International Journal of Computer Applications (0975 - 8887)
Volume 121 - No. 9, July 2015

Real Time Intrusion and Wormhole Attack Detection in Internet of Things

Pavan Pongle
Department of computer Engineering
Sinhgad College of Engineering
Pune, India

Gurunath Chavan
Department of computer Engineering
Sinhgad College of Engineering
Pune, India

ABSTRACT

There are currently more objects connected to the Internet than people in the world. This gap will continue to grow, as more objects gain the ability to directly interface with the Internet. Providing security in IoT is challenging as the devices are resource constrained, the communication links are lossy, and the devices use a set of novel IoT technologies such as RPL and 6LoWPAN. Due to this it is easy to attack in IoT network. The proposed system is a novel intrusion detection system for the IoT, which is capable of detecting Wormhole attack and attacker. The proposed methods uses the location information of node and neighbor information to identify the Wormhole attack and received signal strength to identify attacker nodes. Design of such system will help in securing the IoT network and may prevents such attacks. This method is very energy efficient and only takes fixed number of UDP packets for attack detection, hence it is beneficial for resource constrained environment.

Keywords

Intrusion Detection, Internet of Things, RPL, Wormhole, Packet Relay, Encapsulation, RSSI

1. INTRODUCTION

Internet of Things (IoT) is a fast-growing innovation that will greatly change the way humans live. It can be thought of as the next big step in Internet technology. The changing operating environment associated with the Internet of Things represents considerable impact to the attack surface and threat environment of the Internet and Internet-connected systems. IoT is heterogeneous system consisting of various types of sensors nodes or devices with different kind of technology at each layer. However, due to the limited address space of IPv4, objects in the IoT uses IPv6 to accommodate space in Internet. Objects in the IoT can be devices with sensory capabilities, smart metering, health care sensor etc.

RPL (Routing Protocol for low power and Lossy network) [1] is routing protocol used at the network layer in IoT. RPL topology contains one root/sink node directly connected to Internet using 6BR (IPv6 Border Router). RPL topology forms the DODAG (Destination Oriented Directed Acyclic Graph) tree, which contain only 1 root. Root node starts the formation of the topology by broadcasting the DIO (DODAG Information Object) messages. Nodes receiving the DIO message selects the parent to sender by replying DAO (Destination Advertisement Object) message asking can I join you? Parent node gives the permission to join by sending DIO ACK message as yes you can join me. The rank value calculated with respect to the parents rank value and other parameters. The rank value may be depend on the distance from the root node, energy of link etc. The network owner can decide the rank value calculation parameters. If new node want to join the network it first ask is there any DODAG here? By sending DIS (DODAG Info solicitation) message. The nodes continue to broadcast the DIO message and form the tree topology. Fig. 1 shows the comparison of protocols used at traditional IP network and IoT.

IoT Protocol stack	Layers	Traditional IP Protocol Stack
CoAP	Application Layer	HTTP, FTP, SMTP etc.
UDP	Transport Layer	TCP/UDP
IPv6 + RPL 6LoWPAN	Network Layer	IPv4/IPv6 + AODV/DSR/DSDV etc.
802.15.4	Link Layer	802.3/802.11

Fig. 1. Protocols used at traditional IP network and IoT

The rest of the paper is organized as follows: Section II discuss the related work in Wormhole attack detection techniques and IDS systems designed for IoT. Section III gives discussion on architecture of system, modules and algorithm used for detecting attack. In section IV we have discussed the algorithms used to detect the attack and design of wormhole attacker node. Section V is on discussion of how the attacks are detected using proposed system with example. Section VI on evaluation of system using various parameter. Section VII gives the future work and extension for proposed system. Section VIII concludes the work done.

2. RELATED WORK

2.1 Wormhole Attack

RPL can undergo the wormhole attack [2]. The main purpose of this attack is Disrupt the network topology and traffic flow. This attack can takes place by creating tunnel between the two attackers and transmitting the all traffic through it. Wormhole attack tackled in this methods are as,

2.1.1 Wormhole using packet Encapsulation. In this type, attacker node encapsulate the packet in payload and send it to other colleague node, where other attacker takeout the packet from payload and transmit again.

2.1.2 wormhole using Packet Relay. In this mode of the wormhole attack, a malicious node relays packets between two distant nodes to convince them that they are neighbors. It can be launched by even one malicious node.

Wormhole attack detection techniques broadly classify into the following different types as. Hardware Based techniques require the use of extra hardware such as GPS hardware or some specialized nodes. In Clock Based techniques the nodes to have tightly synchronized clocks so that they are able to detect any anomalies in the network. Packet Leashes Based techniques limits the journey of packets across the network beyond a certain limit (either distance or time). RTT Based techniques use the Round Trip time for the detection of wormhole attacks present along a path. Neighbor Discovery/Verification Based techniques uses neighbor/network information for the detection of wormhole attacks. This may either involve verification from neighbors, neighbor information or neighbor monitoring to detect the wormhole attack.

Statistical Analysis based approach (SAM) [3] monitors the occurrence of links returned for a particular destination in multi path protocols for detection of wormhole link. DELPHI [4], Delay Per Hop Indication. They observe delay per hop from source to destination for different paths for wormhole detection. [5] Proposed the use of Directional antennas for handling of wormhole attacks. Their main motive is to avoid nodes that give incorrect location information by installing directional antennas. E2SIW (Energy Efficient Scheme Immune to Wormhole attacks) [6] for the prevention of wormhole attacks, uses location information received from the GPS hardware. The approach only tries to prevent the wormhole attack. It doesn't take into consideration the detection of the wormhole nodes and their punishment.

Raju et al. [7] proposed the use of Average One hop RTT to calculate average time of larger paths to avoid wormhole links. If a link has taken more time than the average RTT times hops of the link, it is considered as suspicious and is not used for further communications. The approach is likely to fail when the attackers are connected via a high speed link or there is congestion in network hence generating false alarms. Yifeng Zhou et al. [8] proposed, a technique for detection of wormhole attacks based on distance verification is proposed for mobile ad hoc network (MANETs) applications. A node estimates its distances to a sender node based on the received signal strength (RSS) of received packets, and uses them to verify against the distances computed from the location information in the packets. The details about attacks on RPL and 6LoWPAN and their measures on it are well discussed in [9].

2.2 IoT and IDS

The table 1 shows the existence IDS system designed for IoT. Only the IDS [2] is evaluated for detection of attacks, rest IDS are proposed frameworks and detecting simple RPL specific attacks. The existing IDS system does not detects the complex attacks such as Wormhole, Blackhole, Sybil and Clone ID attack.

Table 1. Comparison of IDS Systems

IDS	Method	Attack detection	Placement
RIDES[10]	Signature based IDS, uses Bloom filter for signature matching	No	Hybrid
[11]	Network based DOS detection IDS architecture on project ebbits	DOS	Hybrid
[12]	Finite state machine based IDS system	Rank and local repair	Distributed
SVELTE[13]	Host based IDS, construction of network topology at 6BR system	Sinkhole, DODAG inconsistency, Rank, selective forwarding	Hybrid
[14]	Complex event processing IDS, uses EPL and SQL to define attack pattern	No	Centralized

Motivation behind this work is as per now, there is solution against the wormhole attack in RPL based IoT environment yet proposed. This could be step towards the design of such system for detecting and identifying of wormhole attack and attackers.

3. PROPOSED SYSTEM

Proposed system is an novel Intrusion detection system (IDS)[1] based wormhole attack detection system for resource constrained devices.

The attack always brings the abnormal changes in network. Every attack leaves its symptoms on system, from which we can conclude that attack occurred and what kind of that attack was. So we assumed the hypothesis as "more number of neighbor gets formed after attack has been triggered and all new neighbors are from other end of wormhole tunnel". If neighbor is not in transmission range of node then this is due to attack only. During the attack lots of control packets are going to exchange form one end of tunnel to other in that neighbor advertisement, neighbor solicitation and DIO helps in formation of neighbors beyond the transmission range.

3.1 Architecture

The architecture of IDS is shown in Fig 2 consist of the sensor network connected to Internet using IPv6 border router (6BR). The placement for IDS system uses hybrid approach, in which centralized modules on 6BR and Distributed modules on the sensor nodes cooperates to detect attack. We considered the static topology, and

[1]For source code contact pavanpongle@gmail.com or visit at https://github.com/pavanpongle/IoT-Wormhole-IDS

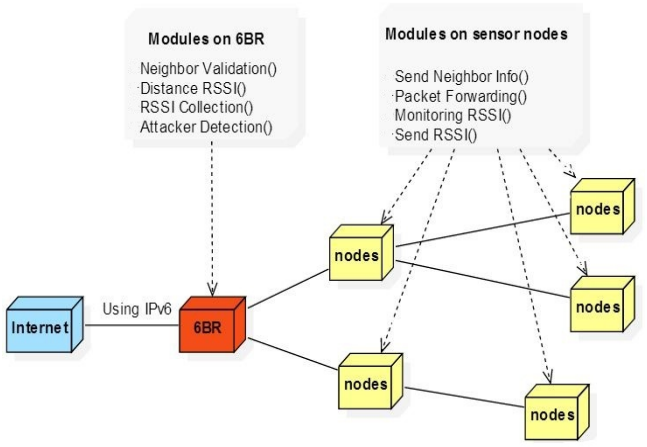

Fig. 2. Architecture of the system

location information of all sensor node known at deployment time at 6BR and during network initialization period there is no attack in network. The working of modules are discussed below,

3.2 Centralized modules

3.2.1 Neighbor Validation. In this module we are collecting the neighbor information from all sensor nodes storing them. The stored neighbors information are verified based on the distance between the node and that neighbor. If the distance found to be more than transmission range of node then this module send the victim packet to the information sender node and to the neighbor whose distance is more than transmission range. The victim packet is shown in Fig 3, first field is code (16 bit) to identifying victim packet at application layer, second field is destination node host ID to which this packet is prepared. and last one is host ID of neighbor which is another victim node. Packet size is 4 Bytes at application layer, if we takes host id as unsigned integer Byte.

Code (3) 2 Bytes	Destination Node ID 1 Byte	Other Victim Colleague Node ID 1 Byte

Fig. 3. Victim packet structure

3.2.2 Distance RSSI. This module calculates the distance between two geographical coordinates. It also convert the RSSI (Received Signal Strength Indicator) value to distance and vice versa. This module provide access to location, range information about each node. For the simulation purpose we have considered the node's range as 100 m. We recorded RSSI value for each meter and stored in 1 dimensional array. This avoids the re-calculation of distance from the RSSI value and vice versa. We have assumed that at initial 5 min are given to network initialization and there is no attack in this short period. In this period we are calculating the distance between each node and storing in 2 dimensional array. For simulation purpose, taking geographical co-ordinates is difficult so we have used the node's x,y position obtained from simulator.

3.2.3 RSSI collection. After detecting the attack, and sending the attack packet (victim packet) to victim nodes, This module wait for the period until the nodes finishes victim packet transmission and records the RSSI value form other victim colleague and send it to 6BR. The received RSSI values are from the the two victim nodes and other nodes in the range of attacker node. The duplicate RSSI received packets from same node are discarded by comparing that node already sent the RSSI value.

3.2.4 Attacker detection. This module process on received RSSI value to find the attacker node. Using RSSI to distance d it states the nodes in that range d, and list the nodes with probability of having attacker nodes.

3.3 Distributed Modules

3.3.1 Send Neighbor info. In network initialization period this module stores the initial neighbor as original neighbors before attack. For each periodic time, if node found the change in neighbor numbers are more than previous then, It send the neighbor information packet as shown in Fig 4 to 6BR. This packets is sent to root node by broadcasting and through the existing route. The code 2 indicate that it is neighbor information packet. The sequence number is to avoid re-processing of the same packet which is received from other node due to packet forwarding nature of algorithm. The second and fourth fields are host id's of information owned nodes and packet forwarder node respectively. This forwarder and sender fields avoids the repeated forwarding of packet from both owned and sender node. For keeping neighbor count nbr_count field added (16 bit) short int, and followed by the neighbors host ID's (8 bit unsigned integer). The information size calculated as (8 + nbr_count) Bytes.

Code (2) 2 Bytes	Sender Node ID 1 Byte	Message Instance 2 Bytes	Forwarding Node ID 1 Byte	Number of Neighbors 2 Bytes	Nbr1 1 Byte	Nbr2 1 Byte	....

Fig. 4. Neighbor information packet structure

3.3.2 Packet Forwarding. Due to UDP protocol at transport layer, there is no guaranty of packet delivery. To assure packet delivery to root node, other nodes helps the sender node in forwarding packet. Initially the important packets are send through default route and another by the broadcasting the packet. In between these transmission, the pause for some second (2 in experimentation) are taken to avoid the packet loss due to collision and buffer over flow. When other node receives the broadcast packets (nbr_info, RSSI_val) it sends this packets to root nodes through its own default route. If the node receives the victim_forward packet from root node it local unicast to the destination node. The packet structure of victim_forward packets are shown in Fig 5. and the structure of RSSI_val, nbr_info packets are same as the shown in Fig 6 and Fig. 4 respectively.

Code (4) 2 Bytes	Destination Node ID 1 Byte	Other Victim Colleague Node ID 1 Byte

Fig. 5. Victim forwarding packet structure

3.3.3 Monitoring RSSI. When node receives the victim packet either from the root node or from broadcasting, it initiate the monitoring process. Node receiving victim packet if it found its own ID at destination place (second field), and other victim colleague in third field. it prepare victim broadcast packet containing destination field as other victim colleague ID and at third field its own id (interchange the two IDs). This two node records the each others RSSI value receiving from broadcast victim packets. and other node records the RSSI value as mentioned in algorithm monitoring_algorithm. The two victim node broadcast the N victim packets to locate the attacker node.

3.3.4 Send RSSI. After broadcasting the n victim packets the recorded RSSI value must reach to the sink node for attacker node detection, so due to unreliable UDP protocol we are sending the RSSI packets repeatedly by unicast, broadcast and through the default route. The each node waits for time until all node finishes victim packet transmission, and send RSSI packets by taking pause of fixed interval between each successive RSSI packet send. The packet structure is shown in Fig. 6. Here first field is code 5 for RSSI packet, second the RSSI value recored by node and third is to which node's it had recorded. and followed by three RSSI values.

Code (5) 2 Bytes	Sender Node ID 1 Byte	RSSI value 1 1 Byte	RSSI value 2 1 Byte	RSSI value 3 1 Byte

Fig. 6. Sending RSSI value packet structure

4. METHODOLOGY AND ALGORITHMS

4.1 Algorithms

4.1.1 Algorithm for detection of wormhole attack on sensor nodes

(1) For every node N do
Wait for settlement of the network
(2) After network initialization, stores the current neighbors as original neighbors
(3) For every periodic time do
Check whether there is change in neighbors
If change in neighbors found then
Send nbr_info (neighbor information) to 6BR through broadcast and default route
(4) If node receives the victim packet then
Initiate monitoring algorithm
(5) If node finishes the recording of RSSI value then
Broadcasts and unicast the RSSI_value packet to reach to sink node multiple times
(6) If node receives broadcast nbr_info and RSSI_value packets then
Send the received packets to the root node through default route

4.1.2 Algorithm for wormhole detection at the 6BR

(1) Calculate distance between each node
(2) If neighbor info received from node N_i then
If actual distance between N_i and its neighbor is more than the range of node N_i then
(a) Generate Alert for attack
(b) Send victim packet to nbr_info sender node and other victim neighbor node
(c) If other victim neighbor node is 6BR the
Initiate the Monitoring algorithm
(3) If victim packet is sent then
Wait for time until all RSSI values to be received and until then no other new attack processing
(4) If RSSI value from node N_i is received the drop all further RSSI_val packets from node N_i as duplicate packets
(5) If RSSI wait timer expires and at least one RSSI value received then
(a) Find the RSSI to distance d for all received RSSI values
(b) Find the all nodes which are in the range of distance d considering error in measurments of RSSI values. these nodes are suspect nodes.
(c) Keep count with all such suspect nodes, for how many time it is suspected as attacker node.
(d) The suspect node having high probability is consider as attacker node.

4.1.3 Algorithm for monitoring node

(1) When node receives victim packet contains its own ID then
(a) It start monitoring for other victim colleague node (Third field in victim packet)
(b) Start transmission of n victim packets to other victim colleague
(c) Similarly other victim colleague node does same on receiving such victim packet
(2) When node receives victim packet that does not contains its own ID then
(a) If both victim nodes are original neighbors of node then it does not monitor for any node
(b) If both victims are not original neighbors of node then it monitor for unknown node, i.e. It records the RSSI value of received victim packets
(c) If one victim is original neighbor of node and other isn't then it monitor for victim node which is not its original neighbor. means record the RSSI value of the victim packets that it is receiving form non original neighbor node.

4.2 Attacker Node Creation

Only Packet relay and encapsulation kind of wormhole attacker are evaluated in experimentation. Various configurations are there to construct the attacker node, ex. encapsulation kind of wormhole can be created at network and mac layer also. Let see how attacker node does malicious activity.

4.2.1 Packet Relay Wormhole Attacker. In this kind of attacker node, it relays/transmits the packets which are received on its radio interface without making any changes in packet. In Fig 7 we can see the various layers of Contiki OS. At radio layer for listening the packets from all node first the radio interface is put in promiscuous mode, so that it can get the packets which are not for him. In cc2420 configuration by making the register bits of CC2420_MDMCTRL0 (address decoder) to 0 we can achieve it. Now we can get the packets of other node. Next step is relaying the packet, here the all packets are relayed including unicast, broadcast, packets belongs to other PAN. This is done at RDC (Radio duty cycling layer) layer only know as sicslowmac in Contiki. We are not letting packets to go at upper layer.

Contiki OS Layer	Protocol/Interface
Transport	UDP
Network, Routing	IPv6, RPL
Adaptation	Sicslowpan
MAC	CSMA
Duty Cycling	Sicslowmac
Radio	cc2420

Fig. 7. Contiki Layer wise

4.2.2 Encapsulation Type Wormhole Attacker. We are constructed the encapsulation kind of wormhole attacker using 3 node, 2 attacker node and 1 intermediate node helping to establish tunnel between them. First step is same as to put radio interface in promiscuous mode. In second step to avoid the loop formation between attacker the change in source PAN ID is used. In Fig 9 MAC PDU is shown, so 2 bytes source PAN ID field is in addressing fields of PDU. Consider attacker in Fig 8 the 1 Byte in Source PAN ID is adjusted or changed such that the node can easily identify that from which node it has received this packet and which is now next node to whom to send. If 1 want to send packet to 3 then PAN ID changes from 170 to 171 to 172 to again 170 which is PAN ID of entire network. During this the normal node listening packets in which PAN ID is not 170 are discarded only attacker node process such packets.

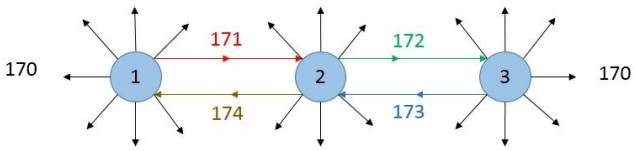

Fig. 8. Encapsulation Kind of Wormhole Attack

Frame Control	Data Seq Number	Destination PAN ID	Destination Address	Source PAN ID	Source Address	Frame Payload	FCS
2	1	2	8	2	8	0-102	2

Fig. 9. MAC Layer Packet Format

5. DISCUSSION ON WORMHOLE ATTACK DETECTION

Fig 10 shows the RPL tree. Initially during the network initialization period each node stores their original neighbor information. Node 1 is 6BR, 10 and 13 (Black colored) are attacker nodes. They formed the wormhole tunnel. When the attack begins, the packets received on radio interface of 10 will be send to 13 through wormhole tunnel, and similarly from 13, node 10 will receive the packets and here node 10 will relay the packets. When node sends the control packets (Neighbor advertisement, Neighbor solicitation DIO etc.), these packets will reach in other side of tunnel. When nodes in the range of attacker realize existence of new neighbors, they send neighbor information to 6BR for validation purpose. In

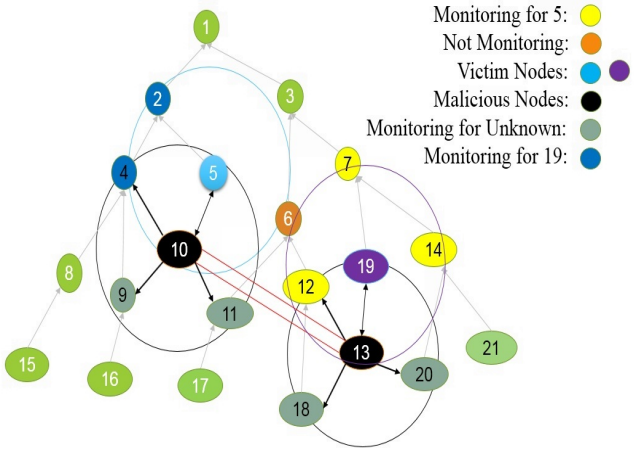

Fig. 10. Wormhole Attack in RPL

this scenario if node 5 send the information about 19 as new neighbors, 6BR confirms using location and range information they are out of range and declares the existence of attack. 6BR send victim packet to 5 and 19. Then according to algorithm n victim packets are broadcast. The nodes which listen broadcast packets follows the monitoring algorithm for recording of RSSI value form right node. Here 4, 2 records RSSI value of packet received from 19 as 5 is original neighbor similarly for 7, 14, and 12. Node 6 doesn't monitor for any node as both 5 and 19 are its neighbors and it may cause confusion for which node to monitor. Node 9, 11, 18, 20 just records the RSSI value of victim packets it received, as both 5 and 19 are not their neighbors. Node 2 and 7, 14 never receives the packets from 19 and 5 respectively, so after waiting for fixed time they will continue to normal operation. When RSSI value received from all node at 6BR, it find out node at distance d_i, where d_i is calculated form RSSI value R_i. At last the nodes having high count of suspect is a attacker node.

6. RESULTS AND DISCUSSIONS

In this section we present the evaluation of proposed system in terms of detection rate, energy, packet and memory overhead.

6.1 Experimental setup

We run our experiments in Contiki's network simulator Cooja that has shown to produce realistic results. Cooja runs deployable Contiki code. In our simulations, we use emulated Tmote Sky nodes. In general, we expect that the 6BR is not a constrained node and it can be a PC or a laptop; however, currently there exists no PC equivalent 802.15.4 devices, therefore we run the 6BR natively i.e. JNI (Java Native Interface) on Linux. The protocol configuration is as, as Radio interface cc2420 is used, at RDC (Radio Duty Cycling) layer sicslowmac is used, which is 802.15.4 compatible. Above this layer, in MAC CSMA (Carrier Sense Multiple Access) protocol is used. At network layer sicslowpan (6LowPAN), IPv6 and RPL as routing protocol is used. UDP is as transport layer protocol.

6.2 Topologies For Experimentation

We have considered there topologies 8, 16, 24 nodes as shown in Fig 11, 12, 13 respectively. The placement of node are random. The node number 1 is 6BR node shown in unique color in each topology. Rest node are the Tmote sky node running same IDS dis-

tributed module, and 6BR runs centralized modules. Topology is adjusted such that each node should be in the range of at least one other node, such that tree structure formed and network partition should not be there.

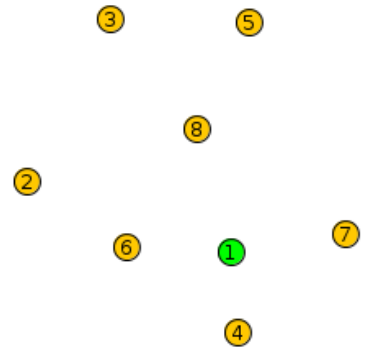

Fig. 11. 8 Nodes Topology

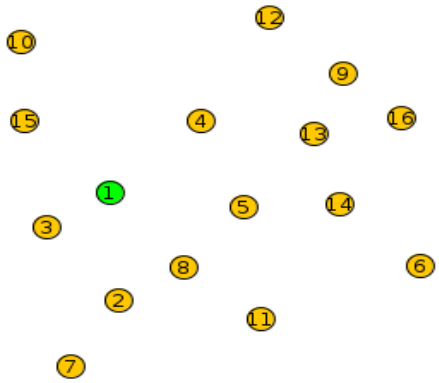

Fig. 12. 16 Nodes Topology

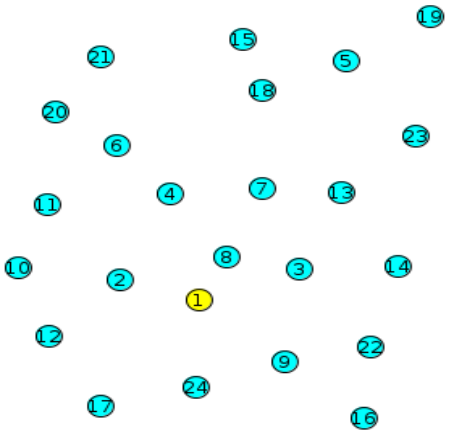

Fig. 13. 24 Nodes Topology

6.3 True Positive Detection Rate

For detection rate we have performed the various number of simulation on 8, 16, 24 node topology. Taking distinct nodes as attacker. This result is combination of both Packet relay and Encapsulation kind of wormhole attack. The result for attack detection in graph 14 is 94%, and of both attacker and attack is 87%. Only attack detected but not attacker is more found in Encapsulation kind of attack and where there is not sufficient number of nodes to monitor RSSI values and if victim packet unable to reach to victim nodes. The nodes at the leaf of tree are mostly not considered as attacker as, attack by them does not affect the normal operation of topology, ex. node 19 in 24 node topology is not considered as attacker and if attack performed by 19 (Packet relay) the no wrong neighbors can be formed and thus, unable to detect such attacker. All detection rate is only depend on successful packet delivery.

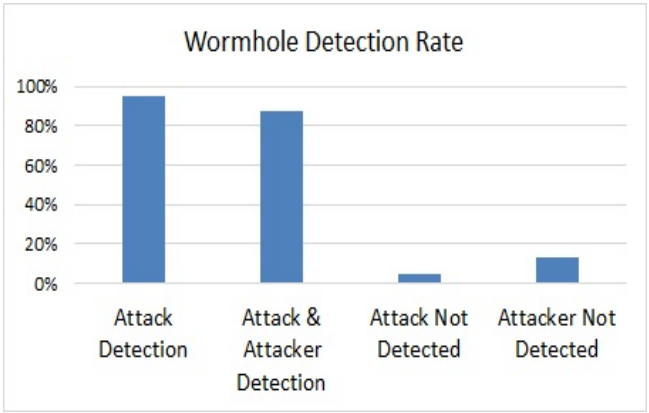

Fig. 14. True Positive Detection Rate

6.4 Energy overhead caused by IDS

The nodes in the IoT are usually battery powered and hence energy is a scarce resource. Here we measure IDS's power consumption and overhead. We use Contiki Powertrace [15] to measure the power consumption. The output from the Powertrace application is the total time the different parts of the system were on. We calculate the energy consumption using the nominal values, the typical operating conditions of the Tmote sky, shown in Table 2. We use 3 V in our calculations. Micro-Controller Unit (MCU) idle while the radio is off is referred to as low power mode, or LPM (Low Power Mode). The time the MCU is on and the radio is off is referred to as CPU time. The time the radio is receiving and transmitting with the MCU on is referred to as listen and transmit respectively.

We measure energy in both IDS and Hello World application running for 30 min for knowing overhead energy consumed by IDS system. We run each experiment in a network of 8, 16 and 24 emulated Tmote sky nodes, with nodes placed at the same locations. Fig. 15 shows the network-wide energy usage for 30 min by all the nodes in Hello World application and IDS, calculated as follows

$$Energy(mJ) = ((CPU * 1.8 + LPM * 0.0545 + Transmit * 19.5 + Listen * 21.8) * 3)/(4096 * 8) \quad (1)$$

Table 2. Tmote Sky Operating Conditions[16]

Typical operating conditions	Min	NOM	Max	Unit
Voltage	2.1		3.6	V
MCU on, Radio RX		21.8	23	mA
MCU on, Radio TX		19.5	21	mA
MCU on, Radio off		1800	2400	μA
MCU idle, Radio off		54.5	1200	μA
MCU standby		5.1	21.0	μA

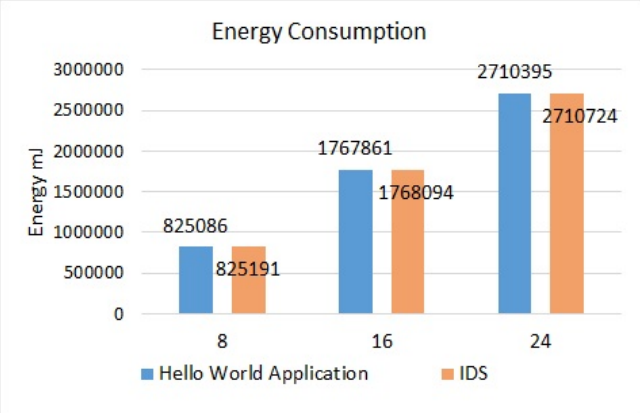

Fig. 15. Energy Consumption in 30 min

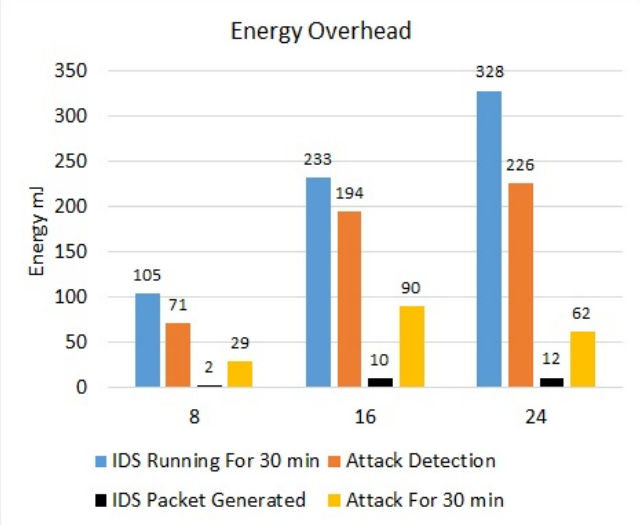

Fig. 16. Energy overhead for various events

The overhead caused by IDS in 30 min is shown in Fig 16, First bar in 8, 16, 24 nodes. It is very low for constrained node also. We also measured the attack detection overhead, i.e. in second bar. This is calculated by taking difference of Energies of topology in which there was no attack (clean network with Hello World running on sensor node) and the other topology in which IDS was installed and we detected the attack. We also measured the energy required for packet during attack detection, i.e. in third bar. This is calculated by taking difference of Energies of topology in which there was attack but we didn't detected it and the other topology in which we detected the attack. Here In both topologies IDS were installed on nodes. The last bar (Yellow color) shows the energy consumption overhead when attack takes place in 30 min. In all cases we given initial 5 min for network settlement and triggering the attack at 5th min.

6.5 Packet Overhead

We have also measured the packet overhead during the attack detection and due to the attack in network. In Fig 17 we can see the various packet overhead for the same topologies discussed in experimental setup section. We ran simulation for 30 min and taken the difference of packets in clean network and attack occurred at 5th min network. Here we can see the there is no change in DIS packets in both cases but Neighbor advertisement and neighbor solicitation found to be more in after attack takes place. The overall increase in control packet after attack occurred is considerable.

In Fig 18, we have shown the packet overhead in attack detection. Our IDS uses only UDP packets to detect the attack, so in graph we can see that the increase in UDP packets. We triggered the attack at 5th min and is detected at average 11th min. The packet requirement is depend on the location of wormhole attacker and denseness around the attacker node.

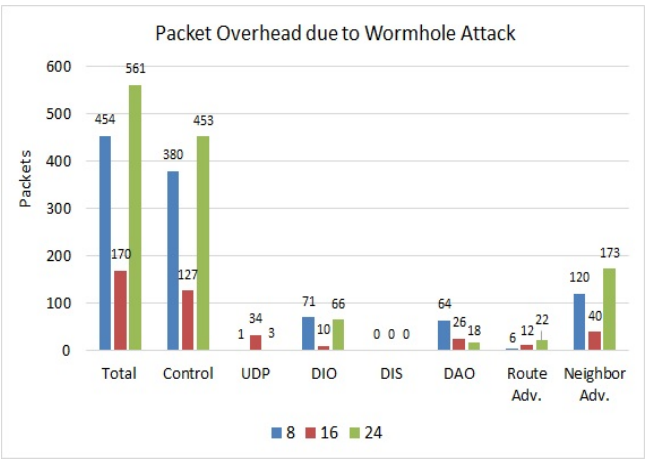

Fig. 17. Packet overhead due to wormhole attack

6.6 Memory consumption

In Table 3 we show the ROM and RAM requirements of IDS's different modules. The baseline for each configuration is different as some depend on different parts of the Contiki system. For example, the 6BR that resides in the PC requires more ROM than other nodes. However, the total additional ROM required to host IDS's modules inside a constrained node is 24.9 KB which is well below the total available ROM in constrained devices such as 48 KB in Tmote sky. It is important to note the overhead column which

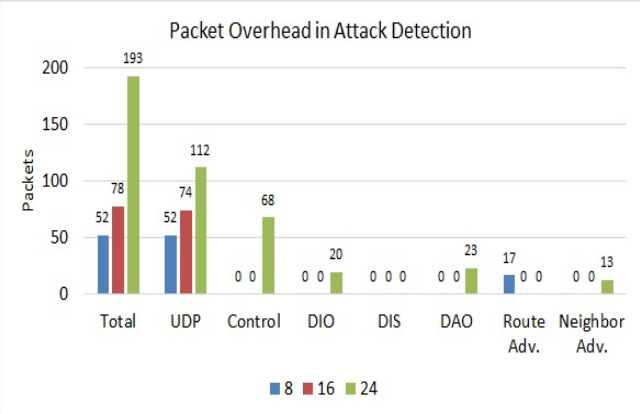

Fig. 18. Packet overhead in attack detection

shows the pure overhead of IDS modules in Contiki. This overhead is the difference between Hello world application and IDS, as Hello world application inside Contiki is the lightest application available. Even though centralized modules are not targeted towards running on constrained nodes it is still lightweight enough and can be used for small networks.The total RAM size in the Tmote sky is 10 KB, hence IDS modules with 2.8 KB additional RAM requirement can easily run in constrained nodes.

Table 3. Additional ROM and RAM usage by IDS

Node /Size (B)	ROM total	RAM total	ROM occupied	RAM occupied	ROM Overhead	RAM Overhead
6BR	1 MB	-	1,58,813	64,340	96,962	42,636
Sky_Mote	48KB	10KB	43,098	7,454B	24,900	2,886

7. FUTURE WORK

The proposed IDS system are very easy to extend. There are a number of potential attacks against the Internet of Things and it is likely that more attacks will be discovered. The location information of nodes will also help to mitigate the Sybil and Clone ID attacks and will enhance its intrusion detection capabilities. RPL specific attacks Version Number and Local Repair attack can be detected by validating DODAG version and ID at 6BR [9]. Wormhole attack can be combined with the selective forwarding attack e.g. sending either data or control packet through tunnel detecting this could be an extension for proposed system. This system also able to detect the neighbor attack [17] only but not evaluated yet. Minor changes in system will help to detect the Neighbor attacker also.

8. CONCLUSION

Considering the potential applications of the IoT it is important that 6LoWPAN networks are protected against internal and external intrusions. This work concludes that, the proposed novel light weight IDS system is basically designed for resource constrained sensor nodes and able to detect Wormhole attacks of two kind packet relay and encapsulation. Mostly centralized modules are used for doing heavy processing and Light weight modules run on sensor nodes causing saving of energy on sensor nodes. Adding location information of nodes made system more efficient for detection of wormhole attack with lesser overhead and with high true positive detection rate. This method takes fixed number of UDP packets for attack detection. The RAM/ROM consumption is also very small as compared to total available sizes. The method given 94% detection rate which is very good for resource constrained environment.

Acknowledgment

I am thankful of my guide G. T. Chavan for his guidance and constant encouragement throughout the course of this work. Lastly, I thank almighty, my family and friends for their constant encouragement without which this work would not be possible.

9. REFERENCES

[1] IETF, RPL. “Routing Over Low Power and Lossy Networks.”

[2] Wallgren, Linus, Shahid Raza, and Thiemo Voigt. “Routing Attacks and Countermeasures in the RPL-based Internet of Things.” *International Journal of Distributed Sensor Networks 2013*, 2013.

[3] Song N, Qian L, Li X, “Wormhole attacks detection in wireless ad hoc networks: a statistical analysis approach”, *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International,vol., no., pp. 8 pp.*, 4-8 April 2005.

[4] H. S. Chiu and K. Lui, “DelPHI: Wormhole Detection Mechanism for Ad Hoc Wireless Networks”,*In Proceedings of International Symposium on Wireless Pervasive Computing, pp. 6-11*, 2006.

[5] L. Hu and D. Evans, “Using directional antennas to prevent wormhole attacks”, *Proc. Symp. Netw. Distrib. Syst. Security*, 2004

[6] Dhurandher, Sanjay Kumar, et al., “E2SIW: An Energy Efficient Scheme Immune to Wormhole Attacks in Wireless Ad Hoc Networks. ”, *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on. IEEE*, 2012

[7] Raju, V. Karthik, and K. Vinay Kumar, “A Simple and Efficient Mechanism to Detect and Avoid Wormhole Attacks In Mobile Ad Hoc Networks”, *Computing Sciences (ICCS), 2012 International Conference on. IEEE*, 2012.

[8] Yifeng Zhou , Lamont L , Li Li,“Wormhole attack detection based on distance verification and the Use of hypothesis testing for wireless ad hoc networks”, *Military Communications Conference, MILCOM IEEE*, 2009

[9] Pavan Pongle, Gurunath Chavan, “A survey: Attacks on RPL and 6LoWPAN in IoT ”, *International Conference on Pervasive Computing (ICPC) IEEE*, 2015

[10] Amin, Syed Obaid, et al. “A novel coding scheme to implement signature based IDS in IP based Sensor Networks.” *Integrated Network Management-Workshops, 2009. IM’09. IFIP/IEEE International Symposium on. IEEE*, 2009.

[11] Kasinathan, Prabhakaran, et al. “Denial-of-Service detection in 6LoWPAN based internet of things.” Wireless and Mobile Computing, Networking and Communications (WiMob), *2013 IEEE 9th International Conference on. IEEE*, 2013.

[12] Le, Anhtuan, et al. “Specification-based IDS for securing RPL from topology attacks.” Wireless Days (WD), 2011 IFIP. IEEE, 2011.

[13] Raza, Shahid, Linus Wallgren, and Thiemo Voigt. "SVELTE: Real-time intrusion detection in the Internet of Things." Ad hoc networks 11.8 (2013): 2661-2674.

[14] Jun, Chen, and Chen Chi. "Design of Complex Event-Processing IDS in Internet of Things." *Measuring Technology and Mechatronics Automation (ICMTMA), 2014 Sixth International Conference on. IEEE*, 2014.

[15] A. Dunkels, J. Eriksson, N. Finne, N. Tsiftes, Powertrace: NetworkLevel Power Profiling for Low-Power Wireless Networks, 2011.

[16] http://www.eecs.harvard.edu/k̃onrad/projects/shimmer /references/tmote-sky-datasheet.pdf

[17] Le, Anhtuan, et al. "The impacts of internal threats towards Routing Protocol for Low power and lossy network performance." *Computers and Communications (ISCC), 2013 IEEE Symposium on. IEEE*, 2013.