American Journal of Engineering and Applied Sciences

Original Research Paper

SFAD2H: Selective Forwarding or Dropping Attack Detection with 2-Hop Acknowledgment Support in Wireless Sensor Networks

Prathap Uppara, Kiran Krishnappa, Deepa Shenoy Punjalkatte and Venugopal Kuppanna Rajuk

Department of Computer Science and Engineering,
University Visvesvaraya College of Engineering, Bangalore University, India

Article history
Received: 22-09-2017
Revised: 30-10-2017
Accepted: 21-11-2017

Corresponding Author:
Prathap U.
Department of Computer Science and Engineering,
University Visvesvaraya College of Engineering,
Bangalore University, India
Email: prathap.u@gmail.com

Abstract: Security in wireless sensor networks is critical due to its way of open communication. In this study we have proposed a technique based on a sensor node having alternate path knowledge and 2-hop acknowledgement mechanism to detect adversary nodes which perform selective forwarding and dropping attacks. In selective forwarding attack nodes on the forwarding path refuses to transfer packets selectively. The proposed approach starts with network initialization where every node decides the list of parent nodes through which Sink can be reached with equal distance. Each node chooses a parent node among selected parents to forward the data and establishes pairwise keys with 2-hop parent nodes. During data forwarding, child forwards the packet to 1-hop distance parent, handles acknowledgement from 2-hop distance node and decides the number of packets forwarded and dropped based on successful and unsuccessful transactions. Every node sends a transaction report containing observations on the parent via alternate path to Sink at a particular interval of time called an evaluation period. Sink identifies the malicious node by comparing report received from each node with number of data packets received. Simulated the algorithm in NS-3 and performance analysis compared with other recently proposed approach. Simulation results show that proposed method detects the malicious nodes efficiently and early.

Keywords: WSN, Malicious Node, Selective Forwarding, Selective Dropping, 2-Hop Acknowledgment

Introduction

Wireless Sensor Networks

A Wireless Sensor Network (WSN) consists of spatially distributed autonomous devices having sensing, computing and communication capabilities. Sensor nodes cooperatively monitor physical or environmental conditions, such as temperature, pressure, sound, vibration, motion or pollutants. Wireless sensor networks are used in environmental conditions where information is difficult to access. Sensor node, also known as a 'mote', is a node in a wireless sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. Sensor network transmits the data from one node to another node in an adhoc way and finally to a base station where the data is stored, processed and displayed.

Security Attacks in Wireless Sensor Network

Sensor nodes are vulnerable to a wide range of attacks (Chan and Perrig, 2003; Butan *et al*., 2014). Attacker can listen to radio transmissions, modify the packet before forwarding, misroute the packet to unintended next hop node, inject false data in the channel, replay previously heard packets to drain the energy of other nodes as battery power is crucial in nodes. Attacker may deploy few malicious nodes with similar or better hardware capabilities or by 'turning' few legitimate nodes by capturing them and physically overwriting their memory. Sybil attack-attacker deployed nodes may also use the identities of the other

genuine nodes to frame other genuine nodes as malicious. In sinkhole attack (Chen *et al*., 2010) malicious node attracts the routing data by publishing the shortest path to Sink and drops most of the packets without forwarding further towards Sink or modifies the forwarded packets. Packet dropping, modification, misrouting are basic problems which have large impact on the information gathered by sensor nodes as network loses lot of important sensed data. Cryptography techniques alone are not sufficient to protect the data. Attacks such as colluding collision (Khalil and Bagchi, 2011), misrouting, sinkhole, wormhole (Bendjima and Feham, 2016), rushing attacks can be launched without the help of cryptography keys (Khalil, 2011).

Selective forwarding attack (Ren *et al*., 2016) is specialized case of Sinkhole attack where a malicious node selectively drops the packets. Sink node or the aggregating node aggregates the sensed information received from different sensors to make a meaningful data. Integrity of the sensed data cannot be trusted due to the selective dropping attack. Wireless medium is inherently not reliable as communication incurs data loss. With the Selective forwarding attack, it is a challenge to decide whether the data dropped by a node due to its malicious behavior or data loss due to unreliable wireless medium.

Introduction to SFAD2H Approach

In this study, we propose a technique to detect and bypass malicious nodes which perform selective forwarding attack. During network initialization, nodes in the network build parent child relationship and create a routing path to reach Sink node. Sink initiates propagation of the distance information to reach Sink node with its neighbor nodes. Similarly on receiving the distance information, each node increments the distance value by one hop to reach Sink node and propagates the distance information to next level nodes. Each node maintains a list of parent nodes through which Sink node can be reached with equal distance by the end of network initialization. At the end of network initialization, every node establishes a pairwise key with all the 2-hop distance grandparent nodes which can be reached through 1-hop distance parent nodes. Each node sends the details of the selected parents to Sink node. Sink forms a tree topology rooted with Sink node. Sink uses the topology for tracing the routing path and finding the source node.

The proposed approach assigns every node on the routing path, the responsibility of sending the packet to 2-hop distance node through 1-hop distance node on the routing path towards Sink. In Fig. 1, node *X* forwards the packet to next hop node *Y* and expects an acknowledgement from 2-hop distance node *Z* to confirm selective dropping attack by 1-hop distance node *Y*. *Z* does not reply acknowledgement unless it receives packet from *Y*. Node *X* maintains the count of successful and unsuccessful packet transmission from *Y* based on 2-hop acknowledgement received from *Z*. Node *X* switches to next parent to forward data towards Sink as soon as the dropping rate of the current parent node crosses the selective dropping threshold. Child node transmits the transaction report at equal time intervals through the alternative parent node. Sink detect the malicious nodes based on the received data packets and also received transaction reports from all nodes. Sink can generate report of the malicious node identities for network administrator to physically flush the memory of malicious nodes.

In order to detect the selective forwarding attack 'Adaptive and Channel-Aware Detection of Selective Forwarding Attacks in Wireless Sensor Networks (CRS-A) (Ren *et al*., 2016)' has been proposed recently in the literature. CRS-A evaluates the data forwarding behaviors of sensor nodes, as per the deviation of the monitored packet loss and the estimated normal loss. CRS-A theoretically derives the optimal threshold for forwarding evaluation, which is adaptive to the time-varied channel condition and the predicted attack probabilities of malicious nodes. Furthermore, an attack-tolerant data forwarding scheme is developed to improve the data delivery ratio of the network. We provide a simulated analysis comparing the CRS-A approach and our proposed approach. The rest of the paper is organized as follows, section II discusses about the related work, section III describes the network model and problem statement, section IV presents the solution and algorithm, section V provides the performance analysis and results and section VI concludes the work and discusses the future challenges.

Related Work

Multipath routing is very basic technique widely applied to minimize the impact of selective dropping attack on data delivery. The idea is either sending multiple copies of the same data through different paths to destination (Karlof and Wagner, 2003; Bhuse *et al*., 2005; Kefayati *et al*., 2006; Mavropodi *et al*., 2007; Pavithra and Reddy, 2015) or splitting the data into *N* shares and sending the *N* shares through different paths to destination (Shu *et al*., 2010; Liu *et al*., 2012; Li *et al*., 2014). Destination needs to collect and merge at least *M* out of N shares to make meaningful data. The selective dropping effect is mitigated even if *N-M* shares are dropped on the forwarding path.

Neighbor node observation or monitoring is another approach (Ye *et al*., 2004; Zhu *et al*., 2004; Yang *et al*., 2005; Prathap *et al*., 2015a; Lim and Huie, 2015; Gerrigagoitia *et al*., 2012; Ju *et al*., 2010) used to find the malicious activities such as packet modification and

dropping of the current forwarding node. In monitoring approach, observer nodes monitor the current sender and current receiver for the packet being transmitted. Observers observe for various malicious activities such as packet dropping, modification and etc. Monitoring methods require observer nodes to buffer the packets which are forwarded to next hop node and compare the packet forwarded by next hop node with its buffered packet to find out packet modifications. In specialized version of monitoring approach, there are designated anchor nodes whose job is to monitor the nodes responsible for data transfer and report the malicious activities to neighbor nodes.

In (Kaplantzis *et al*., 2007), a centralized intrusion detection scheme based on Support Vector Machines (SVMs) was proposed. Approach has used sliding windows for selective forwarding attacks. This approach only detects the attacks. It uses routing information local to the base station of the network and raises alarms based on the 2D feature vector (bandwidth, hop count). This scheme is unable to identify malicious nodes or find alternate paths for packet forwarding.

Paper (Deng *et al*., 2009) proposes an approach to secure the data transmission and detects a selective forwarding attack. Deng *et al*. (2009) used watermark technology to detect malicious nodes. Source path for forwarding the messages is identified with the help of trust values of the nodes on the path. The base station creates a *K* bits binary sequence as the watermark message. Base station compares the extracted watermark to the original watermark to detect a selective forwarding attack. Base station determines the packet loss with the help of watermark.

Stehlik *et al*. (2016) proposed two parameterized collaborative intrusion detection technique and optimize their parameters for given scenarios using extensive simulations and multi-objective evolutionary algorithms. But the approach depends on the specification of the network for configuring the optimization parameters used in the approach.

Cui and Yang (2014) provided an analytical model to estimate the wellness of a node's forwarding behavior. They borrowed the idea of the PageRank algorithm to estimate the most susceptible nodes to selective forwarding attacks in a network. Based on the analyses, they developed a novel reactive routing scheme that bypasses suspicious nodes. The approach suffers if network demands for early detection.

The scheme proposed in (Ren *et al*., 2016) is based on neighbor node observation. The optimal packet loss threshold due to selective forwarding attack is estimated over the inherent loss due to wireless channel by evaluating the channel forwarding behavior of the nodes. This approach suffers from early detection issue and optimal packet loss need to be recalculated based on varying channel conditions.

Energy consumption in both multipath routing and neighborhood monitoring is not affordable for sensor networks. In multipath routing, energy is consumed from nodes along multiple paths to Sink, to transmit same copy of data. In monitoring approach, many nodes observe each hop while a packet being forwarded and energy of all the observer nodes consumed.

Network Model and Problem Statement

Network Model and Initialization

We have considered wireless sensor network with one Sink node with all the sensor nodes are uniformly distributed. After deployment, network initialization and routing path building starts with Sink node (Prathap *et al*., 2015b). Sink node transmits the path distance information to 1-hop neighbors say node *Z* in Fig. 1. 1-hop neighbors increment the distance information and share with 2-hop neighbors and continues till the last hop node. In Fig. 1, node *Z* increment the distance count by 1 and share with node *Y*. Each node maintains a list of parent nodes which have equal and shortest distance to Sink node. Each node transmits the list of all the identified parents to Sink node. Sink establishes a routing tree rooted at Sink node based on the information received from each node. Each node chooses a different parent node as soon as the current parent's malicious behavior crosses the threshold.

Intermediate node prepares marker data containing node identity and adds to the packet before forwarding the packet to parent node. Marker data added by each node helps Sink to trace the nodes participated in forwarding the packet (Prathap *et al*., 2015b). All the nodes transmit the sensed data towards Sink for processing. The typical packet format looks like < id_Z, id_Y, id_X, id_N, $(S,D)_{K_S}$ > for the data generated from node S, where id_Z, id_Y, id_X and id_N are node identities on the forwarding path added by respective node as a path marker, *S* is identity of the source node, *D* is the data generated by the node *S* and K_S is the pairwise key shared between Sink and source node for data encryption.

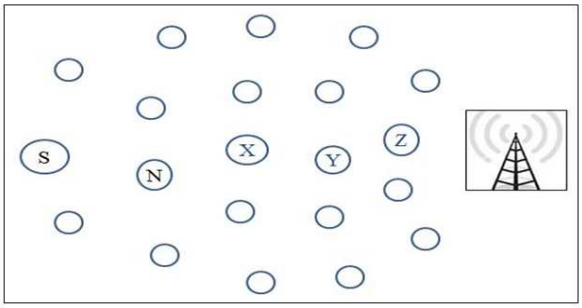

Fig. 1: Initial deployment of nodes

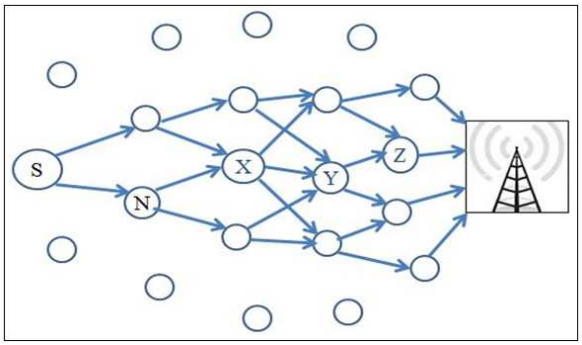

Fig. 2: Sample topology creation

Pairwise Key Establishment with 2-Hop Parents

We have applied the BIBD approach discussed in (Ruj *et al*., 2013) to establish pairwise keys between a child node and their 2-hop distance nodes during network initialization. Each sensor is loaded with set of *n* keys along with their node identities. The keys are chosen from a set of *m* keys. The set of *n* keys are chosen such that any pair of keys from m can occur in precisely two nodes. If K_1 and K_2 are two keys shared by nodes *X* and *Z* then the common unique pairwise key is $hash(K||K_2||X_i||Z_i)$. By knowing the identity of the nodes, pairwise key can be determined. Pairwise key is necessary between a child and a 2-hop distance node to encrypt the acknowledgement.

System Assumptions:

- SFAD2H assumes that the network is static and the links are bidirectional
- SFAD2H assumes that pairwise keys which are shared between Sink and each network node are programmed in nodes before deployment
- Assumed no malicious activity during network initialization
- Source nodes are assumed to be genuine

Problem Definition

The goal of the SFAD2H scheme is to detect and bypass the malicious nodes which perform selective forwarding or dropping attacks. In Fig. 1, without adversary effect, node *X* transmits packet to *Y* to forward towards Sink and node *Y* transmits packet to node *Z* to forward towards Sink. If node *X* is a source node or current sender on routing path then following are the malicious behaviors to be detected. (i) If node *Y* performs selective dropping attack, then node *X* does not hear any packet forwarding from node *Y*. (ii) Node *Z* does not send the 2-hop acknowledgement even though received the packet successfully from node *Y*, just to frame the node *Y* as malicious. (iii) Node *Y* may drop the acknowledgement received from node *Z* without further forwarding to node *X* which is at 2-hop's away from *Z*. (iv) In above three scenarios, either node *Y* is malicious or node *Z* is malicious as both can drop packets and restrain from sending acknowledgement. Problem is to detect malicious nodes among such pair of nodes < *Y*, *Z* >.

Once the child node determines the malicious behavior of current parent node, child node selects a different parent for forwarding the packet towards Sink node bypassing the malicious parent node. 2-hop's distance parent node encrypts the acknowledgment using pairwise key shared with 2-hop's distance child node. (i) Problem is to identify the list of parent nodes through which Sink can be reached with same number of hops. (ii) Identity of the node is a vital value in determining the pairwise key between a child and 2-hops distance parent node. Problem is to find the identities of the nodes at 2-hops distance by each other while the parent selected for forwarding a packet is dynamically decided. In Fig. 1, node *X* transmits packet to any of its parents say *Y*. Node *Y* transmits packet to any of its parents say *Z*. Node *X* need to find the identity of *Z* and also node *Z* need to find the identity of *X* to determine the pairwise key. (iii) Sink should be able to identify the route even when the child bypasses the malicious parent node and selects new parent node.

Selective Forwarding Attack Detection and Isolation of Malicious Node

Proposed scheme SFAD2H has creation of routing paths from every node in network upto Sink node, pairwise key establishment (Ruj *et al*., 2013) between 2-hop distance nodes, malicious node detection scheme at child to detect the selective forwarding or dropping attack from parent and Sink maintains the count of packets received on each path and compares with transaction report sent by each sensor node to finalize the malicious node list.

Detection at Sensor Node and Bypass Malicious Node

Fig. 3, shows the success case of packet transmission. As a path marker, current sender *X* adds its identity to the received packet *P* from previous hop and forwards the packet *A* to next 1-hop parent node *Y*. 1-hop node *Y* prepares packet *B* by adding its identity and transmits the packet to *Z*. Node *X* also overhears the packet transmitted from *Y*, determines the identity of 2-hop distance node and waits for the acknowledgement from *Z*. Node *X* clears the buffer and confirms the successful transmission on receiving the acknowledgement. Acknowledgment *C* is encrypted with the pairwise key determined between *X* and *Z* based on node identities and other pre-loaded keys. The packet *P* contains the data from source node and identity of the forwarded nodes. Specific to the case in Fig. 2, packet *P* contains data from source node S and identity of the forwarded node *N*.

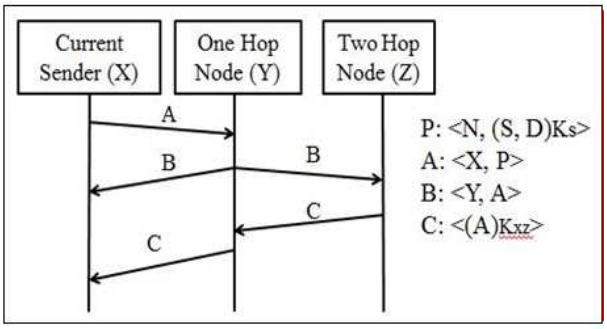

Fig. 3: Successful transmission of packet

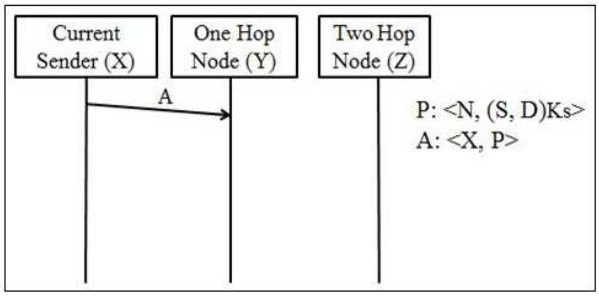

Fig. 4: Selective dropping from 1-hop node

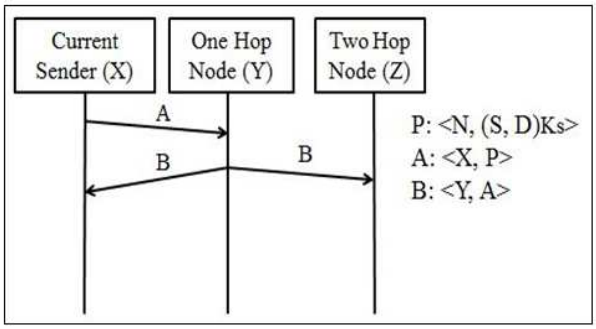

Fig. 5: No acknowledgment from 2-hop node

ID	GC	PC_CC	PID1	SC1	PID2	SC2	CID1	RC1	CID2	RC2

Fig. 6: Sample report format

Figure 4 shows the selective dropping attack from 1-hop node *Y*. Node *X* does not hear the packet transmission from Y even after the timeout period and does not receive acknowledgement packet from any 2-hop node. Node *X* determines the dropping attack from 1-hop node *Y*, increases the dropping count value on *Y*.

Figure 5 shows the case that 1-hop node *Y* does not receive the acknowledgement from 2-hop node *Z* due to *Z*'s malicious behavior, then *Y* increases the packet dropping count on *Z*. And *X* does not receive the acknowledgment from *Y* and increases the packet dropping count on *Y*.

A node changes the current parent to next parent once the current parent's dropping count crosses the threshold and node forwards the packet to Sink through different newly selected parent. Generated report can have more than one pair of parent id and forwarded packet count.

A node prepares report packet as a sample report shown in Fig. 6, encrypts with the pairwise key shared with Sink and sends to Sink node through all the parent nodes. Forwarding nodes add marker information as explained in section III. Sink differentiates data packet over report packet by the length of the decrypted content.

Whereas the report packet contains the *ID* of the node, count of the packets *GC* generated by node since the last generated report, many pairs of parent id (PID_1, PID_2) and count of packets (SC_1, SC_2) forwarded to the parent since the last report generation and many pairs of child id (CID_1, CID_2) and count of packets (RC_1, RC_2) received from children. Four most significant bits in *PC_CC* field provides the number of parent ids added in report and least significant bits in *PC_CC* field provides the number of children nodes from which packets have been received.

Packet Count Update by Sink Node

Sink maintains a table which maps from node id to the count of packets both generated by a node and also forwarded by a node. Sink updates the table with the count as it finds the node's participation in packet generation and forwarding while processing the packet. The received packet at Sink consists of sequence of node ids which are path markers added by each forwarding node and also either encrypted sensed data from source node or encrypted report data from a node. Sink does the received packet processing with below steps:

- Sink maps the marker id to a node in the routing tree at a particular level. First marker id maps to node id in the first level (Sink node being Zero level) in routing tree, second marker id maps to node id in second level and so on. i^{th} marker id mapped to node in i^{th} level is a parent of $(i+1)^{th}$ marker id mapped to node in $(i+1)^{th}$ level. Thus Sink traces the routing path to reach the source node
- After processing all the path markers upto i^{th} node, Sink decrypts the data using the pairwise key shared with first child of i^{th} node. If child node id does not match with the id in the decrypted information, Sink decrypts with the second child of i^{th} node and continues with other child nodes until finds a match. If the id matches with the node id in decrypted data, Sink evaluates the length of data and handles as a data packet or report packet.

Notations:

m: received packet at Sink
U, V: node id
SINK: sink node id
*FC: integer/*Forwarded packet count maintained by Sink for a node*/*
*GC: integer/*Generated packet count maintained by Sink for a node*/*
V_{key}: shared key between Sink and node V
*T: Table /*to store nodeid, PC and GC*/*
Algorithm 1: packet count update for each node
1: Input: Packet <m>
2: U = SINK, m' = m; success = false;
3: for each child node V of node U do
5: if m' starts with <V> then
6: T[V, FC++];
7: trim <V> from m' and get m' = m'-<V>;
8: U = V;
9: endif
10: endfor
11: for each child node V of node U do
*12: P = decSourceMsg(V_{key}, m');/*decrypts source message which is two units*/*
*13: if P starts with <V> then/*V is the source node*/*
14: T[V, GC++];
15: if length(P) > 2 then
16: call Algorithm 2 to process report P;
17: endif
18: endif
19: endfor

The packet count recorded by Sink while processing the packet will help Sink to determine the malicious node when Sink receives the report from each node.

Report Processing at Sink

Sink maintains a hash map with node id as a key and a linked list containing report information as a value. Sink updates the value in the map, which is a linked list with the report information received from each node. Linked list contains information such as node id, packet generated count, sequence of parent ids and packet sent count and sequence of children id and packet received count. Sink can make out a given node's interaction with parents and number of packets sent to each parent and also interaction with children and number of packets received from each child. Sink processes the report as per algorithm 2.

Notations

P: received report at Sink
id, PID, CID: node id
map<nodeid, LinkedList>: HashMap
*SC: integer /*packet sent count in report for a parent*/*
*RC: integer /*packet received count in report for a child*/*
*L: LinkedList /*represent a record for a id in map*/*
pcount, ccount: integer
Algorithm 2: Received and forwarded packet count update from report
1: Input: Packet <P>
2: id = P[id];
3: L = map.get(id);
4: L[GC] = L[GC] + P[GC];
*5: pcount = MSB_Value(P[PC_CC]);/*calculate the parents count based on most significant 4 bits in PC_CC*/*
6: for each i=1 to pcount do
7: if P[PIDi] exists in L then
8: L[SCi] = L[SCi] + P[SCi];
9: else
10: insert [PIDi, SCi] into L;
11: endif
12: endfor
*13: ccount = LSB_Value(P[PC_CC]);/*calculate the children count based on least significant 4 bits in PC_CC*/*
14: for each i=1 to ccount do
15: if P[CIDi] exists in L then
16: L[RCi] = L[RCi] + P[RCi];
17: else
18: insert [CIDi, RCi] into L;
19: endif
20: endfor

Malicious Node Affirmation from Sink

Sink runs the algorithm to affirm the selective dropping from each node based on the data it has. Sink has three data items such as α, β and γ for any node say *Y* in Fig. 2. To affirm the node *Y*, packet count claimed by children of and parents of *Y* are used:

- α: Count of packets a node say *Y* has both generated as well as participated in forwarding. This data is counted based on marker id added in packet while processing the packet by Sink.
- β: Sink calculates based on the reports sent by the nodes for which *Y* is a parent. This is sum of packets count from child nodes claimed in report that they sent to parent *Y*.
- γ: Sink calculates based on the reports sent by the nodes for which *Y* is a child. This is sum of packets count from parent nodes claimed in report that they received from child *Y*.

Notations:

T: Table of packet count a node generated and participated
*α, β, γ: integer/*packet count*/*
map<nodeid, LinkedList>: HashMap
*SC: integer /*packet sent count in report for a parent*/*

*RC: integer/*packet received count in report for a child*/*
*L: LinkedList/*represent a record for a id in map*/*
Algorithm 3: Packet dropping affirmation by Sink
1: Input: Table T, HashMap map
2: for each nodeid V in NW do
3: $\alpha = \beta = \gamma = 0$;
4: if V is leaf then
5: continue;
6: else
*7: int GC = T[V, GC];/*generated count(GC) stored in table for node V*/*
*8: int FC = T[V, FC];/*forwarded count(FC) stored in table for node V*/*
9: α = GC + FC;
10: for each child C of V do
11: L = map(C);
12: if L contains V then
*13: $\beta = \beta$ + L[V,SC]/*adding sent count from child C to V*/*
14: endif
15: endfor
16: for each parent P of V do
17: L = map(P);
18: if L contains V then
*19: $\gamma = \gamma$ + L[V,RC]/*adding received count from V to parent P*/*
20: endif
21: endfor
22: L = map(V);
*23: if (γ = (β + L[GC])) then/*V has not dropped*/*
24: if ($\alpha < \gamma$) then
*/*there is a dropping on the path from parent of V to Sink*/*
25: endif
*26: elseif(γ < (β + L[GC])) then/*V dropped the packets*/*
27: if ($\gamma = \alpha$) then
28: mark V malicious;
29: endif
30: endif
31: endfor

There are two cases in algorithm 3 in which selective dropping is determined.

- **Case (γ = (β + L[GC])) and ($\alpha < \gamma$)**: There is a packet dropping on the path from parent of the node under evaluation to Sink node.
- **Case (γ < (β + L[GC])) and ($\gamma = \alpha$)**: The node under evaluation has dropped the packets selectively.

Performance Analysis

The efficiency and effectiveness of proposed method SFAD2H are evaluated in NS-3 simulator. We have compared proposed approach with CRS-A approach (Ren *et al*., 2016). Simulation is done by distributing 100 stationary nodes uniformly in a 500×500 m square area. Each node is installed with 802.15.4 MAC protocol and with channel delay of 2 milli seconds. Simulation ran with generating 20 packets per node in each evaluation period. Non leaf nodes are randomly selected as malicious nodes. All nodes act as a source node and generate the data to forward towards Sink. Each node generates the transaction report packet at the end of the evaluation period. Obtained simulation results from the algorithm for various number of malicious nodes.

Percentage of Detection

Simulated and found the detection rate when the number of malicious nodes are 10, 20, 30 and 40 in the network:

$$\%\ detection = \left(\frac{No.\ of\ malicious\ nodes\ detected}{No.\ of\ malicious\ nodes\ in\ network} \right) * 100$$

For each quantity of malicious nodes, traffic is generated in 5 evaluation periods and averaged the detected malicious nodes by Sink in each evaluation period. As shown in Fig. 7, percentage of detection is improved in SFAD2H approach when compared to CRS-A approach. In CRS-A, the percentage of detection deteriorates as the number of malicious nodes increases. SFAD2H detects malicious node by Sink considering the total packets generated, forwarded and also reports received from each node.

Percentage of False Isolation

Simulated and analyzed the false detection when the number of malicious nodes are 10, 20, 30 and 40.

$$\%\ false\ detection = \left(\frac{No.\ of\ genuine\ nodes\ isolated}{No.\ of\ genuine\ nodes\ in\ network} \right) * 100$$

As shown in Fig. 8, percentage of false detection is reasonably high in CRS-A approach as the node's reputation is calculated from opinion of neighbor nodes. In SFAD2H, considered report from all children and parents of a node to avoid bad mouth attack from a particular child which tries to frame the parent as malicious by sending incorrect values to Sink. Sink detects the malicious nodes having the complete state of the network data transmitted. Sink detects the malicious activity of a node based on the claims of children and parents of a node.

Early Detection Rate

Simulated and analyzed the early detection when the number of malicious nodes are 20 in the network. In both SFAD2H and CRS-A, traffic is generated in 5 evaluation

periods of equal duration and tried to find the malicious nodes after each evaluation period. CRS-A needs long operation of the network to detect the malicious nodes as it waits until the reputation value crosses the threshold. And in CRS-A there is no way to mitigate the effect of low reputation from colluding nodes.

As shown in Fig. 9, SFAD2H detects the malicious nodes early compared to CRS-A, so that network cannot afford to lose lot of meaningful information before all malicious nodes are detected. In SFAD2H a node selects next parent node as soon as it confirms the malicious activity of the current parent node. SFA2DH detects early as it operates in rounds, detects malicious after each round.

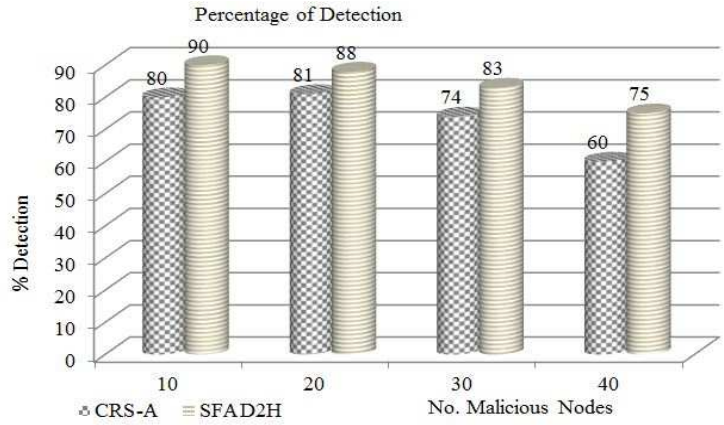

Fig. 7: Percentage of malicious node detection

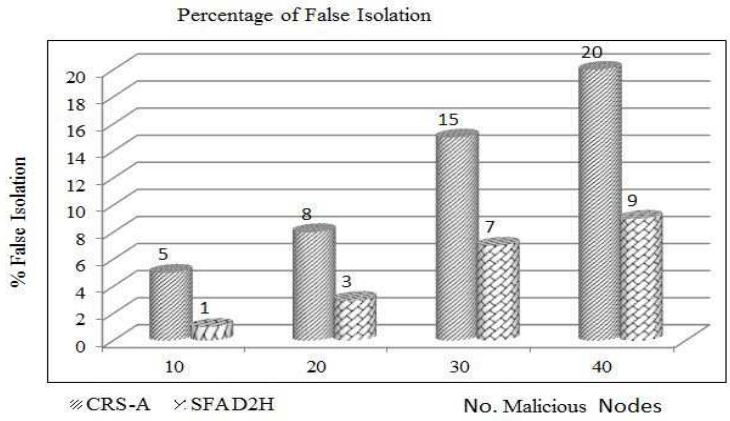

Fig. 8: Percentage of false isolation

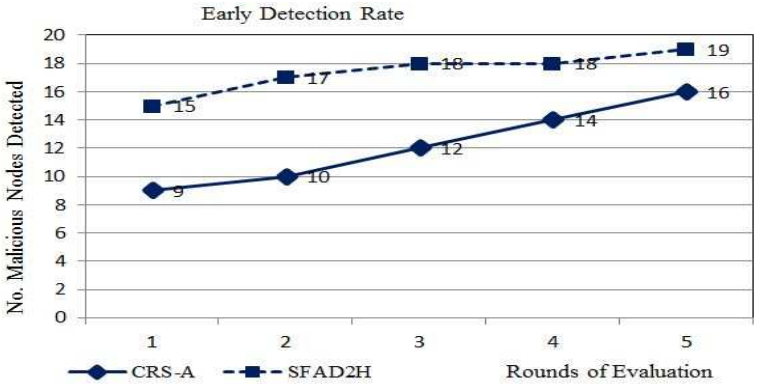

Fig. 9: Early detection rate

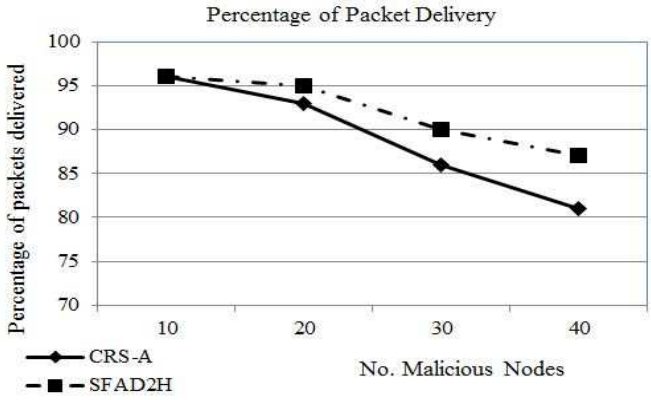

Fig. 10: Packet delivery ratio

Packet Delivery Ratio

Packet delivery ratio is evaluated when the number of malicious nodes are 10, 20, 30 and 40 in the network. Each node generates 20 packets in each evaluation period and the delivery ratio averaged over 5 consecutive evaluation periods. Delivery ratio is evaluated without packet re-transmission in SFAD2H. Packet delivery ratio is calculated by Sink node as below. *TR* is the total number of packets received by Sink during one evaluation period. *TS* is the total number of packets generated by n nodes and 20 packets per node and across 5 evaluation periods during the simulation. *PDR* is the packet delivery ratio of the network with a particular quantity of malicious nodes:

- TR = $Sum\sum_{i=1}^{n} PC_i$
- TS = 20 * 5 * n
- PDR = $Sum\sum_{n=1}^{5} TR / T$

As shown in Fig. 10, packet delivery ratio is improved as in SFAD2H, each child decides the parent node to be used on the forwarding path with its own experience of transaction with parent node. A child forwards the packet through a different parent once the child determines the malicious activity from current parent. Even Sink determines the malicious nodes early compared to CRS-A, as an effect the delivery ratio improves.

Conclusion

Selective forwarding or dropping is a critical security attack to disrupt the data integrity and degrade operation efficiency in wireless sensor networks. Proposed method is proven to be efficient to detect selective forwarding attack and bypass malicious node compared to CRS-A approach. SFAD2H starts with selection of parents for forwarding the data towards Sink. SFAD2H establishes pairwise key with 2-hop parent and expects acknowledgement from 2-hop node to detect selective dropping attacks. Early detection is possible as SFAD2H operation includes detection of malicious nodes after each evaluation period. It also provides flexibility to change the parent node based on the experience of child node with parent node. SFAD2H approach does not lose lot of meaningful information as the node changes the parent as soon as child detects malicious activity of parent. Performance results show that SFAD2H detects the malicious nodes early with high detection rate and low false detection.

Author's Contributions

All authors have contributed for the technique explained in the proposed method, implementation of the algorithm to detect the malicious nodes and also for creating the manuscript.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and there are no ethical issues involved.

References

Bendjima, M. and M. Feham, 2016. Wormhole attack detection in wireless sensor networks. Proceedings of the SAI Computing Conference, Jul. 13-15, IEEE Xplore Press, London, UK., pp: 1319-1326. DOI: 10.1109/SAI.2016.7556151

Bhuse, V., A. Gupta and L. Lilien, 2005. DPDSN: Detection of packet- dropping attacks for wireless sensor networks. Proceedings of the 4th Trusted Internet Workshop, (TIW' 05).

Butan, I., S.D. Morgera and R. Sankar, 2014. A survey of intrusion detection systems in wireless sensor networks. IEEE Commun. Surveys Tutorials, 16: 266-282. DOI: 10.1109/SURV.2013.050113.00191

Chan, H. and A. Perrig, 2003. Security and privacy in sensor networks. Computer, 36: 103-105. DOI: 10.1109/MC.2003.1236475

Chen, C., M. Song and G. Hsieh, 2010. Intrusion detection of sinkhole attacks in large-scale wireless sensor networks. Proceedings of the IEEE International Conference on Wireless Communications, Networking and Information Security, Jun. 25-27, IEEE Xplore Press, pp: 711-716. DOI: 10.1109/WCINS.2010.5541872

Cui, B. and S.J. Yang, 2014. NRE: Suppress selective forwarding attacks in wireless sensor networks. Proceedings of the IEEE Conference on Communications and Network Security, Oct. 29-31, IEEE Xplore Press, pp: 229-237. DOI: 10.1109/CNS.2014.6997490

Deng, H., A.X. Sun, B. Wang and Y. Cao, 2009. Selective forwarding attack detection using watermark in WSNs. Proceedings of the ISECS International Colloquium on Computing, Communication, Control and Management, Aug. 8-9, IEEE Xplore Press, pp: 109-119. DOI: 10.1109/CCCM.2009.5268016

Gerrigagoitia, K., R. Uribeetxeberriay, U. Zurutuzaz and I. Arenaza, 2012. Reputation-based intrusion detection system for wireless sensor networks. Proceedings of the IEEE Complexity in Engineering, Jun. 11-13, IEEE Xplore Press, Aachen, Germany, pp: 1-5. DOI: 10.1109/CompEng.2012.6242969

Ju, L., H. Li, Y. Liu, W. Xue and K. Li et al., 2010. An improved intrusion detection scheme based on weighted trust evaluation for wireless sensor networks. Proceedings of the IEEE 5th International Conference on Ubiquitous Information Technologies and Applications, Dec. 16-18, IEEE Xplore Press, Sanya, China, pp: 1-6. DOI: 10.1109/ICUT.2010.5677764

Kaplantzis, S., A. Shilton, N. Mani and Y. Sekercioglu, 2007. Detecting selective forwarding attacks in wireless sensor networks using support vector machines. Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information, Dec. 3-6, IEEE Xplore Press, Melbourne, Qld., Australia, pp: 335-340. DOI: 10.1109/ISSNIP.2007.4496866

Karlof, C. and D. Wagner, 2003. Secure routing in wireless sensor networks: Attacks and countermeasures. Proceedings of the IEEE 1st International Workshop Sensor Network Protocols and Applications, May 11-11, IEEE Xplore Press, pp: 113-127. DOI: 10.1109/SNPA.2003.1203362

Kefayati, M., H.R. Rabiee, S.G. Miremadi and A. Khonsari, 2006. Misbehavior resilient multi-path data transmission in mobile ad-hoc networks. Proceedings of the 4th ACM Workshop Security of Ad Hoc and Sensor Networks, Oct. 30-30, ACM., Alexandria, Virginia, USA., pp: 91-100. DOI: 10.1145/1180345.1180357

Khalil, I.M. and S. Bagchi, 2011. Stealthy attacks in wireless ad hoc networks: Detection and countermeasure. IEEE Trans. Mobile Comput., 10: 1096-1112. DOI: 10.1109/TMC.2010.249

Khalil, I.M., 2011. ELMO: Energy aware local monitoring in sensor networks. IEEE Trans. Dependable Secure Comput., 8: 523-536. DOI: 10.1109/TDSC.2010.74

Li, S., S. Zhao, X. Wang, K. Zhang and L. Li, 2014. Adaptive and secure load-balancing routing protocol for service-oriented wireless sensor networks. IEEE Syst. J., 8: 858-867. DOI: 10.1109/JSYST.2013.2260626

Lim, S. and L. Huie, 2015. Hop-by-hop cooperative detection of selective forwarding attacks in energy harvesting wireless sensor networks. Proceedings of the IEEE International Conference on Computing, Networking and Communications, Feb. 16-19, IEEE Xplore Press, Garden Grove, CA, USA., pp: 315-319. DOI: 10.1109/ICCNC.2015.7069361

Liu, A., Z. Zheng, C. Zhang, Z. Chen and X. Shen, 2012. Secure and energy-efficient disjoint multipath routing for wsns. IEEE Trans. Veh. Technol., 61: 3255-3265. DOI: 10.1109/TVT.2012.2205284

Mavropodi, R., P. Kotzanikolaou and C. Douligeris, 2007. SECMR-a secure multipath routing protocol for ad hoc networks. Ad Hoc Netw., 5: 87-99. DOI: 10.1016/j.adhoc.2006.05.020

Pavithra, B. and K.S. Reddy, 2015. Energy efficient detection of malicious nodes using secure clustering with load balance and reliable node disjoint multipath routing in wireless sensor networks. Proceedings of the International Conference on Advances in Computing, Communications and Informatics, Aug. 10-13, IEEE Xplore Press, Kochi, India, pp: 954-958. DOI: 10.1109/ICACCI.2015.7275734

Prathap, U., K.B. Nisha, P.D. Shenoy and K.R. Venugopal, 2015a. SDLM: Source detection based local monitoring in wireless sensor networks. Proceedings of the IEEE Region 10th Conference, Nov. 1-4, IEEE Xplore Press, pp: 1-5. DOI: 10.1109/TENCON.2015.7372989

Prathap, U., P.D. Shenoy and K.R. Venugopal, 2015b. CPMTS: Catching packet modifiers with trust support in wireless sensor networks. Proceedings of the IEEE International WIE Conference on Electrical and Computer Engineering, Dec. 19-20, IEEE Xplore Press, Dhaka, Bangladesh, pp: 255-258. DOI: 10.1109/WIECON-ECE.2015.7443911

Ren, J., Y. Zang, K. Zang and X. Shen, 2016. Adaptive and channel-aware detection of selective forwarding attacks in wireless sensor networks. IEEE Trans. Wireless Commun., 15: 3718-3731. DOI: 10.1109/TWC.2016.2526601

Ruj, S., A. Nayak and I. Stojmenovic, 2013. Pairwise and triple key distribution in wireless sensor networks with applications. IEEE Trans. Comput., 62: 2224-2237. DOI: 10.1109/TC.2012.138

Shu, T., M. Krunz and S. Liu, 2010. Secure data collection in wireless sensor networks using randomized dispersive routes IEEE Trans. Mobile Comput., 9: 941-954. DOI: 10.1109/TMC.2010.36

Stehlik, M., V. Matyas and A. Stetsko, 2016. Towards better selective forwarding and delay attacks detection in wireless sensor networks. Proceedings of the 13th IEEE International Conference on Networking, Sensing and Control, IEEE Xplore Press, Apr. 28-30, Mexico City, Mexico, pp: 1-6. DOI: 10.1109/ICNSC.2016.7478978

Yang, H., F. Ye, Y. Yuan, S. Lu and W. Arbaugh, 2005. Toward resilient security in wireless sensor networks. Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, May 25-27, ACM, Urbana-Champaign, IL, USA., pp: 34-45. DOI: 10.1145/1062689.1062696

Ye, F., H. Luo, S. Lu and L. Zhang, 2004. Statistical en-route filtering of injected false data in sensor networks. Proceedings of the 33th Annual Joint Conference of the IEEE Computer and Communications Societies, Mar. 7-11, IEEE Xplore Press, Hong Kong, China, pp: 839-850. DOI: 10.1109/INFCOM.2004.1354666

Zhu, S., S. Setia, S. Jajodia and P. Ning, 2004. An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. Proceedings of the IEEE Symposium on Security and Privacy, May 12-12, IEEE Xplore Press, USA., pp: 259-271. DOI: 10.1109/SECPRI.2004.1301328