IMNTV-Identifying Malicious Nodes Using Trust Value in Wireless Sensor Networks

Swathi B H[1]; Megha V[2]; Gururaj H L[3]; Hamsaveni M[4]; Janhavi V[5]
1,2,3,4,5 Vidyavardhaka College of Engineering
(email2swathibharish@gmail.com)

Received (2018-04-23) Accepted (2018-06-21)

Abstract: *Security is the major area of concern in communication channel. Security is very crucial in wireless sensor networks which are deployed in remote environments. Adversary can disrupt the communication within multi hop sensor networks by launching the attack. The common attacks which disrupt the communication of nodes are packet dropping, packet modification, packet fake routing, badmouthing attack and Sybil attack. In this paper we considered these attacks and presented a solution to identify the attacks. Many approaches have been proposed to diminish these attacks, but very few methods can detect these attacks effectively. In this simple scheme, every node selects a parent node to forward the packet towards base station or sink. Each node append its unique identity and trust to the parent as a path marker. It encrypts the bytes using a secret key generated and shared among the sink. The encrypted packet is then forwarded to the parent node. Base station can identify the malicious nodes by using these unique identity and trust value.*

Keywords: *WSNs, Packet modification, Packet Dropping, Packet fake routing, bad mouthing attack, Sybil attack.*

How to cite this article:
Swathi B H, Megha V, Gururaj H L, Hamsaveni M, Janhavi V. IMNTV-Identifying Malicious Nodes Using Trust Value in Wireless Sensor Networks. J. ADV COMP ENG TECHNOL, 4(2) Spring 2018 : 87-100

1.INTRODUCTION

Wireless Sensor Networks(WSNs) consists of limited distributed self-reliant components having sensing, evaluating and communication capabilities. Sensors keep track of environmental conditions like sound, temperature, motion, vibration or pollutants. Sensor network transmits the data from one hop to another hop in an adhoc way and to the destination. That could be a base station, sink or gateway where the data is stored, computed and displayed. Sensor nodes are usually deployed in an unmanned and remote environment[1].

When sensor nodes are deployed in such environment, they are highly prone for wide varieties of attacks. Packet dropping and modification are the basic problems which have got the major impact on the statistical information collected by the sensor nodes. As a result lot of vital sensed data will be lost. Other type of attacks includes injection of false data in the channel, using the identities of genuine node to make other nodes as malicious nodes, replay previously heard packets to the drain the energy of other nodes as battery capacity is essential in nodes. Cryptographic methods alone are not enough to safe guard the data. Attacks like wormhole, rushing attacks can be launched ignoring the cryptographic keys[2][9]. Hence it is important to provide more security for sensitive information.

In this paper, we proposed a scheme 'Identifying Malicious Nodes using Trust Value in Wireless Sensor Networks(IMNTV)' which effectively identifies the packet modifiers and packet droppers. After deployment, each node chooses a list of parent nodes. Each of which consists of equal and shortest distance to sink node. Each node selects a parent node from the list of parent nodes and selection information about the parent node is sent to the sink node. Sink establishes a routing tree rooted at sink node. Data transmission is equally divided among the intervals. Each node selects a different parent node during the initialization stage of a round from the selected parent list. Intermediate node generates marker data which contains node identity and trust factor on its parent node. The marker data is encrypted and added to the packet before forwarding to the parent node. Sink uses the marker data to trace the nodes in the routing path. Based on marker information sink can calculate the dropping ratio for each node. A node categorization algorithm is used to recognize the node as suspiciously bad nodes or bad for sure. Tree structure dynamically changes for every time interval. So that behaviour for the node can be recorded. During packet decryption process, sink identifies a pair of nodes which are responsible for packet modification. If packet decryption fails, uses the trust value to filter the malicious node among the pairs[3][4][5][6][8].This scheme also effectively identifies some common attacks of WSNs such as packet fake routing, badmouthing attack where the intruders collude to present negative feedback on the victim to lower or destroy its reputation and Sybil attack where identity of the genuine node is used by the attackers for getting an illegal entry into a network. These attacks can significantly destroy the performances of the network. The proposed approach IMNTV in this paper provides a solution for identifying the attacks not considering in existing approaches. We provide a simulated performance analysis which shows the comparison among existing approach.

The rest of the paper is organized as follows ,section 2 discusses about literature review and related work, section 3 describes the proposed scheme to identify the malicious node , section 4 gives the performance analysis and section 5 concludes the work and describes the future challenges.

2.LITERATURE REVIEW AND RELATED WORK

In literature many schemes have been proposed to identify the packet droppers and packet modifiers. The below Table 1 shows the techniques used to detect the malicious nodes.

Table 1: Approaches used for detection of malicious nodes

Packet dropping	Packet modification
1. Multipath routing approach	1.Packet filtering
2.Neighbour observation approach	2.Probabilistic nested marking
3.Acknowledgement approach	

Modified messages can be removed from certain number of nodes. So that energy can be saved without transmitting the modified messages.

Packet droppers can be handled by using multipath routing approach, neighbour observation or monitoring approach and acknowledgement approach[16]. Multipath routing approach is wildly adopted measure to avoid packet droppers. In this approach several copies of a packet are forwarded using multiple paths to reach the destination[10][11][12]. Neighbour observation approach is used to detect the packet droppers in WSNs[13][14][15]. The watchdog method is used to monitor the neighbourhood nodes. Each node collects the information about its neighbour node behaviour to detect the malicious activity. Based on this information node takes further forwarding decisions. This method needs to buffer the packets that are forwarded to next node. Then packet droppers can be identified by comparing the forwarded packet by the next node with its buffered packet. This method is prone to false praise attack and bad mouthing attack.[8]

Another approach to find out the packet

dropper is acknowledgement approach. This can be done with receiving response from intermediate nodes. Both multipath routing approach and neighbour observation approach observes each hop during a packet being transmitted. Hence its requires more energy consumption.

In literature, Chuang W et al. proposed a scheme called Catching Packet Droppers and Modifiers (CPDM)[8].This scheme frames the source node even the intermediate node modifies or drops the forwarding packet. CPDM identifies the packet dropper and modifiers with high percentage of false isolation. This scheme does not use multipath routing approach and neighbour observation approach or monitoring approach. But it detects the packet droppers and modifiers after long time operation.

Another method Catching Packet Modifiers with Trust Support in wireless Sensor Networks (CPMTS) is proposed to overcome from those issues. It makes the child node to observe its parent node for successful or unsuccessful transaction. Both the approaches do not detect fake routing, Sybil attack and bad mouthing attack, which impacts the basic packet modifier detection techniques[4][5].

The proposed scheme IMNTS detects the malicious nodes which lead to packet modification, packet dropping, Sybil attack, packet fake routing and bad mouthing attack. It identifies the malicious nodes early with high detection rate and low false detection rate.

3.IMNTV

The proposed Identifying Malicious Nodes using Trust Value in Wireless Sensor Networks method has several steps of operation. The below Fig.1 shows the operations starting from creating a network topology, selection of parent node ,packet forwarding and traffic generation, processing of packet at sink, detecting malicious node, selecting a parent node for next round[5][6][7][8].The loop repeats as long as the nodes are being successfully tested for the malicious node. Once the malicious node detected that node will drop from the path. It will terminate when the transmission of all the nodes be done.

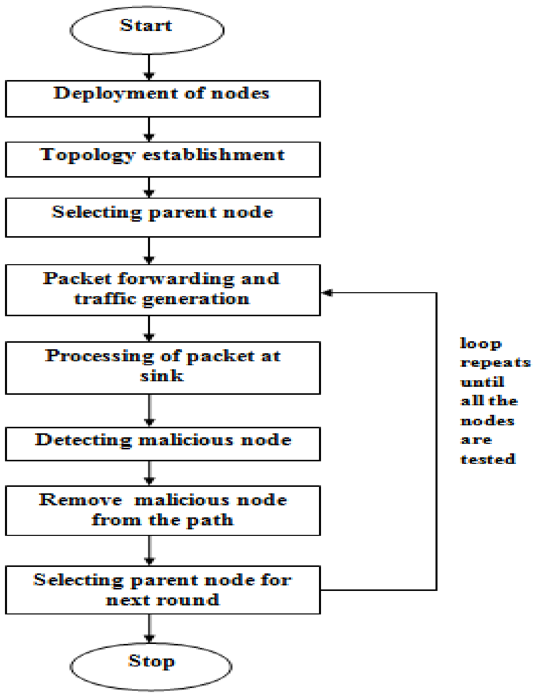
Start
Deployment of nodes
Topology establishment
Selecting parent node
Packet forwarding and traffic generation
Processing of packet at sink
Detecting malicious node
Remove malicious node from the path
Selecting parent node for next round
Stop
loop repeats until all the nodes are tested

Fig.1- Steps of operation

System assumptions: IMNTV assumes that the network is static. The links are bidirectional. Pair wise keys are shared among the nodes and sink. Network consists of n nodes where n=n1,n2,n3... nn. Each node with the trust value Tv, where Tv=Tv1,Tv2,Tv3... Tvn. The entire network can be represented as the sum of every single node along with its trust value which is represented as

$$\sum n * Tv \qquad (1)$$

$$\sum(n1 * Tv1 + n2 * Tv2 \dots . n * Tvn) \qquad (2)$$

It assumed that sender node uses the transmission power level. Hence current forwarder node and next hop node can hear the packet transmission.

3.1 Creating a network topology

Sensor nodes which are deployed, creates a DAG and form a routing tree from the DAG. The below Fig.2 shows the creation of topology. The sink holds the information about DAG ,extracted routing tree and it shares a unique secret key with each node. During packet transmission each node adds a packet sequence number and encrypts the packet using unique secret key. Then it forwards the packet to its randomly chosen parent. Child node notice the parent node for packet successful and unsuccessful transaction ,builds trust value on parent. The trust value is shared with the sink node. The receiving node adds few bits to the received packet as packet marker to identify the forwarding path. A malicious node may drop a packet during the packet transmission. Then the packet received by the sink is decrypted. Hence sink can determine the original packet sender. The packet sequence number is keeps track by the sink node. After n rounds sink can identify the packet dropping ratio for each node. Based on sequence number, knowledge of topology, marker data, trust value and dropping ratio, the sink detects the malicious nodes.

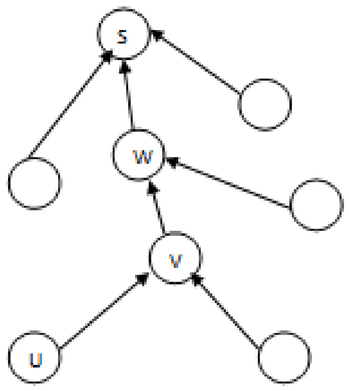

Fig.2 Topology establishment

Sink sends beacon message to nodes in communication range. Each receiving sensor node is loaded with $<K_s, R_r, P_n, N_{seq}>$ where K_s is a unique secret key shared between the sink and the node, R_r is the duration of each round, P_n is the maximum no. of parent list that each child node identifies during the DAG establishment. Each node can picks the random number of parents between the range 0 to P_n and N_{seq} is the maximum sequence number of packets. Each node receiving the beacon message computes all possible paths to reach the sink. Each node sends a report on all computed path to sink upon receiving route reports from all nodes. Sink generates a unique secret key for each node and acknowledge to respective node.

After establishment, base station or sink node sends a tuple <node ID, distance to sink> = <S,0> to all its neighbour node. The tuple contains two fields. First, is the node ID. Here we assume sink ID as 0.Second is the distance from sink to sender node.

1. When receiving the first tuple $<V, d_v>$, node U sets its own distance to the sink as $d_u=d_v+1$.

2. Node U records each node W (which includes node V) as its parent node on the DAG if it has received (W,d_w) where $d_w= d_v$. i,e., node U is recorded as its parents on the DAG. The nodes whose distance from hop to the sink is equal and the distance is one hop less than its own. If the number of parents is greater than P_n, only P_n parents are recorded while others are discarded. The actual number of parents recorded is denoted by $P_{n,u}$.

3. After stipulated time interval, node U broadcasts tuple $<U, d_u>$ to its downstream one-hop neighbours to continue the process of DAG establishment. Then, among the recorded parents on the DAG, node U randomly picks one (whose ID is denoted as P_u) as its parent on the routing tree. Node U also picks a random number (which is denoted as R_u) between 0 and P_{n-1}. Random number Ru is used as a short ID of node U. That is attached to each packet node U. Hence the sink can trace out the forwarding path. Finally, node u sends P_u, R_u and all recorded parents on the DAG to the sink.

3.2 Packet sending ,forwarding and traffic generation

If node U is the source node and if it needs to send certain amount of sensed information to the sink, then node U generates the following packet and forwards to its parent P_u.

$$m=<P_u,\{R_u,U_{id},P_c MOD\ N_{seq},D,pad_{u,0}\}K_{s,u},pad_{u,1}> \quad (3)$$

where R_u is a random number chosen by node U. R_u is used with the packet to identify the packet forwarded path. P_c MOD N_{seq} is the sequence number for a packet. P_c is the counter of the packet which is keep tracked by each node. U_{id} is the unique id of the node. D is the data that is generated by node U.

Padding $pad_{u,0}$ and $pad_{u,1}$ are used to keep the packet size as equal. So that parent node cannot drops a received packet based on packet size. Packet size is depends on the number of hops away from node to sink. If there are h hops between sink and a node, the length of $pad_{u,1}$ is $\log(P_n)*(h-1)$ bits. When a parent receives a packet from one hop from its child node $\log(P_n)$ bits information will be added to the beginning of the packet and $\log(P_n)$ bits will be removed at the end.

The below Fig.3 shows the operation of packet sending and forwarding with respect to Fig.2, where node V is considered as source node.

The maximum length of the packet is P_l bits, length of a node ID is Id_l bits and data length is D_l bits. $Pad_{u,0}$ should be P_l-Id_l*2-$\log(P_n)*h$-$\log(N_{seq})$-D_l bits, where Id_l*2 bits are for P_u, R_u is $\log(P_n)$ bits, $pad_{u,1}$ is $\log(P_n)*(h-1)$ bits and P_c MOD N_{seq} is $\log(N_{seq})$ bits long. Padding $pad_{u,0}$ to this values tells that each packet in the network has the same length P_l.

Packet m is encrypted by using key $K_{s,u}$ and sends the packet to node V. Node V act as an intermediate between node W and node U. When node V gets the packet<V,m> ,it forms the packet $<P_v,\{R_v,m^1\}K_{s,v}>$ and forwards to its parent node P_v. Here R_v, with $\log(P_n)$ bits are appended to the front of m^1 and removed the $\log(P_n)$ bits from right most of m. Hence packet size can be maintained.

It generates marker information and trust value of node V on parent W. Node V also uses its secret key $K_{s,v}$ to create m^2. In the same way all nodes add the encrypted marker information to the data packet during forwarding.

3.3 Processing of packet at sink

Sink is denoted as node 0. When sink node receives a forwarded packet $<0,m^1>$, it performs the following steps.

1.Let U=0 and $m=m^1$ where U and m are two temporary variables.

2.The sink node tries to identify the child of node U(denotes as V). So that decrypt($K_{s,v}$,m) produces a string starting from R_v, where m is decrypted with key $K_{s,v}$. Then it tries to match with the marker information.

3.If this attempt fails to match the marker information for all children of node U, the packet is marked as modified and that should be dropped.

4.If the attempt succeed by matching the

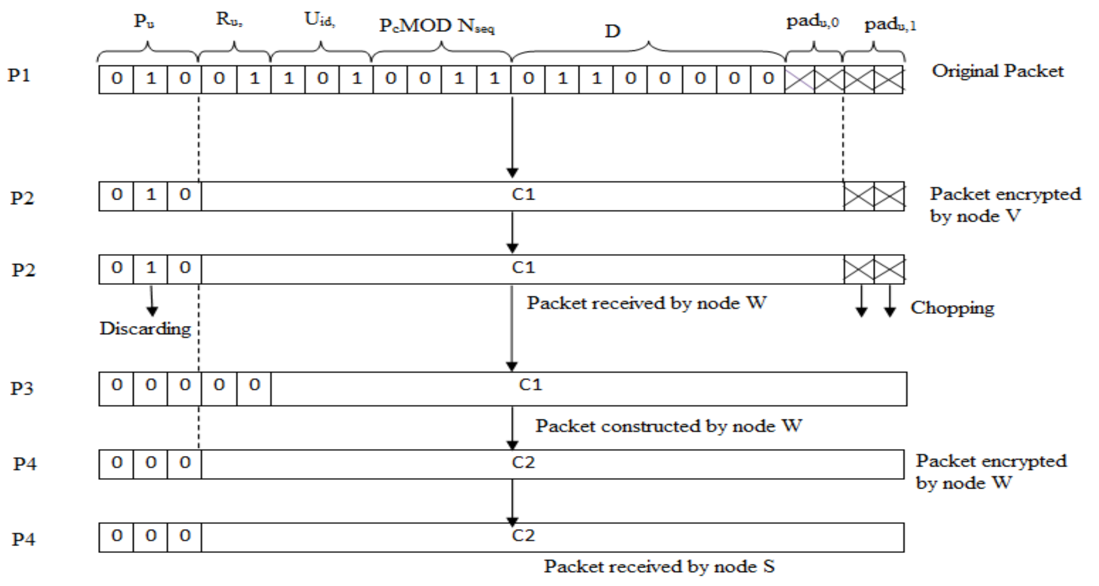

Fig.3 Example for packet sending and forwarding

marker information, then it indicates that node V is the child node and the packet was forwarded from node V to node U.

Then there exists two cases.

a. If decrypt($K_{s,v}$,m) begins with <R_v,V> then it shows that node V as the original packet sender. The packet sequence number is recorded.

b. Otherwise, it shows the node V as an intermediate packet forwarder. Then node U is updated to be node V,m is updated to be the string that is obtained by removing R_v from leftmost. Then, it repeats step 2-4.

3.4 Node Categorization Algorithm

In each round, sink records the packet information. For a sensor node U, it records the total number of packets sent, sequence number of those packets. After the completion of each round, the sink node calculates the ratio of dropping packet. If $n_{u,f}$ is the number of packets that are forwarded and $n_{u,rec}$ is the number of packet received. Then the dropping ratio of each round is calculated as follows.

$$dropping\ ratio(d_u) = \frac{((n_{u,f} - n_{u,rec} * n_{u,f}))}{n_{u,f} + n_{u,rec} + (n_{u,f} * n_{u,f} - n_{u,rec})} \quad (4)$$

By using the knowledge of tree topology and dropping ratio the sink detects the nodes as droppers for sure and suspiciously droppers. Hence a threshold α is used. The dropping ratio of the node should be lower than α, if the node doesn't drop the packet intentionally. If the packets are dropped by collision then α should be greater than 0.

The below Fig.4 shows the node status pattern from a leaf node to the sink node. We mark node with "-" if the dropping ratio is greater than α, otherwise with "+".

Algorithm 1 Packet receiving at the sink

```
1  Input:packet<0,m>
2  U=0,m^1=m; successAttempt=false;
3  for each child node V of node U do
4        P=dec(K_{s,v},m^1);
5        if the attempt fails to decrypt then
6           continue;
7        else
8           successAttempt=true;
9           if P begins with <R_v,V>then
10             record the packet sequence number ,mark V as sender;
11             break;
11          else
12             remove R_v from P and obtain m^1,mark V as a forwarder node
13             U=V, successAttempt=false;
14             goto line 3;
15 if successAttempt=false then
16          drops the packet;
```

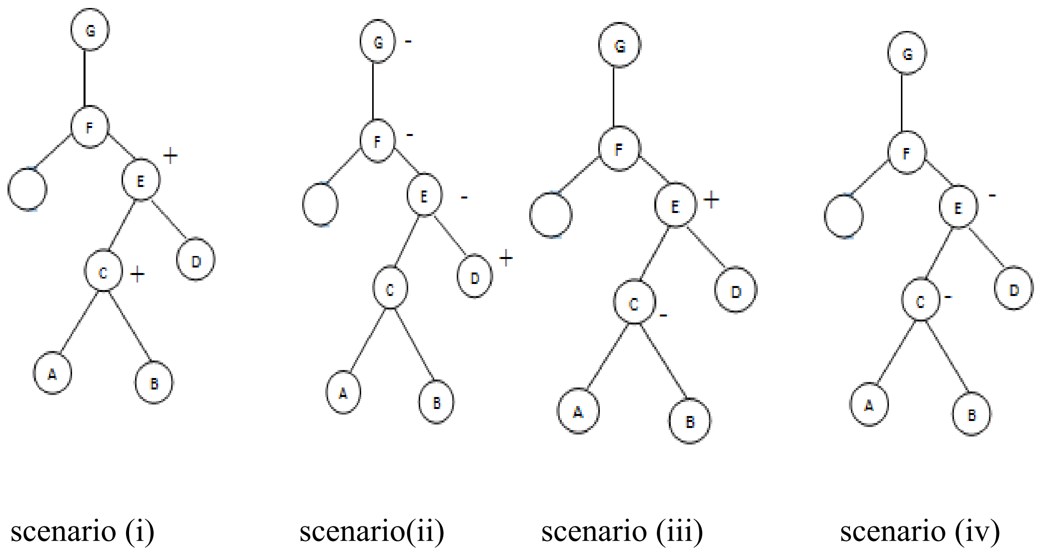

Fig.4 -Node's status pattern

We mark each node in a path from leaf node to the sink by using the basic combination of node's mark pattern.

scenario(i): +{+} Both child and parent nodes are marked as "+".The child node and its parent node do not drop packets on their path. But they many drop packets on other forwarding paths. Hence the sink node decides that these nodes are temporarily good. For example in the above scenario (i) node C and node E are marked with "+" which shows them as temporarily good. There is a special issue where, if child node is marked with "+". It ensures that, it cannot drop other's packet.

scenario(ii): +-{-}* A node is marked as "+" and its one or more nodes are marked as "-". The above scenario (ii) illustrate that node C is marked with "+" and node E, F and G are marked with "-". If this conclusion is incorrect and node E is good, E should not drop its own packets. Since node E is marked with "-", there must be upstream nodes of E which is dropping node E's packets. The bad upstream nodes are at least one hop above E ad at least two hops above the node C. It is not possible for them to differentiate packet from node E and node C. Hence they cannot drop the packets from E during packet transmission from node C. Each packet from node C must forward through node E. Node E encrypts the packet and then forwards to the next upstream. Hence the bad upstream node cannot determine the packet to selectively drop the packets. If the packet is forwarded to the bad upstream node without forwarding through node E, Then the packet cannot be decrypted properly by the sink and that will be dropped. There for node E must be bad and we can also conclude that node F and G are also bad for sure.

scenario(iii): -{+} A node is marked as "-" and parent node as "+".In this scenario either the child node marked with "-" or its parent marked with "+" must be bad. But it can't be further inferred whether a) only the node marked as "+" is bad b) only the node marked as "-" is bad c) both the nodes are bad. Therefore, both nodes are suspiciously bad. Scenario (iii) shows that node C is marked with "-" and node E is marked with "+". If both node C and node E are good and there must exist at least one upstream node of E which drops the packets sent by node C. However, it is not possible to determine such an upstream node since node F and G and other upstream nodes cannot selectively drop packets from C while forwarding packet from E. Hence , either C is bad or E is bad in this scenario.

scenario(iv): -{-} Both parent and child is marked as "-" could be good or bad and they have to be considered as suspiciously bad. Specifically, if node V is the node at highest level that is marked with "-" and U is its parent node. If node U is the sink, node V must be bad for sure. Otherwise, both U and V are suspiciously bad. If the dropping ratio of node U is larger than that of node V, node U is bad for sure. Otherwise both node U and node V are suspiciously bad.

For each cases we can conclude whether a node has dropped packets called as bad for sure, or is predicted to be malicious and have dropped the packets, or to be temporarily good and finally the packets might not have been dropped such packets are called good for sure.

3.5 Detecting bad nodes from suspicious bad nodes

After the completion of each round of traffic generation the sink calculates the each node's dropping ratio and also trust on a parent from their child nodes. Then runs the node categorization algorithm. It identifies the nodes as bad for sure or as suspiciously bad nodes. If the number of suspiciously bad nodes are larger in size then we can identify most likely bad nodes. If the sink contains a list of <parent,child> as suspiciously bad nodes, we call it as suspicious pair. For each round i, all suspiciously identified pairs are listed in a suspicious set which is denoted as follows.

$S_i=\{<U_j,V_j>|<U_j,V_j>$ and $<U_j,V_j>=<V_j,U_j>\}$ is a suspicious pair of nodes.

Algorithm 2 Node Categorization Algorithm

```
Input: Tree T, with each node U marked by "+" or "-" and $d_u$ is the dropping ratio.
for each leaf node U in T do
    V=U's parent;
    while U is not the sink do
        if U.mark = "+" then
            if V.mark ="-" then
                b=V;
                repeat
                    e=V;V=V's parent node;
                until V.mark="+" or V is sink, set nodes from b to e as bad nodes;
            else
                if V is  sink then
                    Set U as bad for sure;
                if V.mark="+" then
                if V is not bad for sure then
                    Set U and V as suspiciously bad nodes;
            else
                if $d_v-d_u> \alpha$ then
                    Set V as bad node;
                else if $d_u-d_v> \alpha$ then
                    Set U and V as suspiciously bad nodes;
    U=V,V=V's parent node
```

After the examination of n rounds, we can get number of suspicious sets $\{S_1,S_2,S_3.....S_n\}$.Then S^1 can be defined as most likely bad nodes for the set of suspicious nodes, if it has the following properties.

Minimality: The size of set S^1 should be small to minimize the number of nodes that are falsely accuse as innocent.

Most-likeliness: After n rounds $\forall<U,V> \in S_i(i=1,2,3.....n)$ if $U \in S^1$ but V does not belongs to S1, then U must have higher probability as bad node the node V.

Coverage: For any suspicious pair, there must be at least one of the node among the pair in the set of most likely bad nodes. i.e., $\forall<U,V>\in S_i(i=1,2,3....n)$ and it must holds either U belongs to S^1 or V belongs to S^1.

To find the malicious node sink performs the following on each parent node. Sink calculates the average trust value by using the trust value that is received from each child node. If the average value of trust is less than the threshold value(α), then parent node is the malicious node. If the average value of trust is greater than α, then find a child whose average trust value less than the α. If it is successful to find such child node, then that child node is the malicious node. If average trust values of both children and parent are greater that the threshold, then they are considered as a suspicious pairs but not malicious yet.

3.6 Changing parent for next round

During malicious node identification phase traffic generation is carried in equal duration. After the completion of a round child chooses the next parent by checking its parent list. Then it chooses a parent with which it never had an iteration. In other way child chooses the parent node by considering the highest trust value.

Algorithm 3 Detecting malicious node

Notations: PId,CId: Indicates ParentId and ChildId respectively.
AvgTrustValue: It is a function, which calculate the average trust value from all children.
α: pre-defined threshold value.
SPair: Set of tuple<PId,CId> which are identified as suspiciously bad.

```
1 for each SPair in SPair do
2     if AvgTrustValue (SPair.PId) < α then
3         SPair.PId is the malicious node;
4     else if AvgTrustValue (SPair.CId) < α then
5         SPair.CId is the malicious node;
6     else
7         do nothing
8 /*If both parent and child are still suspicious, handle more packet in next round*/
```

```
Algorithm 4   Changing parent for next round

Notations: Selected: boolean value
           PIds: Set of parent node Ids
           PID: Selected parent Id in current round
           GParentIds: Parent Ids whose trust value is greater than threshold value

1  Selected=false;
2  for each ID in PIds do
3      if ID was never had an interaction then
4      selected=true;
5      PID=ID;
6      break;
7  if Selected==false then
8     for each ID in PIds do
9       if TrustValue of ID>=Threshold then
10        add ID to GParent Ids;
11 PID=Random(GParentIds);
```

```
Algorithm 5   Detection of Packet modification

Notations: Packets p, next node n
1 U keeps p in buffer;
2 V->n forwarded packet will be traced;
3 if V(p)=U(p) then
4     no modifications in the forwarded packet;
5 else
6 modification has happened, Parent V has changed the packet and reduces the trust
  value;/*SUB_TRUST ID*/
7 if V(p)≠U(p) then
8 U+V=suspicious pairs list;
```

3.7 Analysis of various security attack

The proposed approach IMNTV identifies the various attacks, which disrupt the communication in WSNs.

Each node say U in Fig.2 forwards the packet to parent V and observes V, till V forwards the packet to next hop node W.

Packet Modification: Node U keeps the packet in the buffer till parent V forwards the packet to next hop and listens to the packet that V forwards. U compares the packet forwarded by V with the packet in buffer. If there is any modification in the packets forwarded by V then U determines that the parent V has changed the packet and reduces the trust accordingly. The sink adds both parent and the child into the suspicious pair list when the packet decryption fails.

Packet Dropping: Node U keeps the packet in the buffer, if node U does not hear the forwarding from parent V in the determined timeout then U determines the packet dropping from V and reduces the trust on parent V.

Algorithm 6 Detection of Packet Dropping

```
1  U keeps p in buffer;
2  if U does not hear packet forwarding from V in given time out then
3      U determines packet dropping has occurred from V;
4      V=SUB_TRUST ID;
```

Algorithm 7 Detection of Packet Misrouting

```
1  U initialize its next hop as V;
2  V initialize its next hop as W;
3  U maintains next hop details W until  V transfers packet to W;
4  U determine whether V has transferred packet to selected parent node W;
5  if id recorded= id transferred then
6      no misrouting; /*ADD_TRUST ID*/
7  else misrouting /*SUB_TRUST ID*/
8      add  it to suspicious pair list;
```

Algorithm 8 Detection of Sybil attack

```
1  if provided marker information=recorded marker information then
2      authentication successful;
3  else
4      authentication failure;
5  add child node to suspicious pair list;
```

Packet Misrouting: In this form of attack the node forwards the packet to the unintended next hop node. Before starting the traffic generation of that cycle each node announces the parent node information with one hop neighbour nodes. In figure 3, when V announces its selected parent W to one hop neighbour nodes, node U maintains the next hop node W of the selected parent V in the memory along with selected parents list. Node U on comparing the next hop node id W to the node id to which the V forwarded determines whether the packets are being misrouted or not and determines the trust values. The packet decryption also fails and adds the valid node to suspicious pair list.

Sybil Attack: A node uses unauthenticated or the other authenticated identity to frame other node as malicious node. In IMNTV approach while adding the marker information, malicious node can add wrong identity, the packet description method at the sink determines whether the marker matches with any of the children at the same level and add the nodes to suspicious pair list.

Bad Mouthing Attack: Though low trust on the parent with sink is shared by the node, sink consider the mean trust from all children to suppress the bad mouthing attack .i.e., false statement about the trust of the node is reduced.

$$average = \frac{\sum Trust\ value\ from\ all\ children}{No.of\ children} \quad (5)$$

4.PERFORMANCE ANALYSIS

The coherence and productiveness of the IMNTV are estimated using NS-3 simulator. The analogize of proposed method with CPDM and CPMTS is being done in this paper. The performance of both CPMTS and CPDM degrades with the insertion of misrouting attack in malicious nodes. The Table 2 shows the simulation parameters.

Table 2: Simulation Parameters

Simulator Used	NS-3
Compared Methods	CPDM and CPMTS
Number of Static Nodes Considered	100 nodes
Protocol Installed	802.15.4 MAC
Delay In The Channel	2 milli second
Generation of Packet Per Node	50 packets
Malicious Node Consideration	Non leaf nodes

4.1 Percentage of Detection

The number of malicious nodes are considered to be 15,25,35,and 45 out of 100 nodes in a network. As illustrated in the Fig.5, percentage of detection is improved in IMNTV than in CPDM and CPMTS methods. It is said that IMNTV is efficient as it can handle misrouting attack. Let Z be the total number of malicious node in a network and z be the malicious nodes detected. Then,

$$Percentage\ of\ detection = \frac{\sum z}{\sum Z} * 100 \quad (6)$$

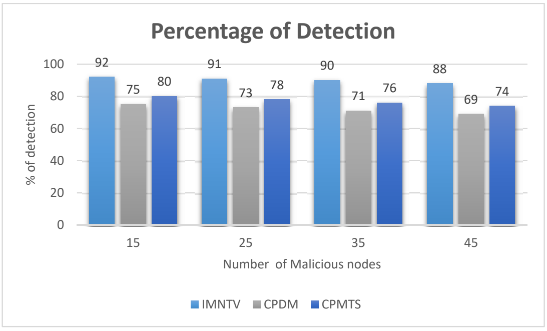

Fig.5: Percentage of Detection

4.2 Percentage of False Isolation

The number of malicious nodes are considered to be 15,25,35 and 45 out of 100 nodes in a network. As illustrated in the Fig-6,percentage of false detection is high in CPDM approach. CPMTS reduces the false isolation compare to CPDM, but false isolation increases on injecting misrouting attack. In the proposed method, only the current parent and children nodes where the packet decryption fails are considered for identifying the malicious node. Let Y be the number of genuine nodes in a network and y be the number of genuine nodes isolated. Then,

$$Persentage\ of\ false\ isolation = \frac{\sum y}{\sum Y} * 100 \quad (7)$$

The trust consideration is done based on the trust values from all children node to avoid the bad mouthing attack from specified child which portrays the parent as suspiciously malicious by updating false trust value.

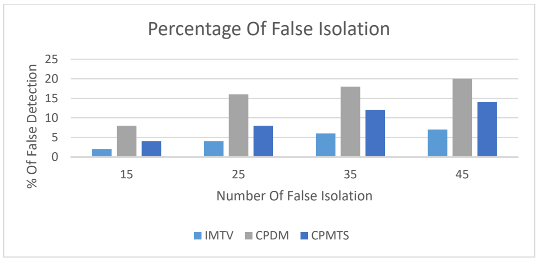

Fig.6 : Percentage of false isolation

4.3 Early Detection Rate

In all the considered method i.e, CPDM, CPMTS and IMNTV traffic is generated in multiple rounds of equal duration and tries to find the malicious nodes after each round. As illustrated in the Fig.7 IMNTV detects the malicious nodes early compared to other two methods.

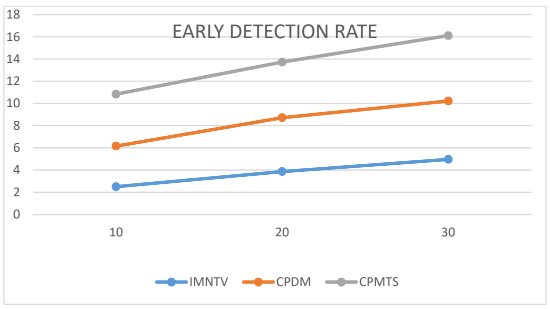

Fig.7 Detection rate of malicious node

5.CONCLUSION

In WSNs, the sensor nodes which are interrupted by the intruder can perform the malicious activities and disrupt the communication between nodes. These malicious activities can drops the valid data or inject the false data during packet transmission. We propose a effective method to identify the security problem such as packet modification, packet dropping, fake routing, bad mouthing attack and sybil attack which uses wrong identity. IMNTV begins with creating a tree topology from DAG. Sink holds the information about parent-child relationship. Each child node chooses its parent at the beginning of each round. Each packet is padded and encrypted with a secret key. The packet is added with trust value and other small number of extra bit. So that sink can identify the original sender and packet dropping ratio of each node. At the end of each round, IMNTV tries to identify the bad nodes. Performance analysis shows that IMNTV identifies the malicious nodes with low false detection and with early detection rate. It also identifies fake routing, using fake identity, bad mouthing attack. In future IMNTV can be further improved to avoid black hole attack, transmission power control attack.

REFERENCES

1. GhazalehTahandowt, FatehmanGhasseuri, An adaptive sinkhole algorithm in wireless neural networks- Elsevier2017.

2. A.R Dhalene and P.N Chatur, Detailed Survey on attacks in wireless sensor network, Proceedings of the International conference on data engineering and communication technology. Advances in Intelligent systems and computing 469, 2017.

3. BharathBhushan, Gadharsahoo, Recent advances in attacks, Technical challenges, Vulnerabilities and their countermeasures in Wireless sensor networks, 2017.

4. Noor Alsaedi, Fazirul Hisyam Hashim, A.Sali, Fakheul Z Rokhani- Detecting Sybil attacks in clustered wireless sensor network based on energy test system(ETS)- Elsevier 2017.

5. PrathapU,Deepa ShenoyP,Venugopal K P,"CMNTS:Catching Malicious Nodes with Trust Support in Wireless Sensor Networks",IEEE Region 10 symposium ,December 2016

6. PrathapU,DeepaShenoyP,Venugopal K P,"CPMTS:Catching Packet Modifiers with Trust Support in Wireless Sensor Networks",IEEE International WIE Conference on Electronical and Computer Engineering, December 2015.

7. Nirupama A.S., Ch.VijayaBhaskar and Sudarson Jena, An Efficient Protocol to Identify Packet Droppers and Modifiers to Improve QoS in Wireless Sensor Network(WSN) IEEE ICCSP conference 2015.

8. Chuang W, Taiming F, Jinsook K, Guiling W, and Wensheng Z, "Catching Packet Droppers and Modifiers in Wireless Sensor Networks," In IEEE Transactions on Parallel and DistributedSystems, volume 23, pages 835–843, May 2012.

9. Issa M. Khalil, "ELMO: Energy Aware Local Monitoring inSensor Networks," In IEEE Transactions on dependable andsecure computing, volume 8, pages 523–536, August 2011.

10. R. Mavropodi, P. Kotzanikolaou, and C. Douligeris, "SECMRa Secure Multipath Routing Protocol for Ad Hoc Networks," In Ad Hoc Networks, volume 5, pages 87–99, 2007.

11. M. Kefayati, H.R. Rabiee, S.G. Miremadi, and A. Khonsari, "Misbehavior Resilient Multi-Path Data Transmission in Mobile Ad-Hoc Networks," In Proc. Fourth ACM Workshop Security ofAd Hoc and Sensor Networks (SASN 06), 2006.

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

13. F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical En-Route Filtering of Injected False Data in Sensor Networks," In Proc.IEEE INFOCOM,, 2004.

14. S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An Interleaved Hopby- Hop Authentication Scheme for Filtering False Data in Sensor Networks," In Proc. IEEE Symp. Security and Privacy, 2004.

15. H. Chan and A. Perrig, "Security and Privacy in Sensor Networks," In Computer, volume 36, pages 103–105, Oct 2003.

16. C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," In Proc. IEEE FirstIntl Workshop Sensor Network Protocols and Applications, 2003.