Wireless Ad hoc Sensor Network for Depletion of Node Battery Life for Denial of Service

V.Senthilvel ¹ and S.K.Srivatsa ²

¹ Research Scholar in Computer Science and Engineering, Dr.M.G.R University, Madhuravoyal, Chennai, India.
² Senior Professor, St.Joseph Engineering College, Chennai, India.

Abstract- Ad-hoc low-power wireless networks are an exciting research direction in sensing and pervasive computing. Prior security work in this area has focused primarily on denial of communication at the routing or medium access control levels. In this project I have focused on resource depletion attacks at the routing protocol layer, which permanently disable the networks by quickly draining node’s battery power. These “Vampire” attacks are not specific to any specific protocol, but rather rely on the properties of many popular classes of routing protocols. It is found that all the examined protocols are susceptible to Vampire attacks, which are devastating, 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. We discuss methods to mitigate these types of attacks, namely the carousel and stretch attacks with a new proof-of-concept protocol that provably bounds the damage caused by Vampires during the packet forwarding phase.

1. INTRODUCTION

The Wireless Sensor Networks (WSNs) is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors to monitor the physical and environmental conditions such as temperature sound, pressure, etc. These sensor nodes are spatially distributed. The more modern networks are bi-directional, also enabling control of sensor activity. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

Fig 1.1 Wireless sensor networks (WSN)
This paper makes three primary contributions. First, we thoroughly evaluate the vulnerabilities of existing protocols to routing layer battery depletion attacks. We observe that security measures to prevent Vampire attacks are orthogonal to those used to protect routing infrastructure, and so existing secure routing protocols such as Ariadne [29], SAODV [78], and SEAD [28] do not 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 using 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.

Second, we show simulation results quantifying the performance of several representative protocols in the presence of a single Vampire (insider adversary). Third, we modify an existing sensor network routing protocol to provably bound the damage from Vampire attacks during packet forwarding.

The wireless ad hoc sensor networks offer certain capabilities and enhancements in operational efficiency in civilian applications as well as assist in the national effort to increase alertness to potential terrorist threats.

Two ways to classify wireless ad hoc sensor networks are whether or not the nodes are individually addressable, and whether the data in the network is aggregated.

1.1 Attacks Focused
Here I focus on the major two types of attacks that are used for Denial of Service communication

1.2 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 Fig. 1.2 the thick path shows the honest path and thin shows the malicious path.

Fig. 1.2 shows the carousel attack same node appears in the route many times.

1.3 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. In the example given below honest path shown with thick lines and adversary or malicious path with thin lines. The honest path is very less distant but the malicious path is very long to make more energy consumption.

Fig 1.3 shows stretch attack where malicious path takes the longest route

Per-node energy usage under both attacks is shown in figure below

Fig 1.4 Node energy distribution under various attack scenarios.

As expected, the carousel attack causes excessive energy usage for a few nodes, since only nodes along a shorter path are affected. In contrast, the stretch attack shows more uniform energy consumption for all nodes in the network, since it lengthens the route, causing more nodes to process the packet. While both attacks significantly network-wide energy usage, individual nodes are also noticeably affected, destination.(4-9-10-11-12-8-9—long route), with some losing almost 10 percent of their total energy reserve per message.

2. RELATED WORK
The carousel attack can be prevented entirely by having forwarding nodes check source routes for loops. While this adds extra forwarding logic and it increase more overhead. The routes in link-state and distance-vector networks are built dynamically from many independent forwarding decisions. Adversaries have limited power to affect packet forwarding, making these protocols immune to carousel and stretch attacks. Using a directional antenna they can still waste energy by restarting a packet in various parts of the network. Another attack on all previously-mentioned
routing protocols is spurious route discovery. Every node will forward route discovery packets meaning it is possible to initiate a flood by sending a single message

2.1 Drawbacks Of The Existing System
1. Adding extra forwarding logic which increases the overhead.
2. Adversaries have limited power.
3. Usage of directional antenna.
5. Possibility of flooding.

3. PROPOSED SYSTEM
Here a clean slate secure sensor network routing protocol by Parno, Luk, Gaustad, and Perrig (“PLGP” from here on) consists of a topology discovery phase, followed by a packet forwarding phase, with the former optionally repeated on a fixed schedule to ensure that topology information stays current. In discovery phase each node has a limited view of the network, each node knows only itself and discovers their neighbors using local broadcast. Nodes build a tree of neighbor relationships and group membership that will later be used for addressing and routing. During the forwarding phase, all decisions are made independently by each node. When receiving a packet, a node determines the next hop. No-backtracking property, satisfied for a given packet if and only if it consistently makes progress toward its destination in the logical network address space.

3.1 Advantages Of The Proposed System
1. No forwarding logic or directional antennas are used.
2. Highly secured authentication.
3. High efficiency.
4. No flooding
5. Timely delivery of packets.

3.2 Clean-Slate Sensor Network Routing
In this section a clean-slate secure sensor network routing protocol by Parno, Luk, Gaustad, and Perrig (“PLGP” from here on) can be modified to provably resist Vampire attacks during the packet forwarding phase. The original version of the protocol, although designed for security, is vulnerable to Vampire attacks. PLGP consists of a topology discovery phase, followed by a packet forwarding phase, with the former optionally repeated on a fixed schedule to ensure that topology information stays current. (There is no on-demand discovery.) Discovery deterministically organizes nodes into a tree that will later be used as an addressing scheme. When discovery begins, each node has a limited view of the network — the node knows only itself. Nodes discover their neighbors using local broadcast, and form ever-expanding “neighborhoods”, stopping when the entire network is a single group. Throughout this process, nodes build a tree of neighbour relationships and group membership that will later be used for addressing and routing. At the end of discovery, each node should compute the same address tree as other nodes. All leaf nodes in the tree are physical nodes in the network, and their virtual addresses correspond to their position in the tree. All nodes learn each others’ virtual addresses and cryptographic keys. The final address tree is verifiable after network convergence, and all forwarding decisions can be independently verified. Furthermore, assuming each legitimate network node has a unique certificate of membership (assigned before network deployment), nodes who attempt to join multiple groups, produce clones of themselves in multiple locations, or otherwise cheat during discovery can be identified and evicted.

3.3 PLGP In The Presence Of Vampire
In PLGP, forwarding nodes do not know what path a packet took, allowing adversaries to divert packets to any part of the network, even if that area is logically further away from the destination than the malicious node. This makes PLGP vulnerable to Vampire attacks. Consider for instance the now familiar directional antenna attack: a receiving honest node may be farther away from the packet destination than the malicious forwarding node, but the honest node has no way to tell that the packet it just received is moving away from the destination; the only information available to the honest node is its own address and the packet destination address, but not the address of the previous hop (who can lie). Thus, the Vampire can move a packet away from its destination without being detected. This packet will traverse at most log N logical hops, with O((√2) ± ) physical hops at the ith logical hop, giving us a theoretical maximum energy increase of O(d), where d is the network diameter and N the number of network nodes.

3.4 Provable Security Against Vampire Attacks
First we introduce the no-backtracking property, satisfied for a given packet if and only if it consistently makes progress toward its destination in the logical network address space.

4. RESULT ANALYSIS
The architecture diagram below shows the relationship between different components of system. This diagram is very important to understand the overall concept of the system. Here each node acts as an individual node. The neighbour node is identified within the network. When a message is transmitted the node forwards a local broadcast message to its neighbor node. A tree is formed on identifying these neighbor nodes. After tree formation, the packet is forwarded to the shortest identified path among the formed tree with the property of no-backtracking algorithm.

4.1 Finding Neighbors
This module is used to identify the neighbor node within the network. When a message is transmitted this module forwards a local broadcast message to its neighbor node.
The neighbor nodes on receiving this broadcast message sends an acknowledgement to the sender node and from this acknowledgement message the sender node identifies the neighbor node. Only the neighbor nodes are involved in transmitting the message.

Discovery of nodes begins with a time-limited period during which every node must announce its presence by broadcasting a certificate of identity, including its public key (from now on referred to as node ID), signed by a trusted offline authority. Each node starts as its own group of size one, with a virtual address 0. Nodes who overhear presence broadcasts form groups with their neighbors. When two individual nodes (each with an initial address 0) form a group of size two, one of them takes the address 0, and the other becomes 1. Groups merge preferentially with the smallest neighboring group, which may be a single node. The groups act as individual nodes, with decisions made using secure multiparty computation.

### 4.2 Tree Formation
Each node after identifying its neighbor node forms a local tree structure. Similarly each node is within a network forms a local tree and a final tree is formed recursively when the tree structure reaches it convergence. Like individual nodes, each group will initially choose a group address 0, and will choose 0 or 1 when merging with another group. Each group member prepends the group address to their own address, e.g. node 0 in group 0 becomes 0.0, node 0 in group 1 becomes 1.0, and so on. Each time two groups merge, the address of each node is lengthened by one bit. Implicitly, this forms a binary tree of all addresses in the network, with node addresses as leaves. Note that this tree is not a virtual coordinate system, as the only information coded by the tree are neighbor relationships among nodes. Nodes will request to join with the smallest group in their vicinity, with ties broken by group IDs, which are computed cooperatively by the entire group as a deterministic function of individual member IDs. When larger groups merge, they both broadcast their group IDs (and the IDs of all group members) to each other, and proceed with a merge protocol identical to the two-node case. Groups that have grown large enough that some members are not within radio range of other groups will communicate through “gateway nodes,” which are within range of both groups. Each node stores the identity of one or more nodes through which it heard an announcement that another group exists. That node may have itself heard the information second-hand, so every node within a group will end up with a next-hop path to every other group, as in distance-vector. Topology discovery proceeds in this manner until all network nodes are members of a single group. By the end of topology discovery, each node learns every other node’s virtual address, public key, and certificate, since every group members knows the identities of all other group members and the network converges to a single group.

### 4.3 Transmission History
This module maintains a history of previous packet transmission done by each node. This history consists of the hop count of previous route and it keeps on updating this history through which attack immunity of our system can be improved.

### 4.4 Packet Forwarding
This module is used to transmit or route packet to nodes using the above formed tree structure. Here each node has independent route constructed from the tree structure. And also checks for the condition to match No Backtracking. During the forwarding phase, all decisions are made independently by each node. When receiving a packet, a node determines the next hop by finding the most significant bit of its address that differs from the message originator’s address (see Figure 4.4). Thus every forwarding event (except when a packet is moving withgroup in order to reach a gateway node to proceed to the next group) shortens the logical distance to the destination, since node addresses should be strictly closer to the destination.

![Fig 4.4 The final address tree for a fully-converged 6-node network. Leaves represent physical nodes, connected with solid lines if within radio range. The dashed line is the progress of a message through the network. Note that non-leaf nodes are not physical nodes but rather logical group identifiers.](image)

### 4.5 PLGP Satisfies No-Backtracking
To show that our modified protocol preserves the no-backtracking property, we define a network as a collection of nodes, a topology, connectivity properties, and node identities, borrowing the model used by Poturalski et al. in . Honest nodes can broadcast and receive messages, while malicious nodes can also use directional antennas to transmit to (or receive from) any node 11 in the network without being overheard by any other node. Honest nodes can compose, forward, accept, or drop messages, and malicious nodes can also arbitrarily transform them. Our adversary is assumed to control m nodes in an N-node network (with their corresponding identity certificates and other secret cryptographic material) and has perfect knowledge of the network topology. Finally, the adversary cannot affect connectivity between any two honest nodes. Since all messages are signed by their originator, messages from honest nodes cannot be arbitrarily modified by malicious nodes wishing to remain undetected. Rather, the adversary can only alter packet fields that are changed en route (and so are not authenticated), so only the route attestation field can be altered, shortened, or removed entirely. To prevent truncation, which would allow Vampires to hide the fact that they are moving a packet away from its destination, we use Saxena and Soh’s one-way signature chain construction [64], which allow nodes to add links to an existing signature chain, but not remove links, making attestations append-only.
NO-BACKTRACKING ALGORITHM

Begin
Send broadcast message Pk(B) for node detections
Identify neighbor nodes when receive an acknowledgement Pk(B)a
Creates tree structure using the neighbor node list <Tn>
After creation of tree each node share their tree structure <Tn>
Update individual tree structure <Tu>
Check for convergence
If convergence = true
Stop updating
Else
Begin
Share tree structure <Tu>
Update tree structure <Tu>
End
Maintains transmission history
Check Packet hop count Hn
If hop count (Hn) = Previous hop Count (Hp)
Drop packet
Else
Transfer packet
End

5.CONCLUSION
A large part of this phase, I have been dedicated to explain Vampire attacks, a new class of resource consumption attacks that use routing protocols to permanently disable ad-hoc wireless sensor networks by depleting nodes’ battery power. These attacks do not depend on particular protocols or implementations, but rather expose vulnerabilities in a number of popular protocol classes. I have defined the PLGP protocol the first sensor network routing protocol that provably bounds damage from Vampire attacks. I have explained the working procedure, algorithm and proving its efficiency theoretically by comparing it with the existing system models.

REFERENCES