International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp. 6926-6934

Detecting Selective Packet Droppers and Modifiers with Triple Key Distribution in Wireless Sensor Networks

Prathap U, Deepa Shenoy P and Venugopal K R

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

Abstract

Packet dropping and modifying are important security attacks among the active attacks in wireless sensor networks. We have proposed a technique to detect malicious nodes which perform selective packet modification and dropping. Any consecutive three nodes on the routing path secure the 2-hop channel with the distribution of polynomial based triple key. A node encrypts the packet with a triple key shared with 1-hop and 2-hop nodes, observes the packet forwarding from 1-hop node and decides the malicious activity of the 1-hop node. Every node takes responsibility of detecting the attacks by 1-hop distance node on the routing path towards destination or Sink. The proposed method considers tree topology with multiple hops to reach Sink from the source node. The proposed method provides a solution to the triple key overlapping problem arises in multi-hop routing path. Sink detects the malicious nodes by running a heuristic algorithm considering the number of packets received from each node. We have simulated the algorithm in NS-3 and compared the performance results with recently proposed approaches. Simulation results show that proposed method detects the malicious nodes efficiently and early.

Keywords: WSN, malicious node, packet dropping, packet modification, selective forwarding, triple key.

INTRODUCTION

A. 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. A Sensor node, also known as a 'mote', is a node in a wireless sensor network that can perform 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 ad-hoc way and finally to a base station where the data is stored, processed and displayed.

B. Security Attacks in Wireless Sensor Network

Sensor nodes are vulnerable to a wide range of active attacks [1], [2]. An 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 with programs to perform malicious activity. Sybil attack - attacker deployed nodes may also use the identities of the other genuine nodes to frame other genuine nodes as malicious and to achieve intruder behavior. In sinkhole attack [3] malicious node attracts the routing data by publishing the shortest path to Sink and drops most of the packets without forwarding further towards the destination or modifies the forwarded packets. Packet dropping, modification, misrouting are basic problems which have a large impact on the information gathered by sensor nodes as network loses a lot of important sensed data. Cryptography techniques alone are not sufficient to protect the data. Attacks such as colluding collision [4], misrouting, sinkhole, wormhole [5], rushing attacks can be launched without the help of cryptography keys [6].

Sink node or the aggregating node aggregates the sensed information received from different sensors to make a meaningful data or to understand the environment. The integrity of the sensed data cannot be trusted due to the packet dropping and modification attacks. Wireless medium is inherently not reliable as communication incurs data loss. With the selective packet modification and dropping attacks, the challenge is to decide whether the data dropped or modified by a node is due to malicious behavior or due to unreliable wireless medium.

C. Introduction to proposed Approach

In this paper, we propose a technique to detect malicious nodes which perform selective dropping and modification attacks. 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 triple key with selected 1-hop parent and 2-hop grandparent nodes. The triple key is generated based on a polynomial, and three parties can secure the data

communication. 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 detecting the malicious nodes and tracing the routing path up to 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. A child node requests 1-hop parent node to share the key to secure the communication with up to 2-hops. 1-hop parent node generates triple key and shares with child and 2-hop grandparent node; thus 2-hops path is secured for packet transmission. Both child and parent encrypt the packet with the shared triple key before sending the packet. The child observes the parent for packet modification and dropping as the child shares the key with the parent. In figure 2, node *X* forwards the packet to next hop node *Y* and observes node *Y* for modification and dropping. Node *X* maintains the count of successful and unsuccessful packet transmission from *Y* based on observation. A node adds observation factor on the parent to every packet it generates and forwards. Sink also computes the observation factor on each node while processing packets received from all nodes. Sink detects the malicious nodes by comparing the observation factor received from individual nodes with observation factor estimated while processing the packet.

In order to detect the selective dropping attack 'Adaptive and Channel-Aware Detection of Selective Forwarding At- tacks in Wireless Sensor Networks (CRS-A) [7]' 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 net- work. We proposed a method 'SFAD2H: Selective Forwarding or Dropping Attack Detection with 2-Hop Acknowledgment Support [8]' to detect selective forwarding attack which is based on 2-hop acknowledgment protocol and Sink detects the malicious nodes based on reports received from each node in network. We provide a simulated analysis comparing the CRS-A approach, SFAD2H approach, and our proposed approach. The rest of the paper is organized as follows, section II discusses 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 basic technique widely applied to minimize the impact of selective dropping and modification attacks on data delivery. The idea is either sending multiple copies of the same data through different paths to destination [9], [10], [11], [12], [13] or splitting the data into N shares and sending the N shares through different paths to destination [14], [15], [16]. 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 used to find the malicious activities such as packet modification and dropping of the current forwarding node [17], [18], [19], [20], [21], [22], [23]. 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, 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 a 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.

[24] proposed a centralized intrusion detection scheme based on Support Vector Machines (SVMs). This approach has used sliding windows for selective forwarding attacks and 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.

[25] proposed an approach to secure the data transmission and detecting a selective forwarding attack. Authors 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. The base station compares the extracted watermark to the original watermark to detect a selective forwarding attack and determines the packet loss with the help of watermark.

[26] proposed two parameterized collaborative intrusion detection techniques and optimized 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.

[27] provided an analytical model to estimate the wellness of a node's forwarding behavior. They borrowed the idea of the PageRank algorithm to determine 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 early detection.

[28] proposed audit based misbehavior detection to detect packet droppers. The approach integrates reputation management, trustworthy route discovery and detection of malicious nodes based on behavioral nodes. An audit is done in a specific duration to detect the behavioral pattern. The approach suffers if malicious behavior changes over time.

The scheme proposed by [7] is based on neighbor node observation. The optimal packet loss threshold due to selective forwarding attack is estimated over the inherent loss

due to the wireless channel by evaluating the channel forward 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 the 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

A. Network Model and Initialization

We 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 [29]. Sink node transmits the path distance information to 1-hop neighbors say node *Z* in figure 1. 1-hop neighbors increment the distance information and share with 2-hop neighbors and continues till the last hop node. In figure 1, node *Z* increments the distance count by 1 and shares 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 requests the selected 1-hop parent node to establish a triple key among the node, 1-hop parent node and 2-hop grandparent node to secure the 2-hops channel towards Sink. 1-hop parent node generates a triple key and shares with child and 2-hops parent node. The proposed approach has two levels of data encryption. Every node has a pre-distributed key shared with Sink to encrypt the marker data and sensor data added to the packet. And a triple key is used to encrypt the entire packet before forwarding to next hop node.

During packet forwarding, intermediate node prepares marker data containing < *node identity, observation factor on parent* > and adds to the packet before forwarding the packet to the parent node. Marker data is encrypted with the pairwise key shared with Sink. Marker data added by each node helps Sink to trace the nodes participated in forwarding the packet [29]. All the nodes transmit the sensed data to- wards Sink for processing. The typical packet format looks like < $(id_Z, OF_{SNK})_{KZ}$, $(id_Y, OF_Z)_{KY}$, $(id_X, OF_Y)_{KX}$, $(id_S, OF_X, SN, D)_{KS}$ > for the data generated from node

S, where id_Z, id_Y, id_X , and id_S are node identities on the forwarding path added in path marker by respective node, OF_{SNK}, OF_Z, OF_Y, and OF_X are observation factors for parents added by their individual children nodes, *D* is the data generated by the source node *S*, *SN* is sequence number of the packet generated by the source node *S*, and K_Z, K_Y, K_X, and K_S are the pairwise keys shared between Sink and respective node for data encryption.

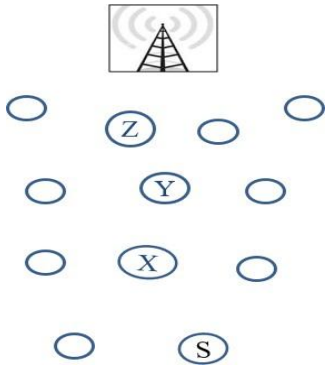

Figure 1. Initial deployment of nodes

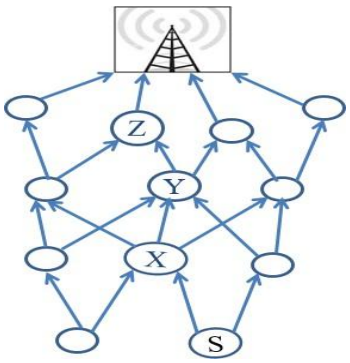

Figure 2. Sample topology creation

B. Triple key establishment among 2-hop distance nodes

We have applied the polynomial based triple key generation approach discussed by [30]. When child node requests 1-hop parent node to generate and distribute the triple key, 1-hop parent node chooses a *C*-degree symmetric polynomial with coefficients from F_q (finite field with q elements, *q* can be $2^{16}+1$ or $2^{32}+1$). The chosen polynomial by the parent is in three variables *P(a,b,c)* and parent assigns to both child and 2-hop grandparent the value *P(a,Y,c)* of the polynomial evaluated at point *Y*. Three nodes *X*, *Y*, and *Z* in figure 2 calculates the shared key as *P(X,Y,Z)* for sharing information. Every combination of three nodes which form 2-hops on the routing path to Sink shares a common key with the distribution of triple key.

Consider the nodes *X*, *Y*, and *Z* in figure 2, which form 2-hops path, triple key distribution takes place as per the following steps.

i) Node *X* being child requests 1-hop parent *Y* to generate and share the triple key among *X*, *Y* and *Z*.

i) Parent node *Y* chooses a symmetric polynomial *P(a,b,c)* of degree *c*, with coefficients in F_q.

iii) Parent *Y* assigns the share *P(a,b,X)* to child node *X* and *P(a,b,Z)* to 2-hop grandparent node *Z*.

iv) Triple key among nodes *X*, *Y*, and *Z* is calculated as *P(X,Y,Z)*.

Example 1. Let $q=7$ and $P(a,b,c)=a^2 + b^2 + c^2 + 4ab + 4bc + 4ca + a + b + c + 1$.

Nodes are given the following shares considering the identities of *X, Y,* and *Z* are respectively 1, 2 and 3:

Node X: $P(a,b,1) = a^2+b^2+4ab+5a+5b+3$

Node Y: $P(a,b,2) = a^2+b^2+4ab+2a+2b$

Node Z: $P(a,b,3) = a^2+b^2+4ab+6a+6b+6$

For instance, nodes *X, Y,* and *Z* calculates the triple key K_{xyz} as $(P(X,Y,Z) \bmod q)$ i.e., $P(3,2,1)=P(1,3,2)=P(1,2,3)=2$.

System Assumptions:

i) Proposed approach assumes that the network is static and the links are bidirectional.

ii) Proposed approach assumes that pairwise keys which are shared between Sink and each network node are programmed in nodes before deployment.

iii) Assumed no malicious activity during network initialization.

iv) Source nodes are assumed to be genuine.

C. Problem Definition

The goal of the proposed scheme is to detect the malicious nodes which perform selective dropping and modification attacks. In figure 2, without adversary effect, node *S* transmits the packet to *X* to forward towards Sink and node *X* transmits the packet to node *Y* to forward towards Sink. If node *S* is a source node or current sender on routing path then following are the malicious behaviors to be detected.

i) If node *X* selectively drops the packet, then node *S* must detect the attack by *X* and update the observation factor value on *X*.

ii) If node *X* performs selective modification attack, then node *S* must observe the modification from *X* and update the observation factor value on *X*.

iii) Every node must share the observation factor on the parent with Sink node.

iv) To keep the 2-hop channel secure and confirm that dropping and modification from the parent node, a triple key need to be established among three nodes which make 2-hops.

v) Sink need to estimate the observation factor of a parent node based on the packets receives and processes, and compare the estimated observation factor with observation factor recorded by each node.

vi) Even with the access to the triple key, an intermediate node should not make out the contents of the packet.

SELECTIVE DROPPING AND MODIFICATION ATTACKS DETECTION

The proposed scheme has following steps of operation to detect malicious nodes which perform selective dropping and modification attacks.

i) Creation of routing paths from every node in network up to Sink node.

ii) Triple key establishment [30] among nodes which make the 2-hop distance on routing path.

iii) Malicious behavior detection at a child to detect the selective modification and dropping attacks from parent and maintaining observation factor.

iv) Sink processes the packets received and maintains a count of packets participated by each node in forwarding and count of packets generated from source nodes.

iv) Sink estimates the observation factor based on node's participation count and compares with observation factor sent by each sensor node to detect the malicious nodes.

A. Attacks Detection at Sensor Node

Case 1 in figure 3, shows the success case of packet transmission. Node *X* receives the packet $P: < ((S, OF_X, SN, D)K_S)K_{SXY} >$ from source node *S*, where *P* is encrypted with pairwise key K_S shared with Sink and with triple key K_{SXY} to observe the modification from the parent. Node *X* decrypts the packet *P* with triple key K_{SXY} and adds an encrypted path marker $(X,OF_Y)K_X$ containing identity *X* and observation factor OF_Y on the parent *Y*. Path marker is encrypted with pairwise key K_X shared between Sink and node *X*. Node *X* encrypts the packet to be forwarded $A: < ((X,OF_Y)K_X, P)K_{SXY} >$ with triple key K_{SXY} , where the triple key K_{SXY} is shared among nodes *S*, *X*, and *Y*. Node *Y* decrypts the received packet *A* with triple key K_{SXY} and adds an encrypted path marker $(Y,OF_Z)K_Y$ containing identity *Y* and observation factor OF_Z on parent *Z*. Path marker is encrypted with pairwise key K_Y shared between Sink and node *Y*. Node *Y* encrypts the packet to be forwarded $B: < ((Y,OF_Z)K_Y , A)K_{XY\ Z} >$ with triple key $K_{XY\ Z}$, where the triple key $K_{XY\ Z}$ is established among nodes *X, Y,* and *Z*.

Case 2 in figure 3, shows the case that 1-hop node *Y* modifies the packet before forwarding. Node *X* observes the modified packet and compares with the buffered packet to determine the modification from the parent node *Y*. Node *X* increases the modification count on parent *Y* and uses to calculate observation factor.

Case 3 in figure 3, 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. Node *X* determines the dropping attack from 1-hop node *Y*, increases the dropping count value on *Y* and uses to calculate observation factor.

Key Overlapping Problem and Solution:

Nodes *X* and *Y* being part of both 2-hop paths such as *(S, X,*

Y) and *(X, Y, Z)*. Both *X* and *Y* has two triple keys established as part of both the 2-hops. Node *S* being a child node, requests parent *X* to establish a triple key among *S*, *X*, and *Y*. All three nodes *S*, *X*, and *Y* add the triple key to the key list by tagging the key with path identifier (*S, X, Y)*. Node *X* being a child node, requests parent node *Y* to establish triple key among *X*, *Y* and *Z*. All three nodes *X*, *Y*, and *Z* add the triple key to the key list by tagging the key with path identifier *(X, Y, Z)*. Path identifier consists of node ids respectively child id, 1-hop parent id, and 2-hop parent id.

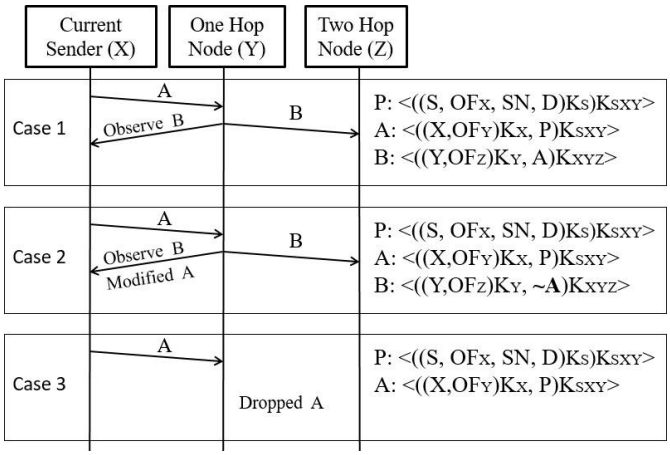

Figure 3. Different scenarios in packet forwarding

When node *X* receives a packet from *S* and needs to forward the packet *Y*, node *X* uses a triple key tagged with path identifier *(S, X, Y)* for both decryption and encryption. Node *S* can observe the parent and detect any modification as they share a same triple key K_{SXY}. When node *Y* receives a packet from *X*, decrypts the packet with key K_{SXY}, and encrypts with key K_{XYZ} before forwarding to *Z*. Node *X* can still detect the modifications from parent *Y* even though the forwarded packet is encrypted with key $K_{XY\ Z}$, as node *X* has the triple key $K_{XY\ Z}$ tagged with path identifier *(X, Y, Z)*. Key to be used for encryption and decryption during forwarding is selected by creating a path identifier based on the id of the node from which packet is received and the id of the node to which packet need to be forwarded.

B. Packet Participation Update by Sink Node

Sink maintains a table as shown in table 1. The table contains data as follows,

i) *ID* is an identity of a node, and there will be a row for every node in the network.

ii) *PID* is parent node identity for the node in *ID* column.

iii) *OF* is an observation factor of a parent node by its child node. Sink continuously update *OF* value during packet processing. iv) *SNum* is the latest sequence number seen in packet generated from a source node. v) *APart* is the node's actual participation in forwarding a packet. vi) *IPart* is the node's ideal participation in forwarding a packet. vii) *EOF* is an estimated observation factor by Sink node.

Both *ID* and *PID* are updated during network initialization. *OF*, *SNum*, and *APart* are updated while processing the packet. *IPart* and *EOF* are updated when Sink runs malicious node detection algorithm.

Table I. Data Maintained By Sink

ID	PID	OF	SNum	APart	IPart	EOF
S	X					
X	Y					
Y	Z					
Z	SNK					
...	...					

On receiving the packet from a neighbor node, Sink starts processing the packet and update the table with latest values accordingly.

i) Path markers are processed by mapping to a node at each level in a tree created during initialization [29]. Sink being at the zeroth level, the first marker is decrypted with a pairwise key shared with a node in the first level and i^{th} marker is processed by decrypting with the key shared with a node in i^{th} level of the tree. Marker information is decrypted and processed level by level to reach the source node. At any level, Sink decrypts the marker by a key shared with the first child and compares the *ID* in marker with child *ID* and if *ID* does not match, Sink decrypts the marker with key shared with siblings at the same level to detect the forwarded node. If the marker is successfully decrypted and matched with a node *ID*, table 1 is updated with observation factor in marker *(OF)*. And *APart* value is incremented by one as Sink found the participation of a node in forwarding.

ii) If Sink fails to detect the forwarded node in *step i* at i^{th} level, Sink tries to decrypt four values and tries to match for a source node. Sink determines the source node, if the decrypted 4 values start with a node ID. Sink tries with all nodes at i^{th} level, to find the source node. If source node is located, table is updated with latest observation factor and with most recent sequence number.

iii) If Sink fails to process the packet and fails to gather data generated by the source node, Sink determines the packet modification by some forwarding node between the source node and last successfully processed marker node and discards the packet.

Notations:

m: received packet at Sink U, V: node id

SNK: Sink node id

V_{key}: shared key between Sink and node V

OF_U: observation factor from node V on U success: boolean to track successful decryption T: table maintained by Sink

Algorithm 1: Packet Participation Update at Sink

1: Input: Packet <m>

2: U = SNK, m' = m; success = false; 3: ***for*** *each child node V of node U* ***do***

*4: P = decMarker(V_{key}, m'); /*decrypts only marker which is two units*/*

5: ***if*** *P starts with [V, OF_U]* ***then***

6: T[V][OF] = OF_U;

7: T[V][APart]++;

8: trim [V, OF_U] from P and get m' = P-[V, OF_U];

9: U = V; go to line 3;

10: ***end if***

11: ***end for***

12: ***for*** *each child node V of node U* ***do***

*13: P = decSourceMsg(V_{key}, m'); /*decrypts source message which is four units*/*

14: ***if*** *P starts with [V, OF_U, SNum, D]* ***then*** */*V is the source node*/*

15: T[V][OF] = OF_U;

16: T[V][SNum] = SNum;

17: success = true; ***break****; 18:* ***end if***

19: ***end for***

20: ***if*** *success = false* ***then*** *21: drop this packet;*

22: ***end if***

C. Observation Estimation and Malicious Node Detection from Sink

Sink runs malicious node detection algorithm based on the latest values in the table. While processing the packets, Sink update the details such as observation factor, maximum sequence number, and actual participation of a node in forwarding the packet. Sink estimates the ideal participation of a node in packet forwarding and calculates the observation factor on each node before running malicious detection procedure. Compares the observation factor received from each node with the estimated observation factor to determine the packet droppers and modifiers.

i) A node's participation count is a cumulative count of packets generated by all descendant source nodes. Iterate over each node in the table, say *S*, obtain the sequence number, find all the nodes on the routing path from *S* to Sink node, and add sequence number of *S* to *IPart* value of the nodes from S to Sink.

ii) Estimated observation factor is a ratio of actual participation to ideal participation.

iii) A node is declared as malicious only if both observation factor and estimated observation factor are less than the threshold value.

Notations:

T: table maintained by Sink SNK: Sink node id

SqNum: sequence number NodeID: node id

threshold: predefined threshold value on malicious activity
Algorithm 2: Observation Estimation and Malicious Node Detection

1: Input: T

2: ***for*** *each node T[ID] in T* ***do***

3: SqNum = T[ID][SNum];

4: NodeID = T[ID][PID];

5: ***while*** *NodeID != SNK* ***do***

6: T[NodeID][IPart] = T[NodeID][IPart] + SqNum;

7: NodeID = T[NodeID][PID];

8: ***end while***

9: ***for*** *each node T[ID] in T* ***do***

*10: T[ID][EOF] = T[ID][APart]/T[ID][IPart]*100;*

11: ***if*** *T[ID][OF] < threshold and T[ID][EOF] < threshold*

then

12: / declare node with ID as malicious */ 13:* ***end if***

14: ***end for***

PERFORMANCE ANALYSIS

The efficiency and effectiveness of proposed method 'Detecting Selective Packet Droppers and Modifiers with Triple Key Distribution(DSPDMT)' are evaluated in NS-3 simulator. We have compared results of the proposed approach with CRS-A [7] and SFAD2H [8] approaches. Simulation is done by distributing 100 stationary nodes uniformly in a 500m x 500m square area. Each node is installed with 802.15.4 MAC protocol and with channel delay of 2 milli-seconds. The simulation ran with generating 20 packets on an average per node. Non-leaf nodes are randomly selected as malicious nodes. All non-malicious nodes act as a source node and generate the data to forward towards Sink. Sink runs the malicious node detection procedure after all packets are generated and forwarded. Simulation is run for the various number of malicious nodes and results are obtained.

A. Percentage of Detection

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

*% detection = (No. of malicious nodes detected / No. of malicious nodes in network)*100*

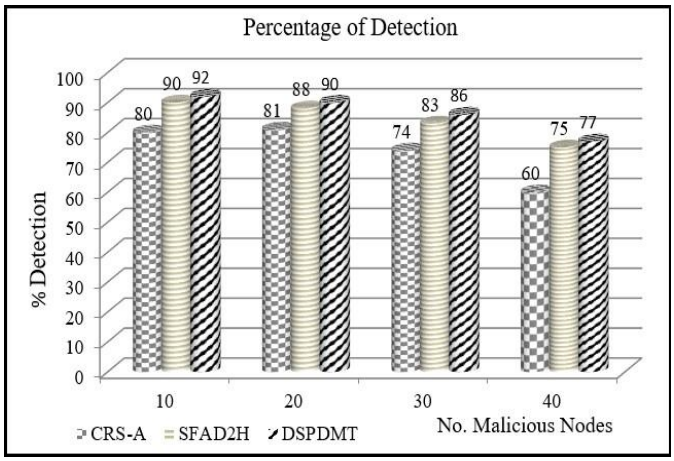

Figure 4. Percentage of malicious node detection

For each quantity of malicious nodes, as shown in figure 4, the percentage of detection is improved in DSPDMT over SFAD2H and CRS-A approaches. In CRS-A, the percentage of detection deteriorates as the number of malicious nodes increases. SFAD2H detects malicious nodes with 2-hop acknowledgment(ACK) protocol and based on reports received at Sink. If reports are dropped by malicious node, the percentage of detection goes down. DSPDMT does not depend on 2-hop ACK and reports from individual nodes as Sink accesses observation factor on each node while processing the packet.

B. Percentage of False Isolation

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

*% false detection = (No. of genuine nodes isolated / No. of genuine nodes in network)*100*

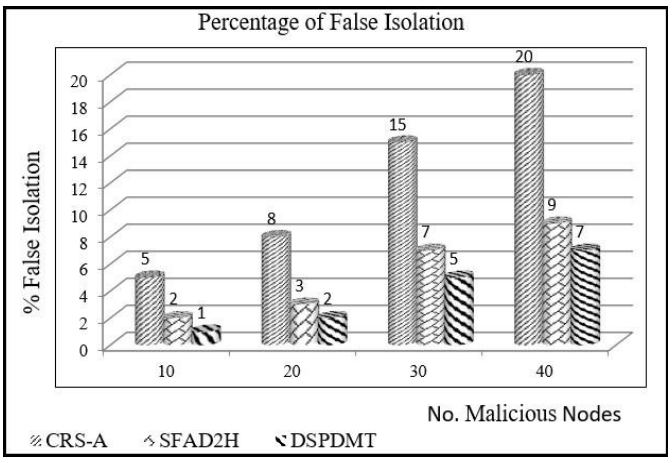

Figure 5. Percentage of false isolation

As shown in figure 5, the percentage of false detection is reasonably high in CRS-A and SFAD2H when compare to DSPDMT approach. In CRS-A approach the node's reputation is calculated from opinion of neighbor nodes. In SFAD2H, Sink detects the malicious activity of a node based on the claims of children and parents of a node. If Sink does not receive the report from a node, the detection goes wrong.

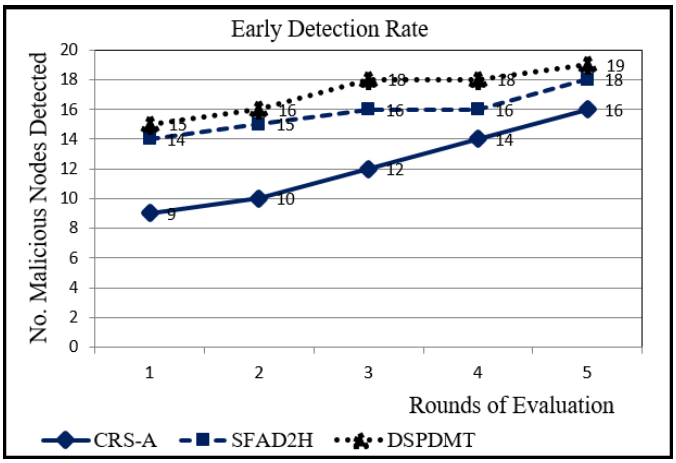

Figure 6. Early Detection Rate

In factor is shared with Sink in every forwarded packet. Early detection is possible as Sink has real-time knowledge of the droppers and modifiers.

D. Packet Delivery Ratio

Packet delivery ratio is evaluated when the 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. The delivery ratio is evaluated without packet re-transmission. Sink node calculates the packet delivery ratio as follows. *TR* is the total number of packets received by Sink across 5 evaluation periods. *TG* 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.

DSPDMT, every node secures the 2-hop channel and Sink gains knowledge of droppers and modifiers in real time while processing packet. Sink cross-validates the observation factor sent by each node with its own estimated value to reduce the false isolation.

C. Early Detection Rate

Simulated and analyzed the early detection when the malicious nodes are 20 in the network. In DSPDMT, 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 the 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 figure 6, DSPDMT detects early compared to other two approaches. In SFAD2H, Sink depends on report packet to detect malicious nodes. The detection of malicious nodes is delayed, if malicious nodes selectively drop report packets. In DSPDMT, the incremental value of observation figure 7 shows that packet delivery ratio is improved in DSDPMT compare to other two approaches.

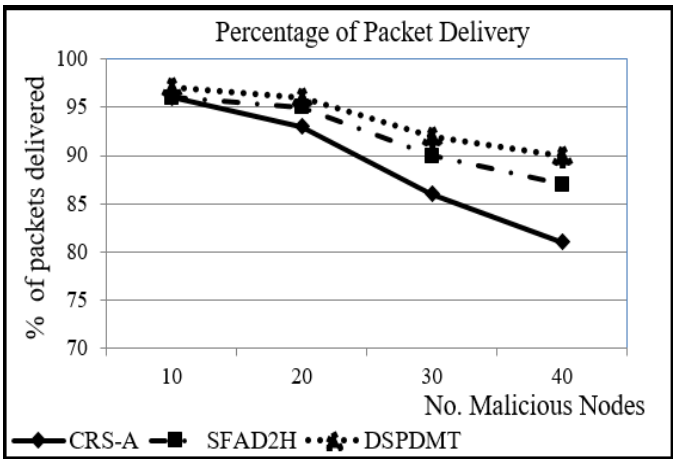

Figure 7. Packet Delivery Ratio

CONCLUSION

Selective dropping and modification are critical active security attacks to disrupt the data integrity and degrade operation efficiency in wireless sensor networks. The proposed method is proven to be effective to detect selective dropping and modification attacks compared to CRS-A and SFAD2H approaches. The proposed approach establishes the triple key to secure every 2-hop channel and Sink detects the malicious nodes comparing observation factor shared from nodes with its own estimated value. Early detection is possible as Sink has real-time knowledge of droppers and modifiers. Performance results show that proposed approach detects the malicious nodes early with high detection rate and with low false isolation even when bad nodes perform both selective dropping and modification attacks. Optimization of packet length can be applied as a further improvement to the proposed approach.

REFERENCES

[1] H. Chan and A. Perrig, ”Security and Privacy in Sensor Networks,” In *Computer*, volume 36, pp. 103-105, Oct 2003.

[2] I. Butan, S. D. Morgera and R. Sankar, ”A Survey of Intrusion Detection Systems in Wireless Sensor Networks,” In *IEEE Communications Surveys & Tutorials*, volume 16, no. 1, pp. 266–282, First Quarter 2014.

[3] C. Chen, M. Song, and G. Hsieh, ”Intrusion Detection of Sinkhole Attacks In Large-scale Wireless Sensor Networks,” In *IEEE Int’l Conf. on Wireless Communications, Networking and Information Security (WCNIS)*, pp. 711–716, June 2010.

[4] I. M. Khalil, S. Bagchi, ”Stealthy Attacks in Wireless Ad Hoc Networks: Detection and Countermeasure,” In *IEEE Transactions on Mobile Computing*, volume 10, no. 8, pp. 1096–1112, August 2011.

[5] M. Bendjima, and M. Feham, ”Wormhole Attack Detection in Wireless Sensor Networks,” In *Proc. SAI Computing Conference*, pp. 1319–1326, July 2016.

[6] I. M. Khalil, ”ELMO: Energy Aware Local Monitoring in Sensor Networks,” In *IEEE Transactions on Dependable and Secure Computing*, volume 8, pp. 523-536, August 2011.

[7] J. Ren, Y. Zang, K. Zang, and X. Shen, ”Adaptive and Channel-Aware Detection of Selective Forwarding Attacks in Wireless Sensor Networks,” In *IEEE Transactions on Wireless Communications*, volume 15, no. 5, pp. 3718–3731, May 2016.

[8] Prathap U, Kiran K, P Deepa Shenoy, and Venugopal K R, ”SFAD2H: Selective Forwarding or Dropping Attack Detection with 2-Hop Acknowl- edgment Support in Wireless Sensor Networks,” In *American Journal of Engineering and Applied Sciences*, volume 10, no. 04, pp. 908–918, Nov. 2017.

[9] C. Karlof and D. Wagner, ”Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures,” In *Proc. IEEE First Int’l Workshop Sensor Network Protocols and Applications*, pp. 113–127, 2003.

[10] V. Bhuse, A. Gupta, and L. Lilien, ”DPDSN: Detection of Packet- Dropping Attacks for Wireless Sensor Networks,” In *Proc. Fourth Trusted Internet Workshop*, 2005.

[11] M. Kefayati, H.R. Rabiee, S.G. Miremadi, and A. Khonsari, ”Misbehav- ior Resilient Multi-Path Data Transmission in Mobile Ad-Hoc Networks,” In *Proc. Fourth ACM Workshop Security of Ad Hoc and Sensor Networks (SASN 06)*, 2006.

[12] R. Mavropodi, P. Kotzanikolaou, and C. Douligeris, ”SECMRa Secure Multipath Routing Protocol for Ad Hoc Networks,” In *Ad Hoc Networks*, volume 5, pp. 87-99, 2007.

[13] B. Pavithra, and K. Satyanarayan R, ”Energy Efficient Detection of Malicious Nodes using Secure Clustering with Load Balance and Reliable Node Disjoint Multipath Routing in Wireless Sensor Networks,” In *Proc. IEEE ICACCI*, pp. 954–958, 2015.

[14] T. Shu, M. Krunz, and S. Liu, ”Secure Data

Collection in Wireless Sensor Networks using Randomized Dispersive Routes," In *IEEE Trans. Mobile Computing*, volume 9, no. 7, pp. 941–954, Jul. 2010.

[15] A. Liu, Z. Zheng, C. Zhang, Z. Chen, and X. Shen, "Secure and Energy Efficient Disjoint Multipath Routing for WSNs," In *IEEE Trans. Vehicular Technology*, volume 61, no. 7, pp. 3255–3265, Sep. 2012.

[16] S. Li, S. Zhao, X. Wang, K. Zhang, and L. Li, "Adaptive and Secure Load-Balancing Routing Protocol for Service-Oriented Wireless Sensor Networks," In *IEEE Systems Journal*, volume 8, no. 3, pp. 858-867, Sep. 2014.

[17] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical En-Route Filtering of Injected False Data in Sensor Networks," In *Proc. IEEE INFOCOM*, pp. 839–850, 2004.

[18] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An Interleaved Hop-by-Hop Authentication Scheme for Filtering False Data in Sensor Networks," In *Proc. IEEE Symp. Security and Privacy*, pp. 259–271, 2004.

[19] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, "Toward Resilient Security in Wireless Sensor Networks," In *Proc. Sixth ACM Intl Symp. Mobile Ad Hoc Networking and Computing (MobiHoc 05)*, 2005.

[20] Prathap U, Nisha K B, P Deepa Shenoy, and Venugopal K R, "SDLM: Source Detection Based Local Monitoring in Wireless Sensor Networks," In *Proc. IEEE TENCON*, pp. 1–5, Nov 2015.

[21] S. Lim, and L. Huie, "Hop-by-Hop Cooperative Detection of Selective Forwarding Attacks in Energy Harvesting Wireless Sensor Networks," In *Proc. IEEE Int'l Conf. Computing, Networking and Communications (ICNC)*, pp. 315–319, 2015.

[22] K. Gerrigagoitia, R. Uribeetxeberriay, U. Zurutuzaz, and I. Arenaza, "Reputation-Based Intrusion Detection System for Wireless Sensor Net- works," In *Proc. IEEE Complexity in Engineering (COMPENG)*, pp. 1-5, June 2012.

[23] L.Ju, H. Li, Y. Liu, W. Xue, K. Li, and Z. Chi, "An Improved Intrusion Detection Scheme based on Weighted Trust Evaluation for Wireless Sensor Networks," In *Proc. IEEE 5th Int'l Conf. Ubiquitous Information Technologies and Applications*, pp. 1-6, 2010.

[24] S. Kaplantzis, A. Shilton, N. Mani, and Y. Sekercioglu, "Detecting Selective Forwarding Attacks in Wireless Sensor Networks using Support Vector Machines," In *3rd Conf. of Intelligent Sensor Networks and Information Processing*, pp. 335-340, 2007.

[25] H. Deng, A. X. Sun, B. Wang, and Y. Cao, "Selective Forwarding Attack Detection using Watermark in Wireless Sensor Networks," In *International Colloquium on Computing, Communications Control, and Management*, pp. 109–119, 2009.

[26] M. Stehlik, V. Matyas, and A. Stetsko, "Towards Better Selective Forwarding And Delay Attacks Detection in Wireless Sensor Networks," In *13th IEEE Int'l Conf. Networking, Sensing, and Control*, pp. 1–6, Apr 28, 2016.

[27] B. Cui, and S. J. Yang, "NRE: Suppress Selective Forwarding Attacks in Wireless Sensor Networks," In *IEEE Conference on Communications and Network Security*, pp. 229–237, 2014.

[28] Y. Zhang, L. Lazos, and W. Jr. Kozma, "AMD: Audit-Based Misbehavior Detection in Wireless Ad Hoc Networks," In *IEEE Transactions on Mobile Computing*, volume 15, no. 8, pp. 1893–1907, Aug. 2016.

[29] Prathap U, P Deepa Shenoy, and Venugopal K R, "CPMTS: Catching Packet Modifiers with Trust Support in Wireless Sensor Networks," In *Proc. IEEE WIECON-ECE*, pp. 255–258, Dec. 2015.

[30] S. Ruj, A. Nayak, and I. Stojmenovic, "Pairwise and Triple Key Distribution in Wireless Sensor Networks with Applications," In *IEEE Transactions on Computers*, volume 62, no. 11, pp. 2224–2237, Nov. 2013.