International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue
1st International Conference on Advent Trends in Engineering, Science and Technology
"ICATEST 2015", 08 March 2015

Mitigation of Vampire Attack in Wireless Ad-hoc Sensor Network

Mr. Shrikant C.Chumble[1], Prof. M. M. Ghonge.[2]
[1] *Student, JCET,Yavatmal,shrikantchumble1@gmail.com*
[2] *Assistant Professor, JCET, Yavatmal, mangesh.cse@gmail.com*

Abstract: Ad-hoc low-power wireless networks are an exciting and most promising research direction in sensing and pervasive computing. An ad hoc network is a group of wireless nodes, in which each node can communicate over multi hop paths to any other node without the help of any preexisting infrastructure such as base station or access points. The security work in this field has focused only on denial of service at the routing or medium access control level. An important security issue that has been identified in these networks is resource depletion attack at routing layer protocol, which permanently disables networks by quickly draining nodes' battery power. In networks, routing protocol gets affected from attack even though designed to be secure. This type of attack called "Vampire attacks" which is not specific to any protocol, but dependent on the properties of many popular classes of routing protocols. This study shows that all examined protocols are vulnerable to Vampire attacks, which are destructive and difficult to detect and are easy to carry out using as few as one malicious insider sending only protocol compliant messages. In the worst case, a single Vampire can increase network-wide energy usage by a factor of O(N), where N is the number of network nodes. This seminar discuss methods to detect and mitigate these types of attacks, including a new protocol concept with attestations that is useful to avoid damage caused by Vampires and provide secure packet forwarding phase. It will also save Ad-hoc wireless nodes from power drainage due to vampire packets.

Keywords – Ad-hoc networks, Denial of service, resource depletion attack, Vampire attacks, security, routing, protocol, sensor networks, wireless networks.

1. INTRODUCTION

1.1. Wireless Sensor Networks and Ad-hoc WSN:

A wireless sensor network (WSN) is a wireless network consisting of specific type of distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. Sensor network is basically a collection of a large number of sensor nodes that are deployed in a wide area with very low powered sensor nodes. The wireless sensor networks can be utilized in a various information and telecommunications applications. The sensor nodes are very small devices with wireless communication capability, which can collect information about sound, light, motion, temperature etc and processed different sensed information and transfers it to the other nodes. The following figure-1.1 illustrated the Wireless Sensor Network scenario.

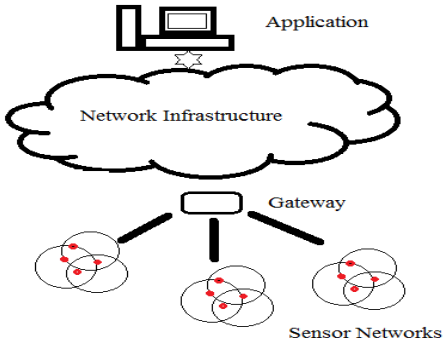


Fig.1. Wireless sensor network

Ad-hoc wireless sensor networks (WSNs) is exciting and most promising new applications in the near future, such as ubiquitous on-demand computing power, continuous connectivity, and instantly-deployable communication for military and first responders. Such networks already monitor environmental conditions, factory performance, and troop deployment, to name a few applications. As WSNs become more and more crucial to the everyday functioning of people and organizations, availability faults become less tolerable lack of availability can make the difference between business as usual and lost productivity, power outages, environmental disasters, and even lost lives; thus high availability of these networks is a critical property, and should hold

even under malicious conditions. Due to their ad-hoc organization, wireless ad-hoc networks are particularly vulnerable to denial of service (DoS) attacks [13] and a great deal of research has been done to enhance survivability [2,4,5,13]. It can prevent attacks on the short-term availability of a network; they do not address attacks that affect long-term availability the most permanent denial of service attack is to entirely deplete nodes' batteries. This is an instance of a resource depletion attack, with battery power as the resource of interest. we consider how routing protocols, even those designed to be secure, lack protection from these attacks, which we call Vampire attacks since they drain the life from networks nodes. These attacks are distinct from previously-studied DoS, reduction of quality (RoQ), and routing infrastructure attacks as they do not disrupt immediate availability, but rather work over time to entirely disable a network. While some of the individual attacks are simple, and power-draining and resource exhaustion attacks have been discussed before [8,9] prior work has been mostly confined to other levels of the protocol stack, e.g. medium access control (MAC) or application layers, and to our knowledge there is little discussion, and no thorough analysis or mitigation, of routing-layer resource exhaustion attacks. Vampire attacks are not protocol specific, in that they do not rely on design properties or implementation faults of particular routing protocols, but rather exploit general properties of protocol classes such as link-state, distance-vector, source routing .

Now the problem is to detect Vampire attacks and defining them i.e. how to detect this attack? Consider the process of routing a packet in any multi-hop network: a source composes and transmits it to the next hop toward the destination, which transmits it further, until the destination is reached, consuming resources at every node including source node through which messages move. The act of sending a message is in itself an act of amplification, leading to resource exhaustion, as long as the aggregate cost of routing a message (at the intermediate nodes) is lower than the cost to the source to compose and transmit it. So, we focus on the cumulative energy consumption increase that a malicious node can cause while sending the same number of messages as an honest node. Vampire attack is the composition and transmission of a message that causes more energy to be consumed by the network than if an honest node transmitted a message of identical size to the same destination, although using different packet headers. We measure the strength of the attack by the ratio of network energy used in the benign case to the energy used in the malicious case, i.e. the ratio of network-wide power utilization with malicious nodes present to energy usage with only honest nodes when the number and size of packets sent remains constant. To avoid Vampire attacks we try to maintain this ratio to 1. Energy use by malicious nodes is not considered, since they can always unilaterally drain their own batteries.

In this seminar we consider and study the effect of Vampire attacks on link state, distance-vector and source routing protocols. Different types of routing protocols which are also vulnerable to Vampire attacks. As we know that ad hoc deployment implies no prior position knowledge so all routing protocols employ at least one topology discovery period. We consider immutable but dynamically organized topologies and then differentiate between on-demand routing protocols, where topology discovery is done at transmission time, and static protocols, where topology is discovered during an initial setup phase, with periodic rediscovery to handle rare topology changes. Adversaries are malicious insiders and have the same resources, same level of network access as honest nodes. Adversary location within the network is assumed to be fixed and random. Using this configuration we try to represents the average expected damage from Vampire attacks and also assumes that a node is permanently disabled once its battery power is exhausted. In this, nodes that recharge their batteries using either continuous charging or switching between active and recharge cycles. In the continuous charging case, power-draining attacks would be effective only if the adversary is able to consume power faster than nodes can recharge. Continuously recharging adversary can keep at least one node permanently disabled at the cost of its own functionality. We know that sending any packet automatically constitutes amplification, allowing few Vampires to attack many honest nodes. Dual-cycle networks (with mandatory sleep and awake periods) are equally vulnerable to Vampires during active duty as long as the Vampire's cycle switching is in sync with other nodes. As active-duty nodes are vulnerable while the Vampire is active; nodes are safe while the Vampire sleeps.

In this study, we will also try to show that in source routing protocols, how a malicious packet source can specify paths through the network which are far longer than optimal, wasting energy at intermediate nodes that forward the packet based on the included source route. On the other side, routing schemes where forwarding decisions are made independently by each node, we suggest how directional antenna and wormhole attacks can be used to deliver packets to multiple remote network positions, forcing packet processing at nodes that would not normally receive that packet at all, and thus increasing network-wide energy expenditure. At last, we show how an adversary can affect not only packet forwarding but also route and topology discovery phases if discovery messages are flooded, an adversary can consume energy at every node in the network.

In our first attack, an adversary composes packets with purposely introduced routing loops i.e. allowing a single packet to repeatedly traverse the same set of nodes this type of attack is called carousel attack. In second attack, by considering source routing, an adversary constructs artificially long routes, potentially traversing every node in the network i.e. it increases packet path lengths, causing packets to be processed by a number of nodes that is independent of hop count along the shortest path between the adversary and packet destination .This attack is called as stretch attack. We study the different mitigation methods to bind the damage from Vampire attacks. The first mitigation method is loose source routing, where any forwarding node can reroute the packet if it knows a shorter path to the destination. In our second method, we modify the protocol from [8] to guarantee that a packet makes progress through the network, it holds if and only if a packet is moving strictly closer to its destination with every hop. We call this the no backtracking property, and it helps to mitigates Vampire attacks. There are two phases, topology discovery period we call it "the night" when vampires are more dangerous, followed by a long packet forwarding period during which adversarial success is provably bounded. We also modify the protocol to detect Vampires during topology discovery and evict them from the network make "dawn".

2. LITERATURE REVIEW

Prior security to mitigate different types of attacks in this field has focused primarily on denial of communication at the routing or medium access control levels. Power draining itself is not novel, but rather that these attacks have not been rigorously defined, evaluated, or mitigated at the routing layer. Early, power exhaustion can be found as "sleep deprivation torture". As name suggest that proposed attack prevents nodes from entering a low-power sleep cycle, and thus deplete their batteries faster. These "denial of sleep" only considers attacks at the medium access control (MAC) layer [9].In addition to this resource exhaustion at the MAC and transport layers [13], but only offers rate limiting and elimination of insider adversaries as potential solutions. . Malicious cycles (routing loops) have been briefly mentioned [8] but no effective defenses are discussed other than increasing efficiency of MAC and routing protocols or switching away from source routing. Flood attack, wherein adversaries make multiple connection requests to a server, which will allocate resources for each connection request, eventually running out of resources, while the adversary, who allocates minimal resources, remains operational hence depletion of resources such as memory, CPU time, and bandwidth may easily cause problems excluding power-constrained systems [14]. This attack is defeated by putting greater burden on the connecting entity like cookies. It place minimal load on legitimate clients who only initiate a small number of connections, but deter malicious entities who will attempt a large number. It is not always desirable as it punishes nodes that produce burst of traffic but may not send much total data over the lifetime of the network.

Several literatures on attacks and defenses against quality of service (QoS) degradation, or reduction of quality (RoQ) attacks, that produce long-term degradation in network performance [3]. It focuses on the transport layer rather than routing protocols. As Vampires do not drop packets, the quality of the malicious path itself may remain high and it should not be even considered. Denial of service in ad-hoc wireless networks as primarily dealt with adversaries who prevent route setup, disrupt communication, or preferentially establish routes through themselves to drop, manipulate, or monitor packets [5,10]. Protocols used only ensuring that valid network paths are found, cannot protect against Vampire attacks, since Vampires do not use or return illegal routes or prevent communication in the short term. In minimal-energy routing, it uses less energy to transmit and receive packets (e.g. by minimizing wireless transmission distance)[6,11] & increase the lifetime of power constrained networks. Vampires will increase energy usage even in minimal-energy routing scenarios and when power-conserving MAC protocols are used though these attacks cannot be prevented at the MAC layer or through cross-layer feedback [12,11]. Attackers will produce packets which traverse more hops than required, so even if nodes spend the minimum required energy to transmit packets, each packet is still more expensive to transmit in the presence of Vampires.

Path-based DoS attacks as in [4] use one-way hash chains to limit the rate at which nodes can transmit packets. In this malefactor overwhelms honest nodes with large amounts of data, but it is applicable only to traditional DoS, doesn't work with intelligent adversaries who use a small number of packets i.e. protocol compliant in which intelligent packet-dropping strategies can degrade performance of TCP streams traversing those nodes [2]. Adversaries are also protocol-compliant in the sense that they use well-formed routing protocol messages. They either produce messages when honest nodes would not used, or send packets with protocol headers different from what an honest node would produce in the same situation. Now the path-based wormhole and directional antennas attack, it allows two non neighboring malicious nodes with either a physical or virtual private connection, these links are not made visible to other network members, but can be used by the colluding nodes to privately exchange messages. It disrupt route discovery, return routes that traverse the

wormhole and may have artificially low associated cost metrics such as number of hops or discovery time. Author gives defense against wormhole and directional antenna attacks called "Packet Leashes" [7]; their solution is of high cost and is not always applicable. Packet Leashes relies on tightly synchronized clocks and packet travel time dominates processing time. In this all types of attacks and disadvantage of defenses mentioned in different literatures by different authors, performance of power drainage of nodes is unavoidable so it is difficult to mitigate the Vampire attack.

3. PROBLEM STATEMENT

Vampire attacks are attack in networks; it is the composition and transmission of a message that causes more energy to be consumed by the network, than if an honest node transmitted a message of identical size to the same destination i.e. Vampire attack means creating and sending messages by malicious node which causes more energy consumption by the network leading to slow depletion of node's battery life. Vampire attack happens in the network in the sense, any of the nodes in the network which is affected or infected and this nodes behavior is abruptly changing for the network behavior, this kind of nodes are called "Malicious node". If malicious nodes present in the network energy that have been using by each and every nodes will increases drastically. The malicious node has been place in the network uniquely. First In between the routing nodes, and the second placed in the Source node itself. The routing path is discovered by source node by using shortest path routing algorithm and the path should not be changeable by the intermediate nodes. In this type of occasion there is a chance to happening attack. The adversary composes packets with purposely introduced routing loops. This is one of the major problems of the network where the consuming energy of each and every node in the network will increase. The main problem these kinds of attackers are it's not easily identified if it attacked or affected the network. It will take some long time to identify and make ensure that it presented in the network. We further study this vampire attacks and identified problems by classifying it on the basis of type of protocol used during packet routing within network.

3.1. Attacks on Stateless Protocols:

In this we present simple but previously neglected attacks on source routing protocols, such as DSR. In these systems, the source node specifies the entire route to a destination within the packet header, so intermediaries do not make independent forwarding decisions, relying rather on a route specified by the source. To forward a message, the intermediate node finds itself in the route (specified in the packet header) and transmits the message to the next hop. The burden is on the source to ensure that the route is valid at the time of sending, and that every node in the route is a physical neighbor of the previous rout hop. This approach has the advantage of requiring very little forwarding logic at intermediate nodes, and allows for entire routes to be sender-authenticated using digital signatures [10]. It will be possible to evaluate both the carousel and stretch attacks in a randomly-generated 30-node topology and a single randomly-selected malicious DSR agent, using the ns-2 network simulator [1].

3.1.1. Carousel attack:-

In this attack, an adversary sends a packet with a route composed as a series of loops, such that the same node appears in the route many times. This strategy can be used to increase the route length beyond the number of nodes in the network, only limited by the number of allowed entries in the source route. An example of this type of route is in Fig2. The path (1-2-3-4-5-6-7) shows the honest path and path (1-2-3-4-5-6-5-4-3-2-3…7) shows loops the malicious path from source node 1 to destination (sink) node 7. Fig. 2.shows the carousel attack same node appears in the route many times.

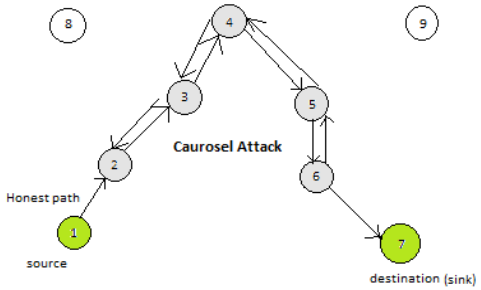


Fig. 2 The carousel attack same node appears in the route many times form loop (1-23456-65432-…….-23456-7)

In this attack, at every node in network through which packet route, it exploits limited verification of message headers at forwarding nodes. So that it is used to increase the route length beyond number of nodes in network and increase energy consumption. Theoretical limit: energy usage increase by a factor of $O(\lambda)$, where λ is the maximum route length.

3.1.2. Stretch attack:-

Another attack in the same vein is the stretch attack, where a malicious node constructs artificially long

source routes, causing packets to traverse a larger than optimal number of nodes. It causes a node that doesn't lie on optimal path to process packets. In the example, given below honest path shown with only green colored node path and adversary or malicious path with green and gray colored node path. The honest path is very less distant but the malicious path is very long to make more energy consumption as shown in fig 3. An honest source would select the route source →1-2-3-4-5-6-7 → destination (sink) affecting seven nodes including it, but the malicious node selects a longer route, affecting all nodes in the network. These routes cause nodes that do not lie along the honest route to consume energy by forwarding packets they would not receive in honest scenarios.eg. As shown in fig instead of honest path source 1-2-3-4-5-6-7 destination (sink), it uses long path by replacing 4-10-8-11-12-13-9-5 long route instead of 4-5 in between packet transmission through network. So more energy is consumed to transmit packets, Theoretical limit over this attack allowed energy usage increase of factor O(min(N, λ)), where N is the number of nodes in the network and λ is the maximum path length allowed. It is potentially less damaging per packet than the carousel attack, as the no of hops per packet is bounded by the number of network nodes.

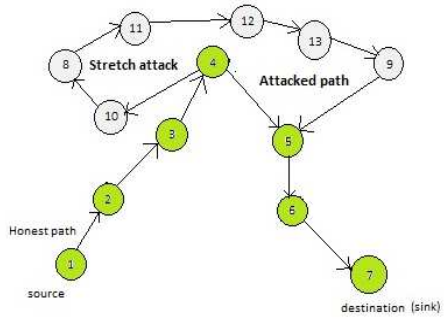


Fig3.The Stretch attack with two different paths from source to destination (4-10-8-11-12-13-9-5—long route instead of 4-5)

3.2. Attacks on Stateful Protocols:

In stateful routing protocols, every node within network is aware of the whole network topology and its state, and makes local forwarding decisions which are based on that stored state. There are two different types of important classes of stateful protocols are link-state and distance-vector. In link-state protocols, such as OLSR, every node in network for particular topology keep a record of the up-or-down state of links in the network, and flood routing try to updates every time a link goes down or make a new link enabled. DSDV distance-vector protocols keep track of the next hop up to every destination, indexed by different route cost metrics, e.g. the number of hops, time of transmission etc. In this protocol system, only routing updates that affect the cost of a given route need to be propagated in forward direction. In both link-state and distance-vector networks routes are built dynamically from many independent forwarding decisions made by every node, so adversaries have very limited power to affect packet forwarding, and it make possible that protocols may caused carousel and stretch attacks.

Basically in these types of protocol system, no full path can be specified by a malicious source which is the case of DSR, but still in this malicious node can still forward packets in wrong direction. An adversary can forward packets either backward direction toward the source if the adversary is an intermediary or to a non optimal next hop within network if the adversary is either an intermediary or the source. Consider a ring topology in which forwarding a packet in the backward direction causes it to traverse every node in the network and it increases energy consumption. On the basis of direction provided by nodes or malicious nodes in route discovery phase is divided into two types i.e. one is Directional antenna attack and other is Malicious Discovery Attack which are as follows:-

3.2.1 Directional Antenna Attack:-

Forwarding decisions that are made independently by each node is much important and Vampires have little control over packet progress at that instance, but it restart to forward packets and waste energy in various parts of the network. Adversaries can use directional antenna to deposit a packet in any arbitrary parts within network, and forwarding the packet locally. It makes consumption of the energy of nodes that would not have had to process the original packet, require additional honest energy expenditure of O(d), where d is the network diameter, making d/2 the expected length of the path as we consider only one direction to an arbitrary destination from the furthest point in the network. Since a directional antenna constitutes a private communication channel, and it is not necessarily required that the node on the other end in link is malicious so it may be considered a half-wormhole attack [7]. It can be repeating itself and performed more than once, depositing the packets at various distant points in the network, make additional cost to the adversary for each use of the directional antenna. Packet Leashes unable to prevent this attack since they are not protect against malicious message sources, only intermediaries nodes [7],so that it increase severity of the attack.

3.2.2 Malicious Discovery Attack:-

Now second type of attack on routing protocols (including stateful and stateless) is spurious route discovery phase in routing packets. In the case of most protocols, every node will forward route discovery packets. It means, may be possible that it

initiate a flood by sending a single message. AODV and DSR are particularly vulnerable in which it perform route discovery when needed since nodes may start route discovery at any time, not only when a topology changes. A malicious node has a number of ways to know about exact topology change, it may show simply that a link is down, or claim a new link an unknown node which is not in existence. Sometimes link between two malicious cooperating nodes may claim to be down. However, nearby nodes monitor communication and might be able to detect link failure. It is more serious when nodes claim a long distance route has changed, short route failure may ignored. Hence, it is trivial in open networks with unauthenticated routes. Whereas in closed networks link states are authenticated repeatedly announce and withdraw routes. So that two cooperating adversaries nodes communicating through a wormhole could repeatedly announce and withdraw routes by using wormhole, and increase a theoretical energy usage by factor of O(N) per packet. As we increase the malicious nodes, it increase repeatedly announce and withdraw routes. Here also packets leashes cannot prevent this attack because it may possible that originators i.e. source are malicious. It is also difficult to avoid as there may be no stable routes in WSNs.

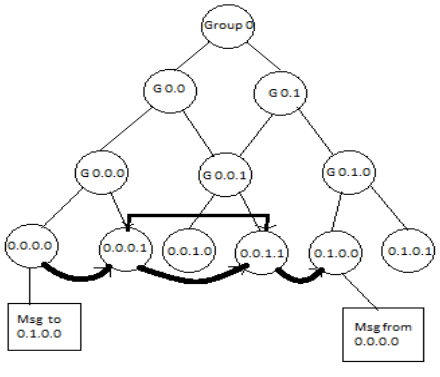
It is very essential to reduce the effect of Vampire attacks that we have study previously. To achieve our objective existing work is done on protocol system and provides secure routing protocol. Existing work in the field of secure routing protocols attempts to ensure that malicious nodes cannot cause path discovery to return an invalid network path, but during path discovery phase Vampires do not disrupt or change discovered paths, instead using and protocol compliant messages and valid network paths which is in existence. These limited power adversaries mainly have to affect forwarding of packets in network, making these protocols resistant to these Vampire attacks. By the use of directional antenna they can consume more energy by restarting packet in various parts of the network. Other such attack is spurious route discovery where each node will forward route discovery packets which means by sending a message it is possible to cause flood attack in network. Protocols rely on cooperative node behavior that maximize power efficiency are also not appropriate, and cannot optimize out malicious activity. So here we need to study clean-slat sensor network routing.

3.3. Clean-slat sensor network routing:

Clean-slat sensor network routing protocol can at some level able to resist Vampire attacks during packet forwarding phase which is given by Parno, Luk, Gaustad, and Perrig, so it is named as PLGP [8] but original version of PLGP is vulnerable to Vampire attacks. PLGP consists of two phases a topology discovery phase, followed by a packet forwarding phase repeated on a fixed schedule if needed to ensure that topology information stays current i.e. no on demand discovery. Discovery organizes nodes in the form of a tree that will later be used for an addressing. Initially the node knows only itself; nodes build a tree of neighbor relationships and group membership used for routing and addressing later. During the end of discovery, each node in network should compute the same address tree as other nodes. All physical nodes in the network are leaf nodes in the tree, and their position in the tree is corresponding to their virtual addresses. Virtual addresses and cryptographic keys of each node are known to all other nodes. All forwarding decisions can be independently verified and each legitimate network node has a unique certificate of membership.

3.3.1 Topology discovery:-

Initially, every node announces its presence by broadcasting a certificate of identity, including its public key. Each node begins as its own group of size one and with a virtual address 0. When two individual nodes (each with an initial address 0) form a group of size two, one of them takes the address 0, and other 1. Groups merge with smallest neighboring group i.e. it may be a single node and initially it takes choose a group address 0 & each group chooses 0 or 1 when merge with another group. Each child group member prefixes the parental group address to their own address.eg. Child member node 0 in parental group 0 becomes 0.0; other node 0 in parental group 1 becomes 1.0, and so on, then address of each member node sequentially increases by one bit. Information coded by the tree is specifying neighbor relationships among nodes. This tree is not a virtual coordinate system. As larger groups merge, then both groups broadcast their group IDs also including IDs of all group members in a group to each other, and proceed with a merge protocol identical to the two-node case. If there is enough growth in groups that some members are out of radio range of other groups will communicate with the help of "gateway nodes," which are within range of both groups. Each node in tree stores the identity of one or more nodes by which it make confirm that another group exists. At the end



of topology discovery phase each node knows every nodes virtual address, public key and certificate. All nodes must know about identity of all other node & form single group i.e. fully-converged node network e.g. As shown in fig.4 where Leaves represent physical nodes, connected with solid lines if within radio range. The dark thick line is showing progress of a message through the network. Here non-leaf nodes are not physical nodes but rather logical group identifiers.

Fig.4 The final address tree for fully-converged 6-node network

3.3.2. Packet forwarding:-

In this phase all decisions are made independently by each node. A node determines next hop by finding the most significant bit of its address that differs from the message originators address when it receives a packets. As shown in fig.4. Thus every forwarding event reduces the logical distance to the destination, since node addresses should be strictly closer to the destination. E.g. Function forward_packet(p) as shown in fig 5.

```
Function forward_packet(p)
s ← extract_source_address(p);
c ← closest_next_node(s);
if is_neighbor(c) then forward(p, c);
Else
r ← next_hop_to_non_neighbor(c);
      forward(p, r);
```

Fig.5. Function forward_packet(p)

Hence the exact problems in PLGP is that in the network forwarding nodes don't know about exact the path of a packet and in which adversaries divert packet to any part of the network ,though honest node may be far distance to destination node than malicious nodes. Honest node knows only its address and destination address no other information. Thus the Vampire moves packet away from the destination without detection. Then the theoretical energy increase of O (d) where d is the network diameter and N the number of network nodes. Worse case if packet returns to vampire as it can reroute similar to carousel attack but only difference is that packet can cycle indefinitely here. This makes it vulnerable to Vampire attack and reduces the security level of system.

4. PROPOSED WORK

To increase security against Vampire attack, we study and modify the forwarding phase of PLGP. We introduce the no-backtracking property, according to which packet consistently makes progress toward its destination in the logical network address space. No-backtracking is satisfied if every packet; consider packet p traverses the same number of hops whether or not an adversary is present in the network i.e. there is same number of hops between source (at location L) and destination(at location D) whether source is honest or malicious node. This means that the number of intermediate honest nodes traversed by the packet between source and destination is not affected by the actions of malicious nodes. It has same network-wide energy utilization. Only exceptions are when adversaries drop or mangle packets on the way called "pre-mangled" situation. No-backtracking used as Vampire resistance because in this each node independently check packet progress i.e. nodes keep track of route "cost" or "metric" and, when forwarding a packet, communicate about local cost to the next hop, that next hop can verify that the remaining route cost is lower than before, and therefore the packet is making progress toward its destination, if not drop packet as there is malicious intervention. But this property does not satisfied by PLGP, in which packets paths are further bounded by a tree, forwarding packets along the shortest route through the tree that is allowed by the physical topology which make it different from other protocols. A packet path depends on both physical neighbor relationships and the routing tree. Each member node of tree knows topology and holds an identical copy of the address tree lastly it can verify the optimal next logical hop. Since path previously traversed by a packet is not fix and it is also possible that adversaries provide false metric.

Now we can make modification in PLGP by embedding no backtracking property into it. We add a verifiable path history to every PLGP packet act as route authentications which we call protocol with attestation PLGPa. It uses this packet history and tree routing structure. Whenever node n forwards packet p attach attestation (signature) to itself. These signatures form a chain attached to every packet, allowing any node receiving it to validate its path. Every forwarding node verifies the attestation chain to ensure that the packet has never travelled away from its destination in the logical address space. As shown in fig.6. Function secure_forward_packet for the modified protocol.

```
Function secure_forward_packet(p)
s ← extract_source_address(p);
a ← extract_attestation(p);
if (not verify_source_sig(p)) or
(empty(a) and not is_neighbor(s)) or
(not saowf_verify(a)) then
                return ;
/* drop(p) */
for each node in a do
```

```
                prevnode ← node;
                if (not are_neighbors(node,
prevnode)) or
                (not making_progress(prevnode,
node)) then
                    return ;
/* drop(p) */
c ← closest_next_node(s);
p′ ← saowf_append(p);
if is_neighbor(c) then forward(p′, c);
else forward(p′, next_hop_to_non_neighbor(c));
```

Fig.6. Function secure_forward_packet(p)

Here PLGPa satisfies no-backtracking property. We define a network as a collection of nodes, a topology, connectivity properties, and node identities. Honest nodes can transmit and receive messages, while malicious nodes can also use directional antennas to transmit to (or receive from) any node in network. Honest nodes and malicious nodes can compose, forward, receive, or drop messages, and malicious nodes can also arbitrarily transform them. Adversary control nodes and have perfect knowledge about topology, so it cannot affect connectivity between any two honest nodes. All messages are signed by their originator. Adversary

Existing system PLGP	Proposed system PLGPa
PLGP does not have attestation.	It is PLGP with attestation.
Forwarding nodes doesn't know the path of the packet.	Each packet has a verifiable path history.
It does not hold Backtracking property.	It holds Backtracking property.
It is Vulnerable to Vampire attacks.	It is resistant to vampire attacks.
It is not enough secure.	It provides more security as compare to PLGP in both phases.
Packet size is small.	Packet size is increased due to attestation.
Extra packet verification is not takes place.	Extra packet verification is requiring.

can only alter packet fields that are changed on the way without authentication, so only the route attestation field can be altered, shortened, or removed entirely. To prevent truncation, use one-way signature chain construction. It allows nodes to add links to an existing signature chain, but not remove links, only append attestations. We must know about the hop count, the hop count of packet consider packet p, received or forwarded by an honest node, is no greater than the number of entries in p's route attestation field, plus 1. When any node receives a message, it checks that every node in the path attestation has a corresponding entry in the signature chain, and is logically closer to the destination than the previous hop in the chain. Forwarding nodes can broadcast progress of a message, and preserving no-backtracking. It follows the principles according to which PLGPa packet p satisfies no-backtracking in the presence of an adversary controlling $m < N - 3$ nodes if p passes through at least one honest node.

For above purpose, we will implement it by showing simulation results quantifying the performance of several representative protocols mentioned above in the presence of a single Vampire. Then, we will modify an existing sensor network routing protocol to provably bind the damage from Vampire attacks during packet forwarding. We also evaluate both the carousel and stretch attacks in a randomly-generated 30-node topology and a single randomly-selected malicious DSR agent, using the ns- 2 network simulator [1].

But our problem not solve completely so here we will also make it more secure by including secure discovery phase. We again modify PLGPa discovery phase, generally malicious node use directional antenna to misguide neighbor to pretend to be someone else node in the network and make a group of size one by merging them. By composing requested group ID as well as all the group members IDs we form merge request then flood this request to other group members by respective receiver node. Groups may generate signed tokens and Vampire makes it able to flood its group with every other group descriptor it knows.It use its directional antenna to investigates on broadcasts outside their neighbor range, sending merge requests from entirely honest groups. If another merge is in progress then denials can be occur in PLGP, so it is needed that modify the reject message by adding the ID of the group and a signature for non-repudiation with which the merge is in progress. This is stored and used at the end of topology discovery period then it is detected and removing nodes who incorrectly deny merge requests. Vampire rejects that proper merge request by continuing this one different group of all Vampires is formed. Vampires could maintain the illusion that it is a neighbor of a given honest group. After termination of topology discovery PLGP may provide a node who is a member of multiple groups will be detected once group join. PLGP detect active Vampires once the network converges and malicious behavior during discovery phase.

Our basic contributions to this project are that we thoroughly evaluate the vulnerabilities of existing protocols to routing layer battery depletion attacks. The study shows security measures to prevent Vampire attacks are different to those used to protect routing infrastructure, and so existing secure routing

protocols such as SAODV, and unable to protect against Vampire attacks. Existing work on secure routing attempts to ensure that adversaries cannot cause path discovery to return an invalid network path, but Vampires do not disrupt or alter discovered paths, instead it use existing valid network paths and protocol compliant messages. Protocols that maximize power efficiency are also inappropriate, since they rely on cooperative node behavior and cannot optimize out malicious action. And we will show simulation results in future and quantifying the performance of protocol in the presence of a single Vampire (insider adversary) also we modify an existing sensor network routing protocol to mitigate the Vampire attacks during packet forwarding which save ad-hoc wireless nodes from power drainage due to vampire packets.

To implement this we will use NS-2, it is n event driven packet level network simulator developed as a part of the VINT project (Virtual Internet Test bed) and used to study dynamic nature of network. The Defense Advanced Research Projects Agency (DARPA) supported development of NS through the Virtual Inter Network Test bed (VINT) project. Version 1 of NS was developed in 1995 and with version 2 in 1996 Ns-2 with C++/OTCL integration feature. Version 2, NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). It is an open source software package available for both Windows 32 and Linux platforms. NS-2 broadly used to evaluate the performance of existing network protocols and evaluate new network protocols before use. It reduces complexity of implementing protocols on large network.

5. CONCLUSION AND FUTURE WORK

In this seminar we first defined ad-hoc wireless sensor networks with Vampire attacks act as resource consumption attacks uses different types of routing protocols leading to slow depletion of node's battery life. Vampire attack is independent of specific protocol, so it is difficult to avoid it. We study number of attacks against existing routing protocols using weak adversaries in network. It increase energy usage can increase by as much as a factor of O(N) per adversary per packet, where N is the network size. We proposed defenses against some of the forwarding-phase attacks and study PLGP protocol find that it is vulnerable to Vampire attack, so makes some modification to it gives PLGPa with attestation the first sensor network routing protocol that provably mitigate Vampire attacks by verifying that packets consistently make progress toward destinations. We make again further modify PLGPa in order to securing its discovery phase. Ad hoc wireless sensor networks promise exciting new applications in the near future so it is necessary to increase battery power life. As there is not completely mitigation of Vampire attack, so more work should be possible by again make some modification to it in future.

REFERENCES

[1] The network simulator — ns-2. http://www.isi.edu/nsnam/ns/.
[2] Imad Aad, Jean-Pierre Hubaux, and Edward W. Knightly, Denial of service resilience in ad hoc networks, MobiCom, 2004.
[3] Mina Guirguis, Azer Bestavros, Ibrahim Matta, and Yuting Zhang, Reduction of quality (RoQ) attacks on Internet end-systems, INFOCOM, 2005.
[4] Jing Deng, Richard Han, and Shivakant Mishra, Defending against pathbased DoS attacks in wireless sensor networks, ACM workshop on security of ad hoc and sensor networks, 2005.
[5] INSENS: Intrusion-tolerant routing for wireless sensor networks, Computer Communications 29 (2006), no. 2.
[6] Sheetalkumar Doshi, Shweta Bhandare, and Timothy X. Brown, An ondemand minimum energy routing protocol for a wireless ad hoc network, ACM SIGMOBILE Mobile Computing and Communications Review 6 (2002), no. 3.
[7] Packet leashes: A defense against wormhole attacks in wireless ad hoc networks, INFOCOM, 2003.
[8] Bryan Parno, Mark Luk, Evan Gaustad, and Adrian Perrig, Secure sensor network routing: A clean-slate approach, CoNEXT, 2006.
[9] David R. Raymond, Randy C. Marchany, Michael I. Brownfield, and Scott F. Midkiff, Effects of denial-of-sleep attacks on wireless sensor network MAC protocols, IEEE Transactions on Vehicular Technology 58 (2009), no. 1.
[10] Yih-Chun Hu, Adrian Perrig, and David B. Johnson, Ariadne: A secure on-demand routing protocol for ad hoc networks, MobiCom, 2002.
[11] Volkan Rodoplu and Teresa H. Meng, Minimum energy mobile wireless networks, IEEE Journal on Selected Areas in Communications 17 (1999), no. 8.
[12] Rahul C. Shah and Jan M. Rabaey, Energy aware routing for low energy ad hoc sensor networks, WCNC, 2002.
[13] Anthony D. Wood and John A. Stankovic, Denial of service in sensor networks, Computer 35 (2002), no. 10.
[14] Daniel J. Bernstein, Syn cookies, 1996. http://cr.yp.to/syncookies.html