Improving the Performance of Energy Attack Detection in Wireless Sensor Networks by Secure forward mechanism

Soram Rakesh Singh[*], Narendra Babu C R[**]

[*] Computer Science and Engineering, Rajiv Gandhi Institute of Technology, Bengaluru
[**] Computer Science and Engineering, Rajiv Gandhi Institute of Technology, Bengaluru

Abstract- Wireless ad-hoc sensor networks and routing data in them is a significant research area. The objective of this paper is to examine resource depletion attacks at the routing protocol layer, which attempts to permanently disable network nodes by quickly draining their battery power. This type of attack is called as vampire attack. Vampire attacks are very difficult to detect because they attack the node only by sending protocol-compliant messages. These attacks are not specific to any protocol, but rather rely on the properties of many popular classes of routing protocols. 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. Methods to detect and secure data packets from vampires during the packet forwarding phase is discussed. PLGP with attestations (PLGP-a) is used for identifying malicious attack. M-DSDV routing protocol is used to detect and eliminate the resource depletion attack from the network.

Index Terms- Denial of service, routing, ad-hoc networks, sensor networks, wireless networks, routing.

I. Introduction

Wireless Sensor Network (WSN) consists of mostly tiny, resource-constraint, simple sensor nodes, which communicate wirelessly and form ad hoc networks in order to perform some specific operation. Due to distributed nature of these networks and their deployment in remote areas, these networks are vulnerable to numerous security threats that can adversely affect their proper functioning. Simplicity in WSN with resource constrained nodes makes them very much vulnerable to variety of attacks. The attackers can eavesdrop on its communication channel, inject bits in the channel, replay previously stored packets and much more. An adversary can easily retrieve valuable data from the transmitted packets that are sent (Eavesdropping). That adversary can also simply intercept and modify the packets' content meant for the base station or intermediate nodes (Message Modification), or retransmit the contents of those packets at a later time (Message Replay). Finally, the attacker can send out false data into the network, maybe masquerading as one of the sensors, with the objectives of corrupting the collected sensors' reading or disrupting the internal control data (Message Injection). Securing the WSN needs to make the network support all security properties: confidentiality, integrity, authenticity and availability.

Attackers may deploy a few malicious nodes with similar or more hardware capabilities as the legitimate nodes that might collude to attack the system cooperatively. The attacker may come upon these malicious nodes by purchasing them separately, or by "turning" a few legitimate nodes by capturing them and physically overwriting their memory. Also, in some cases colluding nodes might have high-quality communications links available for coordinating their attack. The sensor nodes may not be tamper resistant and if an adversary compromises a node, it can extract all key material, data, and code stored on that node. As a result, WSN has to face multiple threats that may easily hinder its functionality and nullify the benefits of using its services.

Routing and data forwarding is a crucial service for enabling communication in sensor networks. Unfortunately, current routing protocols suffer from many security vulnerabilities. For example, an attacker might launch denial of-service attacks on the routing protocol, preventing communication. The simplest attacks involve injecting malicious routing information into the network, resulting in routing inconsistencies. Simple authentication might guard against injection attacks, but some routing protocols are susceptible to replay by the attacker of legitimate routing messages. The wireless medium is inherently less secure because its broadcast nature makes eavesdropping simple. Any transmission can easily be intercepted, altered, played by an adversary. The wireless medium allows an attacker to easily intercept valid packets and easily inject malicious ones. Although this problem is not unique to sensor networks, traditional solutions must be adapted to efficiently execute on sensor networks.

This paper makes three primary contributions. First, a thorough evaluation of the existing routing protocols towards battery depletion attacks is done. We observe that existing secure routing protocols such as Ariadne, SAODV, and SEAD 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 use existing valid network paths to carry out the attack. Protocols that maximize power efficiency are also inappropriate, since they rely on cooperative node behavior and cannot optimize battery power usage.
Second, simulation results quantifying the performance of several representative protocols in the presence of a single

Vampire (insider adversary) is shown. Third, modification of an existing sensor network routing protocol is made to prevent the damage caused by Vampire attacks during packet forwarding phase.

1.1 Classification

Denial of service is an attack, where a victim can use 10 minutes of the CPU time to transmit a data packet, but whereas an honest node uses 1 minute of its CPU time to transmit the same data packet. In multi hop routing network: a source composes the shortest path and transmits the data packet to the next hop, which transmits it further, until the destination is reached; consuming resources not only at the source node but also at every node the packet moves through. Vampire attack can be defined as a voluntary action of composing and transmitting a malicious message that chooses the longest path which consumes more energy of the network than if an honest node transmits a message of identical size to the same destination. The strength of an attack can be measured by the ratio of network energy used in the honest case to the energy used in the malicious case.

1.2 Protocols and Assumptions

In this paper, we consider the effect of Vampire attacks on Destination sequence distance vector routing protocols, as well as a logical ID-based sensor network routing protocol proposed by Parno et al. These protocols are likely to prevent Vampire attacks, so the covered protocols are an important subset of our routing solution space. We differentiate on-demand routing protocols, where topology discovery is done at transmission time, and static protocols, where topology is discovered during an initial phase, with periodic rediscovery to handle rare topology changes. The adversaries are malicious insiders and have the same resources and level of network access as honest nodes. Sending malicious packet automatically allows few Vampires to attack many honest nodes. We will show later that a single Vampire may attack every network node simultaneously, meaning that vampires are to be isolated from the honest nodes. Vampire attacks may be weakened by using groups of nodes with staggered cycles: only active-duty nodes are vulnerable while the Vampire is active; nodes are safe while the Vampire sleeps.

1.3 Overview

In the remainder of this paper, we present a series of increasingly damaging Vampire attacks, evaluate the vulnerability of several example protocols, and suggest how to improve flexibility. In source routing protocols, we show how a malicious packet source, can specify paths through the network, which are far longer than optimal, thus wasting energy at intermediate nodes that forward the packet as suggested by the source. In routing schemes, where forwarding decisions are made independently by each node (as opposed to specified by the source), 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. Lastly, we show how an adversary can target not only packet forwarding but also route and topology discovery phases—if discovery messages are flooded, an adversary can, for the cost of a single packet, consume energy at every node in the network.

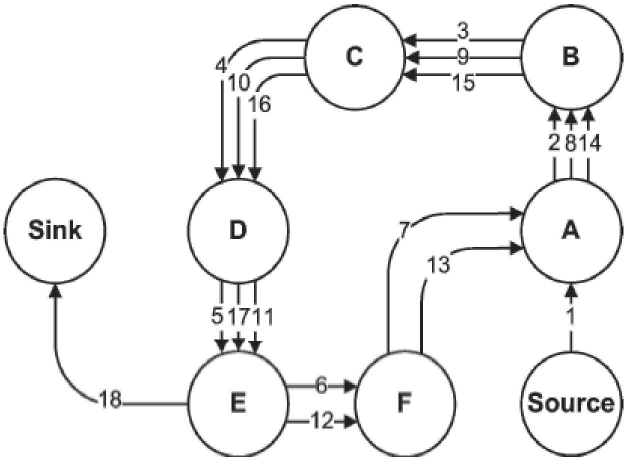


Figure 1: carousal Attack

CAROUSEL ATTACK:

In this type of attack, a malicious node sends a packet with a route composed as a series of loops with the same node appears in the route many times.

STRETCH ATTACK:

In this type of attack, a malicious node constructs artificially long routes from the source in spite of shorter routes being available. 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.

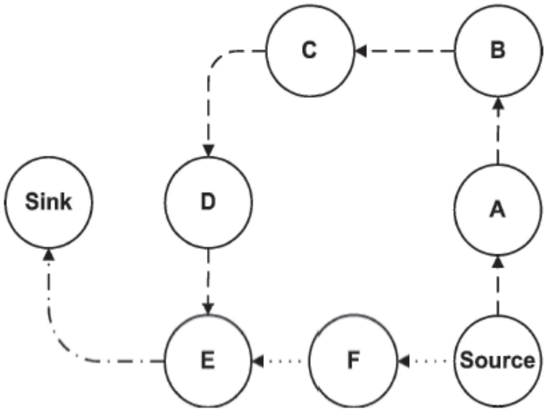


Figure 2: stretch attack

This attack causes the packets to be travelled a long route in the network. An adversary constructs artificially long routes, potentially traversing every node in the network.

Increase 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.

II. RESEARCH ELABORATIONS

SNEP Protocol

SNEP protocol was designed as basic component of another protocol SPINS (Security protocol for wireless Sensor Networks) that were basically designed for secure key distribution in wireless sensor networks. SNEP define the primitives for authentication of sensor node, data confidentiality and data integrity. However the drawback of this protocol is lower data freshness. SNEP protocol uses shared counter for semantic confidentiality not initial vectors. Using SNEP the plain text is ciphered with CTR encryption algorithm. Both sender and receivers are responsible to update the shared counter once when they sent or receive cipher blocks. Therefore sending counter in message is not important; however every message has message authentication code (MAC). This is computed from cipher data with the help of CBC-MAC algorithm. When the receiver node receives data it recomputed MAC and compared with the received MAC.

REWARD

Z. karakehayou proposed a new algorithm know as REWARD for security against black hole attack as well as malicious nodes. It works on geographic routing. There are two different kinds of broadcast messages used by REWARD.

MISS message helps in the identification of malicious sensor nodes. While the second message SAMBA is used to recognize the physical location of detected black hole attacks and broadcast that location. REWARD uses broadcast inter radio behavior to observe neighbor node's transmission and detect black hole attack. Whenever any sensor misbehaves it maintain a distributed database and save its information for future use. However the main drawback of this protocol is high energy consumption.

Statistical En-Route Filtering

F. Y. Haiyon et al present a statistical en-route filtering technique to control attacks on compromised sensor nodes, where a compromised node can easily inject wrong report in the network that cause depletion of finite resources at sensor nodes as well as causes false alarms. Statistical En-Route Filtering is able to detect and destroy such false reports in the network. For this purpose message authentication code (MAC) is used to check the validity of each message. When sensed data is forwarded toward sink node each node in the middle verify that message. Statistical En-Route Filtering relies on collective information from multiple sensor nodes. When an event occurs the sensor nodes in the surrounding collectively generate a legitimate report that carries multiple message authentication codes (MAC's). The report is forwarded toward sink node and each node in the middle verifies the report with certain probability, when the report is found incorrect it is dropped. The probability of message incorrectness increases with number of hops. In many cases a false report may reaches to a sink node where sink node will be responsible to verify it again. However this approach causes delay as well as increase communication overhead and energy consumption in resource limited networks.

The effect of denial or degradation of service on battery life and other finite node resources has not generally been a security consideration, making our work tangential to the research mentioned above.

SYSTEM METHODOLOGY

The network is composed of multiple nodes. An energy based mechanism to detect the Vampire Attacks is implemented. Once we constructed a network, the malicious message will be send from the attacker node to any of the normal node. So that the normal node's energy will be consumed more than the normal message level So that we can conclude that the node is affected by the attack. Once the node is identified as the attacked node, the node is eliminated from the network. Hence the attacked node is not able to communicate with the other nodes in the Network. It uses one-way hash chains to limit the number of packets sent by a given node, limiting the packet transmission rate. Energy usage by malicious nodes is to be reduced, since they can always unilaterally drain their own batteries.

The proposed system containing two important technologies, they are

PLGP:

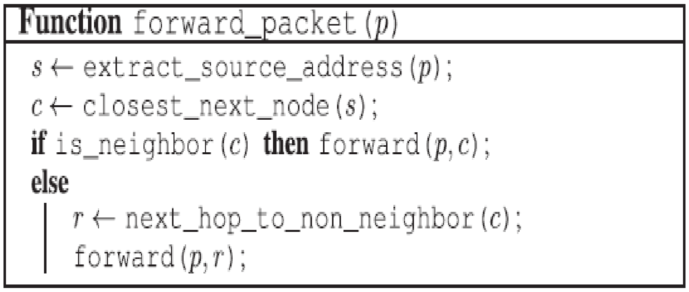
PLGP is a clean-slate secure sensor network routing protocol which is used to detect the vampire node. PLGP consists of **two** levels:

a) Topology Discovery Phase
b) Packet Forwarding Phase

Topology Discovery Phase: Discovery phase organizes nodes into a tree that will later be used as an addressing scheme that is repeated on a fixed schedule and 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 that is 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 neighbor relationships and group membership that will later be used for addressing and routing.

Packet Forwarding Phase: In this 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. Thus, every forwarding event shortens the logical distance to the destination, since node addresses should be strictly closer to the destination.



```
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);
```

PLGP WITH ATTESTATIONS (PLGP-a) Phase:

The verifiable path history is added to every PLGP packet. The resulting protocol, PLGP with attestations (PLGP-a) uses this packet history together with PLGP's tree routing structure so every node can securely verify progress, preventing any significant adversarial influence on the path taken by any packet which traverses at least one honest node. 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 traveled away from its destination in the logical address space.

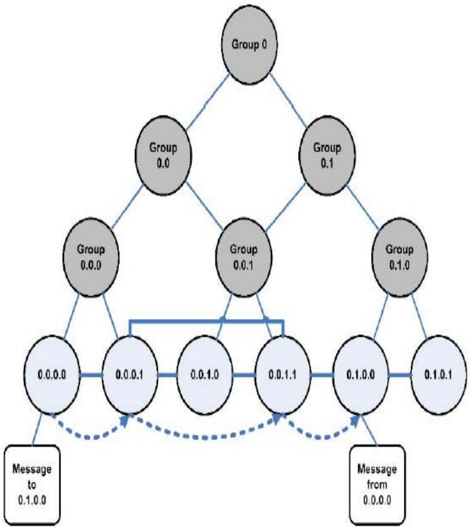
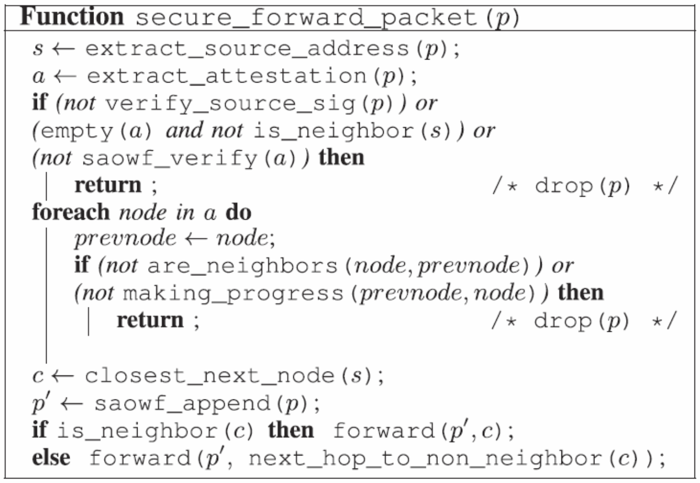


Fig. 6. The final address tree for a fully converged six-node network. Leaves represent physical nodes, connected with solid lines if within radio range. The dashed line is the progress of a message through the network. Note that nonleaf nodes *are not physical nodes* but rather logical group identifiers.

Packet forwarding PLGP-a



```
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) */
  foreach 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));
```

M-DSDV NETWORK ROUTING: In this section, we show that destination sequence distant vector a proactive network routing protocol [24] can be modified to provably resist Vampire attacks during the packet forwarding phase. Even though the existing DSDV is designed to overcome routing loop problems, it is still not a feasible method for efficient packet transmission, as the protocol is proactive which utilizes more battery power and bandwidth. M-DSDV consists of a topology discovery phase, followed by a topology maintenance phase. Legitimate network node has a unique certificate of membership, which includes its public key and code word (assigned by a trusted offline authority before network deployment). Topology Discovery of the neighboring nodes begins, when there is a need to transmit the data packet.

Each node has a limited view of the network—the node knows only itself. Nodes use the local broadcasting scheme to discover their neighbors', where the certificate identity verification is done to isolate the external unauthorized nodes from the network. Thus, each honest node learns its active neighbor node's address and public key.

When a source node S, wants to send a data packet to destination D, first constructs and broadcasts a route request packet consisting of (source address, destination address, sequence number, next hop, metric, index number and time to live) fields. The source address and destination address are the internet protocol addresses, the sequence number is used to differentiate new routes from stale routes, the next hop and metric is a local counter maintained separately by each node and incremented each time a RReq is broadcasted, the index number is initialized to zero, is used to keep track of the loops the packet has made and the final time to live field is used as a clock which increments whenever a RReq packet is sent.

On receipt of RReq, intermediate nodes inspect it to see if it is a duplicate, in which case it is rejected. If not the (source address, next hop, metric) pair is entered into the local history table. The destination address is looked up in the routing table, if a fresh route to it is known an RRep a route reply packet is sent back to S. If not, it increments the index number and rebroadcasts the RReq. This also creates a backward route towards S and exists has an optimization technique.

When destination receives RReq, it sends back an RRep packet to the node from which it got the first RReq packet.

The format of the route reply packet includes (source address, destination address, destination sequence, index number, life time). Here, the source address, destination address and index number are copied from the incoming RReq packet, but the destination sequence number is taken from its counter in memory. The life time field indicates how long the route is valid.

On receipt of RRep, intermediate nodes on the way back, inspect the packet and create a backward route towards destination.Intermediate nodes that got the original RReq packet but were not on the reverse path discard the reverse route table entry when the associated timer expires. When the next hop link in the routing table entry breaks, all active neighbors' are informed by means of RERR packets which updates the sequence number. RERR packets are also generated when anode X is unable to forward packet P from node S to node D on link (X, Y). The incremented sequence number N is included in the RERR. When node S receives the RERR, it initiates a new route discovery for D using the sequence number that is at least as large as N.

In the presence of vampires, carousel attack and stretch attack can be prevented by using the index number. In case of, carousel attack, where a packet which traversed through the shortest path of the network, returns back again to the same node, that could be eliminated by checking the index number stored on the packet header and the index number stored in the local routing table of the node.

We can prevent the stretch attack by independently checking on the packet progress: the nodes keep track of route “Metric” and, when acknowledgement returns back, the route metric value and the index number, which indicates the hop count, can be verified. If the index value is greater than the metric value the source concludes that the stretch attacks as occurred.

Moreover, to prevent truncation of the routing path, 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, which allow nodes to add links to an existing signature chain, but not remove links, making attestations append only. Thus if malicious intervention has been suspected the packet is dropped from further forwarding strategy. Thus, the damage from an attacker is bounded as a function of network size.

IIICONCLUSION

Vampire attacks has been defined as a new class of resource consumption attacks that use routing protocols to permanently disable ad hoc wireless sensor networks by depleting nodes’ battery power. Defenses against some of the forwarding-phase attacks has been proposed and PLGP-a, the first sensor network routing protocol that reduces the damage from Vampire attacks by verifying that packets consistently make progress toward their destinations. The routing protocol has been used at the time of routing to make efficient energy utilization during the packet forwarding phase’s-DSDV, routing protocol that provably bounds damage from Vampire attacks by verifying that packets consistently is proposed in this paper. Prevention of data packets from entering into a malicious node is left for future work.

REFERENCES

[1] Eugene Y. Vasserman and Nicholas Hopper,“Vampire attacks: Draining life from wireless adhocsensor networks”, IEEE Transaction on Network Security for Technical Details, June 17,2013.

[2] J.H. Chang and L. Tassiulas, “Maximum Lifetime Routing in Wireless Sensor Networks,”IEEE/ACM Trans. Networking, vol. 12,no. 4, pp.609-619, Aug. 2004.

[3] INSENS: Intrusion-tolerant routing for wireless sensor net- works, Computer Communications 29 (2006), no. 2.

[4] S. Doshi, S. Bhandare, and T.X. Brown, “An On-Demand Minimum Energy Routing Protocol for a Wireless Ad Hoc Network,” ACM SIGMOBILE Mobile Computing and Comm. Rev.,vol. 6, no. 3, pp. 50-66, 2002.

[5] A.D. Wood and J.A. Stankovic, “Denial of Service in Sensor Networks,” Computer, vol. 35, no. 10, pp. 54-62, Oct. 2002.

[6] J. Deng, R. Han, and S. Mishra, “Defending against Path-Based DoS Attacks in Wireless Sensor Networks,” Proc. ACM

Workshop Security of Ad Hoc and Sensor Networks, 2005.

[7] L.M. Feeney, “An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks,”

Mobile Networks and Applications, vol. 6, no. 3, pp. 239-249, 2001

[8] G. Yang, M. Gerla, and M.Y. Sanadidi, “Defense Against Low-Rate TCP-Targeted Denialof- Service Attacks,” Proc. Ninth Int’l Symp. Computers and Comm. (ISCC), 2004.

[9] J.R. Douceur, “The Sybil Attack,” Proc. Int’l Workshop Peerto- Peer Sys.

[10] L.M. Feeney, “An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks,”

Mobile Networks and Applications, vol. 6, no. 3, pp. 239-249,2001..

AUTHORS

First Author – Soram Rakesh Singh,,has received his B E degree in Computer Science and Engineering from RLJIT VTU university in 2012. He is pursuing M.Tech in Computer Science and Engineering from Rajiv Gandhi Institute of Technology Bengaluru.

Second Author – Narendra Babu C R, Asst Professor, in Dept. of Computer Science and Engineering at Rajiv Gandhi Institute of Technology, Bengaluru.