.4 Repairing a broken path and OHC re-initialization

If in any case any node between the source node and the base station fails, it could make one or more paths useless.

Eventually, in such a case, all the downstream nodes along that particular path get disconnected from the base station. To repair such a broken path, we use the stored upstream knowledge of the sensors. We know that in the first phase, each downstream node stores the ids of the one-hop upstream senders of the control message. Therefore, this knowledge could be used for repairing the path quickly.

Let us illustrate it with an example. Say, in Fig. 1b node 5 is somehow damaged or failed to continue (Fig. 3a). Therefore, nodes 4, 6, and 7 get disconnected from the base station. This failure could first be detected by the one-hop neighbors of node 5 in the tree, i.e., nodes 4, 6, 7, and node 3. In the first phase, as node 4 got a message from node 1 which tried to become its forwarder, node 4 could use that knowledge to repair the path. Therefore, node 4 first does a local broadcast of an error message that it has lost its previous forwarder and sets node 1 as its forwarder. Accordingly, node 1 gets a forwarding status. If there were more senders who had sent control messages to node 4 at the time of tree construction, node 4 would have chosen the node with the least distance and energy ratio as recorded earlier. We know that in the first phase each node stores the information about its neighbors who try to become its forwarder. If node 4 is required to send any packet as a source node, it could simply send it using the OHC number in the sequence, HS_{k+1} which is next to its last used OHC number, HS_k. For node 1, node 4 is a new source, so it could save its HS value in I_4. The subsequent transmissions from node 4 are verified by node 1 based on this initial knowledge. There are other two stranded nodes in our example, node 6 and node 7. In the similar fashion, these nodes use their stored knowledge. The structure of the new path after broken path recovery is shown in Fig. 3b.

As we are considering a highly dense deployment scenario, we think that in most of the cases, a node might initially get two or more upstream senders who would try to be its forwarder. This procedure works fine as long as no more than w packets are lost on the way from any source node (after a path is broken due to a node failure). If within the time of repairing the path more than w packets are lost from a particular source, the OHC chain along that path breaks down. In fact, this is the worst case where all the downstream nodes along the path become invalid to the base station and their sent data are discarded on the way to reach the base station. To overcome this problem, the entire OHC initialization phase (the first phase of our protocol) could be made periodic (after certain interval, which is an application-dependent parameter). Determining the best possible time interval for re-initialization of the first phase is kept as our future work.

5 Simulation results and performance analysis

5.1 Simulation

To understand the performance of our proposed protocol, we simulated the network in NS-2 [20] with 50 to 300 nodes uniformly distributed in a 100×100-m^2 sensor field. The transmission range of each sensor node was set to 25 m. Each node was provided with 2 J of initial energy. Transmitter and receiver electronics were set to dissipate 50 nJ bit^{-1} m^{-2}. The data packet length was set to 2 KB. Sink or base station was located at (150, 150) coordinate. We varied the number of sources from one to seven, and data generation interval was randomly chosen. Initially, tree construction time was set to 10 s. As our protocol creates a hierarchical structure in the network, we compared our protocol with two other hierarchical energy-aware routing protocols LEACH [11] and EAD [12]. All the simulation parameters are shown in Table 1.

After the construction of the sink rooted tree, some of the nodes are selected as the forwarding nodes. The size of the set of forwarding nodes indicates at least how many

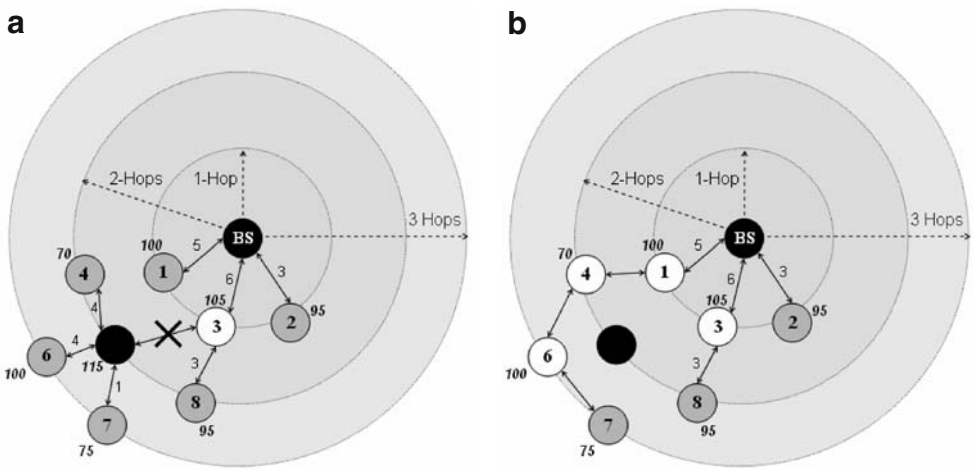

Fig. 3 **a** Node 5 failed. **b** Repairing a broken path. *White nodes* are in forwarding status and *gray nodes* are in non-forwarding status

Table 1 Simulation parameters

Parameters	Values
Simulation time	1,300 s
Simulation area	100×100 m^2
Total number of nodes	50–300
Initial energy	2 J
Transmit/receive electronics (L_E)	50 nJ bit^{-1} m^{-2}
Transmission power	5.85e−5 W
Receive signal threshold	3.152e−20 W
Sleep mode energy	0
Number of sources	1–7
Offered load	4–6 pps
Transmission range	25 m
Packet size	2,048 bytes

nodes are needed to stay awake for data transmission. A small set of forwarding nodes is desirable for minimizing the routing overhead. The smaller the size of the set of forwarders, the better the energy efficiency is for the network, as more nodes could be in the non-forwarding status. Figure 4a shows the percentage of forwarding nodes among the total number of nodes in LEACH, EAD, and our protocol. Now an interesting feature to note for the Fig. 4a is that as the number of nodes in the network grows, the percentage of cluster heads decreases slightly for LEACH because more nodes become associated with a single cluster head in the network. For the reason of dense deployment, relatively more nodes are covered by a cluster head. Thus, the percentage of cluster heads (forwarding nodes) becomes slightly lower than the suggested percentage of cluster heads as the number of nodes increases in the network.

Figure 4b shows the energy dissipation given a number of source nodes. Less energy dissipation eventually helps for increasing the lifetime of the network. The relative gain of our proposed scheme compared to LEACH and EAD increases with the increase of number of sources. More sources issue more data to be transmitted. In the case of LEACH, each transmission requires one hop to reach the cluster head and one hop to reach to the sink. In the case of EAD, multiple hops are required to reach to the sink. As wireless transmission power varies depending on distance, for the same packet size, LEACH requires much higher energy for transmission. EAD requires less energy than that of LEACH as it uses multiple hops (hence, less transmission range). As our algorithm uses adaptive transmission range, the amount of energy consumption is much less than LEACH and EAD considering the same packet size.

Figure 5a,b presents the number of *alive* nodes versus simulation time with 50 and 100 nodes. Our proposed scheme generates less number of forwarding nodes compared to EAD. As a result, the energy dissipation is much less than that of EAD as there are less nodes participating actively in the network operation phase. In addition, adaptive transmission range saves more energy for the same packet size. Single-hop transmission, the main drawback of LEACH, leads to huge energy consumption for data transmission. Our experimental results show that our algorithm achieves better lifetime compared to LEACH and EAD.

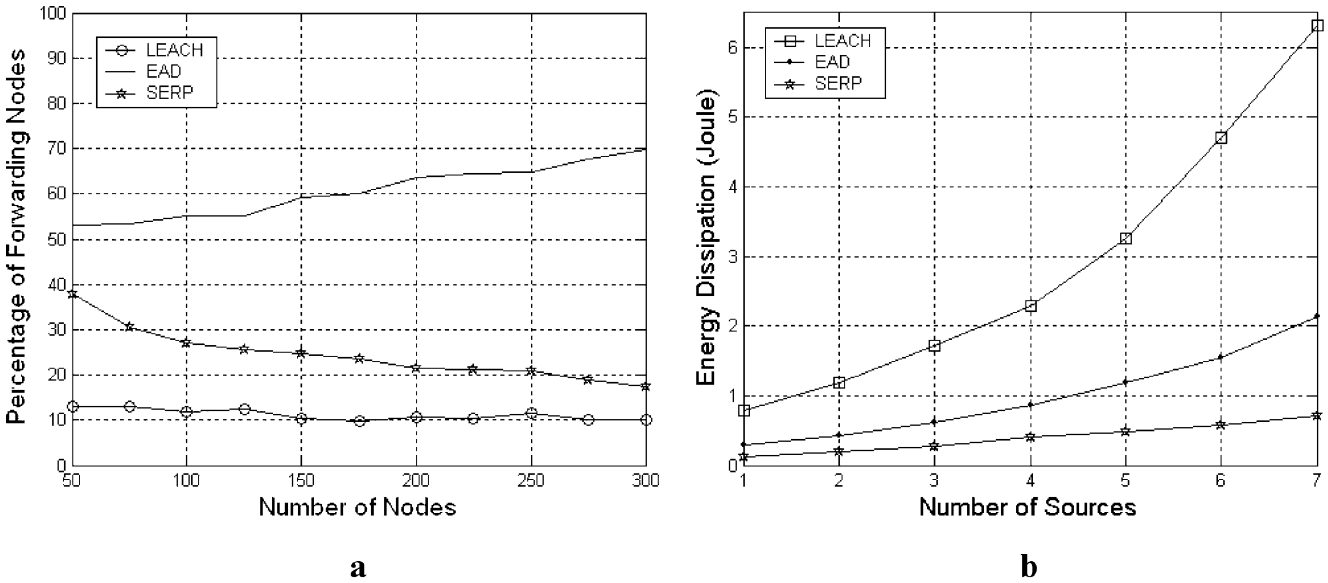

Fig. 4 **a** Percentage of forwarding nodes in total number of nodes in the network. **b** Energy dissipation for different number of sources in LEACH, EAD, and SERP

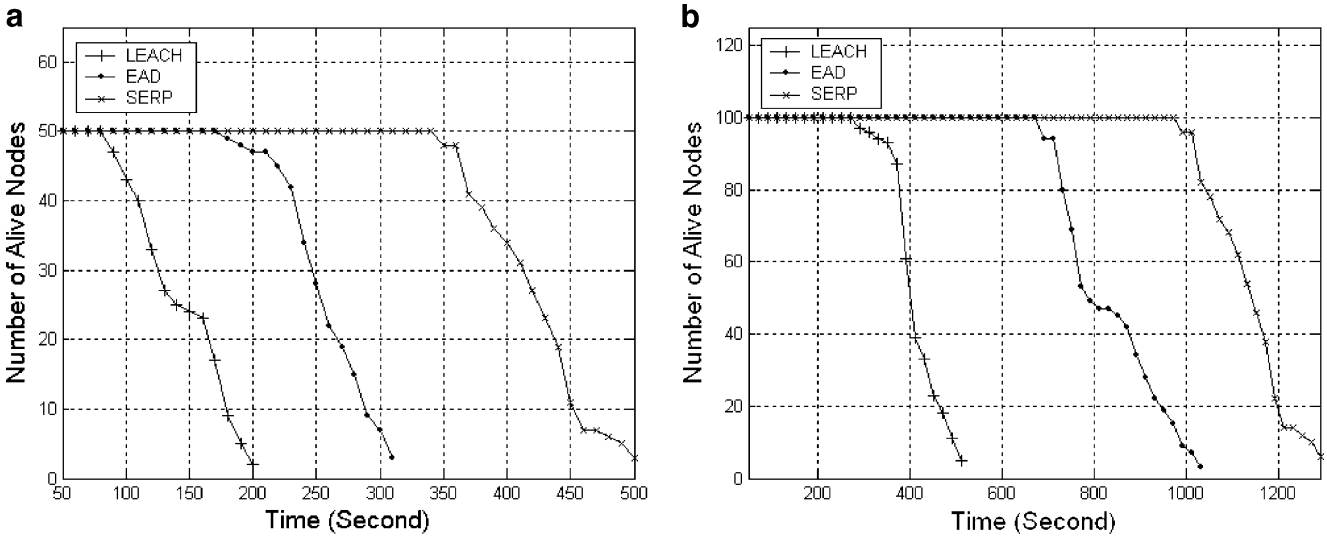

Fig. 5 Number of alive nodes versus time **a** 50 nodes, **b** 100 nodes

5.2 Storage requirement for one-way hash chain

The method of generating and storing a long OHC in a sensor node is a little difficult. Naive algorithms require either too much memory to store every OHC number or too much time to compute the next OHC number. None of these algorithms are practical on resource-constrained sensor nodes. Recently, some efficient OHC generation algorithms for resource-constrained platforms have been proposed [21–23]. Among these algorithms, the fractal graph traversal algorithm [21] could perform well on the traditional sensor nodes. This algorithm stores only some of the intermediate numbers, called pebbles, of an OHC and uses them to compute other numbers. If the size of an OHC is n (there are total n numbers in this OHC), the algorithm performs approximately $\frac{1}{2}\log_2 n$ one-way function operations to compute the next OHC number and requires a little more than $\log_2 n$ units of memory to save pebbles.

The length of an OHC that is needed for a source node is also an important factor. The typical length is between 2^{11} and 2^{22}. If the length of an OHC is 2^{22} and a node uses one OHC number per second, it will take more than a month to exhaust all numbers from this chain. Figure 6 shows the storage requirements for storing pebbles for different lengths of an OHC. This includes a skipjack-based one-way function and OHC generation based on [21]. We see that a node needs about 930 bytes to maintain an OHC of length 2^{22}. This includes 256 bytes lookup table for skipjack, which can be shared with other applications. Other than this, each node has to store only a few ids and neighbor information of its one-hop neighbors. Overall, the memory requirement for our scheme could be well afforded with today's sensor nodes.

5.3 Security analysis

We analyze the security of our scheme with respect to two design goals: the ability of the base station to detect a false report and the ability of the nodes en-route to filter or detect false reports.

5.3.1 Base station detection

In our scheme, whenever the base station receives a report from any sensor, it first checks the id of the sensor, checks the authenticity of the report by verifying the one-way hash chain number for that particular source, and looks for the corresponding shared secret key and decrypts the packet. The base station could not be compromised in any way.

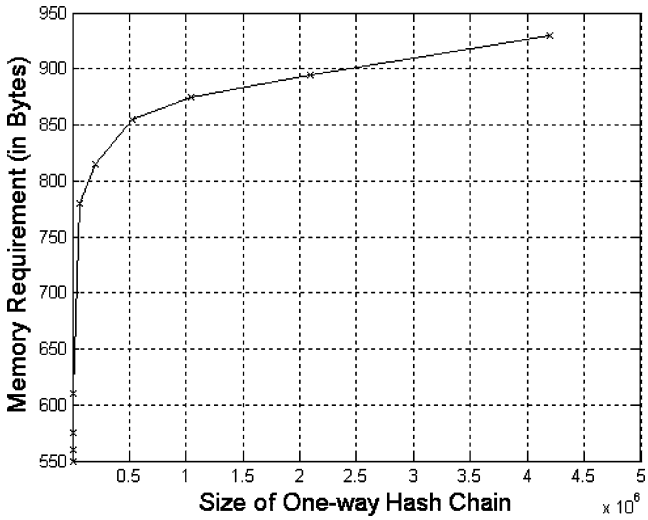

Fig. 6 Memory requirement for one-way hash chain generation

Therefore, it is in fact the final entity that could confirm the authenticity, confidentiality, and integrity of the transmitted reports. Our security scheme is designed in a way that any bogus report cannot reach the base station, rather would be detected and dropped by the intermediate nodes. However, if somehow a bogus packet is sent directly to the base station, it would certainly be discarded by it for the failure of authentication check. If in any application the optional key refreshment mechanism is employed, once the time slot of releasing the new session key is over, the base station first tries to decrypt the incoming packets from any particular source with the X-ORed new key for that node. In case it produces garbage result, the base station tries with the previous shared secret key with that node (the previous key could easily be obtained again by X-ORing the most recent session key with the newly computed key for that node). This case might happen when somehow some node cannot get the new session key released by the base station.

5.3.2 Detection by the intermediate nodes

In this section, we consider two types of attempts from the adversaries. One is the outsider attack where the adversary has not compromised any node in any path of the tree in the network and another attack is the insider attack in which case the adversary has compromised a node in a path.

Outsider attack In this case, as shown in Fig. 2d that if an outsider node generates a packet with fake OHC number, the authentication must be failed in the very next node in the path, and as a result, this packet would never be forwarded even to the node which is only two hops away from it. Simple verification of the OHC number prohibits the forwarding of such bogus packets.

Insider attack If a legitimate node along any path is compromised, the attacker could grab the OHC sequence and the shared secret key with the base station. However, it should be noticed that to use the OHC numbers successfully, the adversary should also know the last OHC number used by that particular node to send packet to the base station. If it gets the last used OHC number, then it could use this for sending false packets successfully. Otherwise, any arbitrary use of the OHC number from that source might not be forwarded by the next intermediate node because of authentication failure. Now, in case a node is fully compromised, that is if the adversary obtains all the required information, it actually gets the status of a legitimate node in the network. This fully compromised node could be used to generate false reports with valid authentication numbers. To prevent such type of malicious adversary, there are several factors that come into play to detect the abnormal behavior of the node. In our scheme, the base station considers a report legitimate if it is reported by at least δ number of source nodes in the network, where δ is an application-dependent parameter. Therefore, the different or modified reports from a single source cannot convince the base station about any event. In addition, the base station could notice the amount of packets generated by a particular source. These are basically a part of an intrusion detection system (IDS) implemented in the base station. The detailed description of the IDS is beyond the scope of this paper and will be reported in our future works.

The worst case scenario occurs if more than δ number of nodes in a particular region in the network are somehow compromised. This sort of collaborative and large scale attack is handled by the periodic restructuring of the whole network. Finding an optimal value of the time interval for periodic restructuring is kept as our future works.

In Fig. 7, we show the number of alive nodes versus simulation time considering the packet authenticity checking method and without checking. We considered two to four attackers in addition to the number of actual source nodes. The graph shows that if the detection method is absent, the nodes lose energy rapidly, which causes shorter network lifetime. The result is plotted for a total of 100 nodes in the network. In this experiment, four to 16 packets per second (pps) were generated by the attackers to drain energy of the nodes. When the packet authentication method is employed, the nodes can detect false packets and by dropping those other intermediate nodes are relieved from the burden of forwarding false reports.

As a whole, the efficiency of our protocol is increased with the number of false packets transmitted by the attackers. The more false packets are tried to be sent by the adversaries, the more gain we have, as those packets cannot travel a long distance towards the base station and

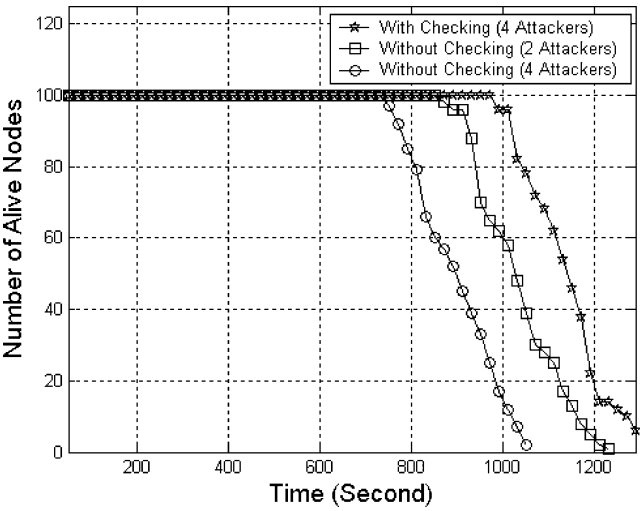

Fig. 7 Number of alive nodes vs time for two situations: without packet authentication and with packet authentication (two to four attackers generating bogus packets)

thus saves the network from consuming unnecessary energy by extra forwarding or transmission. This is in fact very helpful for the longer lifetime of the network in a heavy flooding attack where the attackers try to inject a huge number of false packets in the network data flow.

6 Conclusions and future works

Dense deployment of sensor nodes is often required to ensure better fidelity of the sensed reports sent from the sensor nodes. In this paper, we considered a dense deployment scenario of wireless sensor network and have proposed an energy-aware routing protocol which ensures data transmission security for the network. According to our design goals, our protocol structures the network in an energy-efficient way in which the base station or the intermediate nodes can detect the presence of falsely injected data and the network is robust enough to node failures. In this paper, in case of security, we have mainly considered the delivery of authenticated and encrypted data from the sensors to the base station. Other security schemes could be built upon our scheme to protect the network from other sorts of attacks. In fact, there is a lot of scope to extend the work further. As an example, it could be an interesting topic to find out an optimal value of the time interval for periodic restructuring of the network so that the maximum longevity of the network could be ensured. This particular problem could be left as an optimization problem, and use of exponential moving average [24] could be a solution. For the simplicity of our work, we have used a fixed time interval in our protocol. Furthermore, this value could be set based on the requirements or the application at hand.

Acknowledgments This research was supported by the MKE under the ITRC support program supervised by the IITA (IITA-2008-(C1090-0801-0016)). Dr. CS Hong is the corresponding author. We'd like to give special thanks to Md. Mamun-Or-Rashid for his generous help for this work. Also, special thanks to the reviewers for their valuable comments to improve this paper.

References

1. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) Wireless sensor net-works: a survey. Comput Networks 38:393–422
2. Dai S, Jing X, Li L (2005) Research and analysis on routing protocols for wireless sensor networks. Proceedings of the International Conference on Communications, Circuits and Systems, volume 1, pp 407–411 (27–30 May)
3. Karlof C, Wagner D (2003) Secure routing in wireless sensor networks: attacks and countermeasures. Elsevier's Ad Hoc Network Journal, Special Issue on Sensor Network Applications and Protocols, pp 293–315, (September)
4. Pathan A-SK, Lee H-W, Hong CS (2006) Security in wireless sensor networks: issues and challenges. Proceedings of the 8th IEEE ICACT 2006, Volume II, Phoenix Park, Korea, pp 1043–1048, (20–22 February)
5. Pathan A-SK, Hong CS (2007) A secure energy-efficient routing protocol for WSN. ISPA 2007, LNCS 4742, Springer, pp 407–418
6. Çam H, Özdemir S, Muthuavinashiappan D, Nair P (2003) Energy efficient security protocol for wireless sensor networks. IEEE 58th Vehicular Technology Conference, 2003, VTC 2003-Fall 2003, volume 5, pp 2981–2984, (6–9 Oct)
7. Çam H, Özdemir S, Nair P, Muthuavinashiappan D, Sanli HO (2006) Energy-efficient secure pattern based data aggregation for wireless sensor networks. Comput Commun 29(4):446–455
8. Ye F, Luo H, Lu S, Zhang L (2005) Statistical en-route filtering of injected false data in sensor networks. IEEE J Sel Area Commun 23(4):839–850 (April)
9. Zhu S, Setia S, Jajodia S, Ning P (2004) An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. Proceedings of S&P, pp 259–271
10. Lee HY, Cho TH (2006) Key inheritance-based false data filtering scheme in wireless sensor networks. Lecture notes in computer science, LNCS 4317, Springer, pp 116–127
11. Heinzelman WR, Chandrakasan A, Balakrishnan H (2000) Energy-efficient communication protocol for wireless microsensor networks. Proceedings of the 33rd Annual Hawaii Inter-national Conference on System Sciences (HICSS), pp 3005–3014
12. Azzedine B, Xiuzhen C, Joseph L (2003) Energy-aware data-centric routing in microsensor networks. Proceedings of the 8th MSWiM 03, San Diego, pp 42–49
13. Hyunh TT, Hong CS (2006) An energy delay efficient multi-hop routing scheme for wireless sensor networks. IEICE Trans Inf Syst E89-D(5):1654–1661 (May)
14. Yin C, Huang S, Su P, Gao C (2003) Secure routing for large-scale wireless sensor networks. In Proceedings of IEEE ICCT 2003, volume 2, pp 1282–1286, (9–11 April)
15. Perrig A, Szewczyk R, Tygar JD, Wen V, Culler DE (2002) SPINS: security protocols for sensor networks. Wirel Netw 8:521–534
16. Xbow Sensor Networks. Available at: http://www.xbow.com/
17. Hass ZJ (2001) Design methodologies for adaptive and multime-dia networks. IEEE Commun Mag 39(11):106–107 (November)
18. Heinzelman WB, Chandrakasan AP, Balakrishnan H (2002) An application-specific protocol architecture for wireless microsensor networks. IEEE Trans Wirel Commun 1(4):660–670 (October)
19. Lamport L (1979) Constructing digital signatures from one-way function. Technical report SRI-CSL-98, SRI International, October
20. The Network Simulator-ns-2. http://www.isi.edu/nsnam/ns/
21. Coppersmith D, Jakobsson M (2002) Almost optimal hash sequence traversal. 6th International Financial Cryptography 2002, Bermuda, (March)
22. Jakobsson M (2002) Fractal hash sequence representation and traversal. 2002 IEEE International Symposium on Information Theory, Switzerland (July)
23. Sella Y (2003) On the computation-storage trade-offs of hash chain traversal. The 7th International Financial Cryptography Conference, Guadeloupe, (January)
24. Ee CT, Bajcsy R (2004) Congestion control and fairness for many-to-one routing in sensor networks. Proceedings of ACM SenSys'04, 148-161