Tattle Tale Security: An Intrusion Detection System for Medical Body Area Networks (MBAN)

Lanier Watkins*, Shreya Aggarwal*, Omotola Akeredolu*, William H. Robinson† and Aviel Rubin*
*Johns Hopkins University, USA
†Vanderbilt University, USA

***Abstract*—Medical Body Area Networks (MBAN) are created when Wireless Sensor Nodes are either embedded into the patient's body or strapped onto it. MBANs are used to monitor the health of patients in real-time in their homes. Many cyber protection mechanisms exist for the infrastructure that interfaces with MBANs; however, not many effective cyber security mechanisms exist for MBANs. We introduce a low-overhead security mechanism for MBANs based on having nodes infer anomalous power dissipation in their neighbors to detect compromised nodes. Nodes will infer anomalous power dissipation in their neighbors by detecting a change in their packet send rate. After two consecutive violations, the node will "Tattle" on its neighbor to the gateway, which will alert the Telemedicine administrator and notify all other nodes to ignore the compromised node.**

I. Introduction

The Internet of Things (IoT) Healthcare market (i.e., telemedicine) is predicted to reach $14 Billion by 2022, and remote patient monitoring is the primary application. The core technology behind remote patient monitoring is Medical Body Area Networks (MBANs) that provide real-time health status of patients. Wearable devices, such as pacemakers, stress sensors, and insulin sensors, measure and relay vital information to the health care provider [4]. Many cyber protection mechanisms exist for the infrastructure that interfaces with MBANs; however, not many effective cyber security mechanisms exist for MBANs. Some key reasons are that MBANs have restricted resources (e.g., CPU, memory, power, and communications) [1], and the physical location of the nodes in a MBAN may not be readily accessible (e.g., implanted inside the body). Those mechanisms that do exist tend to be high-overhead and thus not a feasible solution for real MBAN hardware. We introduce a low-overhead security mechanism that is based on having nodes infer anomalous power dissipation in their neighbors to detect compromised nodes in the MBANs. Our logic is that for a given a MBAN composed of homogeneous nodes arranged with fairly equal distances apart, and that use the same sensor reporting cycles, their power dissipation should be very similar according to the Wireless Sensor Nodes First Order Radio Model. Similarly, any active communications (bluetooth, zigbee, or Wi-Fi) to a node outside of the MBAN (especially a significant distance away) will likely result in detectable power dissipation. To demonstrate the feasibility of this approach to detect threats in MBANs, we implement our method as an intrusion detection system (IDS) using the hardware operating system Contiki (with node simulator Cooja). Then, we illustrate the step-by-step behavior of our IDS when subjected to denial of service, node subversion, and replay attacks.

II. Wireless Sensor Nodes for MBANs

Wireless Sensor Nodes are resource-constrained devices that can form an ad-hoc network and are often used to monitor systems that are remote or hard-to-reach. These nodes usually gather certain sensory data and forward it along toward the gateway node through its nearest neighbor (multi-hop routing). MBANs can consist of several wireless sensor nodes (as opposed to 10s or 100s of nodes in a traditional wireless sensor networks), such as pacemakers, blood pressure sensors, or motion sensors, which relay information about the part of the human body where they are positioned. MBANs are particularly useful as home-based patient monitoring systems [8]. Patients no longer need to stay at hospitals or clinics where traditional medical devices may be used for monitoring. This scenario allows for increased mobility and comfort for the patient while the doctor monitors the patient's condition remotely [8]. Another application of MBANs would be to transmit information of the patient's vitals from an ambulance to the hospital emergency room [9]. Paramedics can attach the MBAN to the accident victim to start collecting information about the patient's condition even before the patient arrives at the hospital [9]. This scenario would allow the doctors at the hospital to better prepare for the patient's arrival and also save valuable time in assessing the patient's condition. Given the nature of MBANs (health monitoring) and proximity to vital organs of the body, such as the heart, a security threat to any of these nodes could result in critical health conditions for the patient. This work is motivated by [2], which theoretically discuss the use of wireless sensor network traffic for security. Below are other methods that take similar theoretical approaches. Our work differs in several ways from all of these method, but most notably in that we use a hardware operating system (Contiki) and a node simulator (Cooja) to produce more realistic results.

A. Adaptive Intrusion Detection

The authors in [5] propose a Self-Adaptive Intrusion Detection (SAID) system that is capable of adapting to different threats in a wireless sensor network. The authors aim to bridge the limitations that other wireless sensor network security proposals which focus on encryption and authentication have.

Workshop on Decentralized IoT Systems and Security (DISS) 2019
24 February 2019, San Diego, CA, USA
ISBN 1-891562-56-8
https://dx.doi.org/10.14722/diss.2019.23004
www.ndss-symposium.org

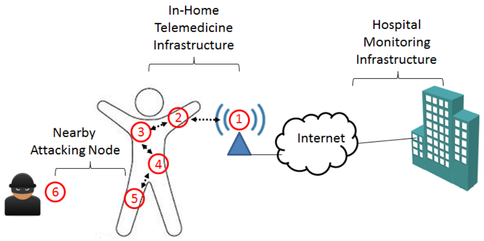


Fig. 1. Proposed Telemedicine Scenario

The system consists of a three-logic-layer architecture that works together to detect both known and unknown attacks. Like an immune system, SAID is capable of learning different pathogens and manipulating its internal algorithms and heuristics to defend against the new attacks. Its adaptive nature also makes it deployable in a limited resource network as the algorithms can be combined in such a way to improve performance. In a resource-rich environment, complex-intrusion rules can be stored in the system to defend against specific attacks.

B. *Lightweight Intrusion Detection*

Several proposals have been made with a focus on lightweight security in wireless sensor networks, such as the authors in [6] who propose an intrusion detection system that uses an over-hearing mechanism to decrease the number of alert packets sent. The system is set up in such a way that each node in the network has two intrusion modules: a local and a global agent. The local agent monitors information sent and received by that node while the global agent monitors communication by neighbors within the node's communication range using two-hop neighbor knowledge. Given the broadcast nature of node communications, nodes can receive all packets sent by neighbors within their radio range making it possible for the global agent to detect external anomalies for a given node. The use of two-hop neighbor knowledge helps to reduce the number of transmissions within the network. The authors in both [5] and [6] claim that their methods are applicable to resource limited wireless sensor networks; however, both methods describe a very complex set of rules that are likely not feasible for real hardware nodes.

C. *Encryption in BAN*

Given the nature of information being transmitted in a MBAN, such as Private Identifiable Information (PII) of patients, data integrity is extremely important. Security measures have focused on keeping data confidential within these networks. For example, the authors in [7] propose a security suite that uses two key management schemes to secure inter-sensor communication within a BAN as well as to secure communication with the monitoring station. Given the security threat of eavesdropping that lies in key sharing/exchange, the security scheme uses a randomly generated key independently at both sender and receiver whenever encrypting a packet. The security suite also has a focus on simplicity making it suitable to be deployed in limited resource sensor networks.

III. Threat Model for MBANs

According to the First Order Radio Model [10], when one node uses its transceiver to communicate with another node, its battery power level dissipates mostly due to the square of the distance between the communicating nodes. Note that the distance between legitimate nodes of the MBAN are likely small as compared to remote MBAN threats. The legitimate power usage in our approach causes all nodes to duty-cycle at the same rate, but nodes that have been compromised will duty-cycle too fast and be detected by their neighbors. We focus on three practical MBAN threat models [15] [11]: (1) node capture attacks, (2) replay attacks, and (3) denial of service attacks.

A. *Node Capture Attacks*

Node Subversion or node capture is a type of attack in which a threat is able to capture and take control of a node. During authentication, the node may reveal some cryptographic keys which the threat can use to capture the node [15]. Once captured, messages from that node can be manipulated before being sent to other nodes, thus compromising the whole network. A real-life threat scenario could unfold as follows, an attacker with direct access to the patient could sabotage the health of the patient in some way, and then use the captured node to send false health information to the remote monitoring service and thus the patient may expire without any medical professional knowing there was ever an issue. Our IDS should detect this threat, since the communications between legitimate nodes in the MBAN and the attacking node would cause unanticipated battery dissipation in the legitimate nodes, which would likely be detected once the compromised nodes attempt to communicate with their neighbors.

B. *Replay Attacks*

In a replay attack, the threat eavesdrops on the network and capture packets in transit. Then the attacker replays the packets to a legitimate node (its target) in the network. Because nodes in MBANs have limited resources, they likely do not use cryptographic nonces (randomly generated numbers used to ensure packets can not be re-used) to defend against replay attacks. This makes replay attacks very practical threats to MBANs [11]. These attacks can affect data freshness and disorganize the packet frames in the network [3]. A real-life threat scenario could unfold as follows, an attacker with only proximity access to a patient could remotely capture MBAN packets from a healthy patient, then later when the patient's health fails, continuously replay the old packets (from when the patient was healthy) to mask the failing health of the patient until the patient expires. Replaying the packets to another node would result in unanticipated use of the node's transceiver and thus cause unanticipated power dissipation, which would likely be detected once the compromised nodes attempt to communicate with their neighbors.

C. *Denial of Service*

A denial of service (DoS) attack occurs when a threat is able to inundate a node or make it become unresponsive, resulting in a disruption in the network. The threat could either send several packets or a very large packet to a node in the network which overloads the node and causes it to die or become unresponsive [11]. The loss of a node means that no sensory data from one part of the body will ever reach the gateway, but more importantly, if the node is implanted inside of the patient's body there could be even more adverse effects. A real-life threat scenario could unfold as follows, an attacker outside of the patient's room or in the next apartment could wage a DoS attack against an implanted node from the patient's MBAN by making excessive connections (even if connections are denied) to the implanted node. This would eventually result in the implanted node dying and thus requiring a surgical removal and re-implant (once batteries are replaced). This attack will likely be detected once the compromised nodes attempt to communicate with their neighbors or fail to communicate with their neighbors.

IV. Experimental Evaluation

A. MBAN Design

In an effort to provide a realistic MBAN implementation and provide realistic attacks and responses to attacks, we used Contiki, a wireless sensor network operating system used by many researchers previously [12] [13] [14] with node simulator Cooja. Our implementation includes: (1) Node Discovery, (2) Multi-Hop Routing, (3) Battery, (4) Dynamic Duty Cycling, (5) Packet Layout, (6) Security Algorithm, and (7) Threat Models.

Note, all of the parameters used in this paper are for demonstration purposes and would need only to be changed to fit the specific need of the specific MBAN. Also, real nodes such as Heart Rate, ECG, or Body Temperature monitors would have different monitoring and reporting cycles; however, in our approach we fix all nodes to the minimum monitoring and reporting cycle of all of the nodes. For this example, all nodes in the MBAN would monitor and report on the cycle for ECG. This way, the overall lifetime of the MBAN is maximized and all nodes still meet their minimizing reporting cycle. This is accomplished by having each node duty cycle (See Duty Cycle Section below). Specifically, sacrificing the precision of the nodes' measurements instead of the monitoring and reporting frequency. For instance, all nodes may start out reporting 8 digit measurements such as "12345678" with their appropriate units, but when duty cycling occurs, the nodes may start to report "1234" and later maybe "12." Each time the precision get lower, the nodes dissipate less battery power, thus extending the lifetime of the nodes in the MBAN.

1) Node Discovery: In our MBAN network, every node starts out clueless about what other nodes exist in the network. Like in real MBANs, every node must discover the nodes around it in order to be able to route messages correctly. Each node announces its existence to every other node in the network. When an announcement is received from a neighbor, the node will take note of its distance from that neighbor. Once each node is aware of all the other nodes in the network, they can start routing messages to the neighbors with the least distance.

$$E_{transmit} = L * P_s(E_{tx/rc} + E_{fs} * d^2) \tag{1}$$

$$E_{receive} = L * P_s(E_{tx/rc}) \tag{2}$$

2) Multi-Hop Routing: When a node sends a packet, it is forwarded from one neighbor to the next until it reaches the gateway since sending packets to the gateway directly will dissipates power too quickly for nodes further away (from the gateway). Based on our design, each node sends a message to its neighbor only once per round and each node has equal send rates. Also, in MBANs the nodes are arranged fairly close together (Figure 1). All of these factors minimize power dissipation for normal MBAN operations and it also extends the overall lifetime of the MBAN. For example, in the first round, all nodes send their sensory data to their one-way neighbor except node 2 (its neighbor is the gateway). In round 2 and every round after that node 2 sends all nodes' sensory data to the gateway.

3) Battery: Energy levels for each node are calculated using the First Order Radio Model (Equations 1 and 2) for transmitting and receiving packets [2]. Transmission or receive energy (Etx/rc) and free space energy (Efs) are typically held constant while packet send rate (Ps), distance (d), and packet length (L) vary based on duty cycle, location of nodes, and length of payload, respectively. More energy is consumed by the node as these variables increase in value. For example, sending a packet to a node that is a distance of 5 meters away consumes more energy than sending to a node that is a distance of 2 meters away. Again, in our design, the nodes are fairly close together and they all have the same send rate. This ensures that the normal operation of the MBAN causes each node to dissipates power equally. The side-effects of these characteristics is an extension of the overall lifetime of the MBAN.

4) Dynamic Duty Cycling: Duty cycling is a way to extend a node's battery life by managing the send rate of each node based on its battery level. As battery level decreases and falls into one of the energy ranges in Table 1 (these ranges are customizeable), packet send rate decreases based on the appropriate energy range to conserve battery power and overall MBAN lifetime. The length of the packet that a node sends has a tremendous effect on battery dissipation. A long packet causes a node to consume much more energy than a short packet. As an example, when a node's battery level is between 68% to 84% of the initial energy level, the duty cycling is at 35.5% which means that the length of the payload is divided into half. Another way to implement this could be to have the node to wait out a round before sending data. Either way, battery dissipation would be reduced by more than half since its own message and also the messages it receives from its peers is being cut in half (or again by waiting out a round). The goal of duty cycling is for the node to live longer and use its battery more efficiently. Essentially, the node can communicate less often and keep precision or communicate with the same frequency but reduce sensor reading precision. In other words, if the nodes started with a 4 digit representation for sensor data and then it duty cycled down to 2 digits to save battery power, the nodes would round its sensor readings to 2 digits and never try to send the remaining 2 digits.

5) Packet Layout: In the routing layer, each node sends a packet per round which causes it to dissipate energy. When a node receives a packet, it attaches its own message to the packet and forwards the concatenated packet to the next neighbor. As shown in Figure 2, each node's message consists

TABLE I. DUTY CYCLE LEVELS

Duty Cycle %	Packet Send Rate	Energy Range
100	Po	E(n) >= 0.84*Eo(n)
35.5	Po/2	E(n) < 0.84*Eo(n) and >= 0.68*Eo(n)
11.5	Po/4	E(n) < 0.68*Eo(n) and >= 0.52*Eo(n)

of its payload, which contains the sensory data it collects, and the node's overhead, which contains information about node compromise, its own ID, and a message ending indicator, in that order. Each node sends the same size packet to ensure that each node dissipates the same amount of power during multi-hop communications. In normal operations, each node will duty cycle together and thus every node will continue to to communicate using the same packet layout. When a node dissipates power unexpectedly, there is a mismatch in packet layout between nodes and this helps neighbors identify (infer) node compromise. Each node has data that it needs to send to the gateway, but there must also be control data embedded into the packet layout (overhead) to support our security mechanism. For example, a full packet received at the gateway for a 4 character payload and a duty cycling level of 100% could be (1) ”22220N2f33330N3f44440N4f55550N5f” during normal operation and it could be (2) ”22223C2f33330N3f44440N4f55550N5f” during an attack. Note the packet in (1) is 32 characters long, because there are 4 nodes sending an 8 character message (4 characters data and 4 characters overhead). From a security perspective, this packet ensures the gateway that no threat was detected; however, the packet in (2) contains the ”C” flag and denotes that node 2 thinks that its neighbor node 3 is compromised. As battery levels decrease, the nodes start to duty cycle and change their packet send rates. Since the initial length of the payload is 4 characters, when duty cycling is at 11.5%, the length of the payload is 1 character whereas the overhead is still 4 characters long. Therefore the packet length has been reduced from 32 characters to 20 characters. After a node's own message, the packet that it received from its neighbor is concatenated to the end. If all of these messages do not take up all of the space within the packet, fillers (”F”) are concatenated to the end so that each node sends the same packet length which allows every node's battery level to stay consistent with its neighbors.

6) “Tattle Tale” Security Algorithm: “Tattle Tale” Security, as the name implies, is a term we use to describe our process of requiring nodes to infer anomalies in their neighbor and report (“Tattle”) their neighbor to the gateway (Node 1). As mentioned in the previous sections, our wireless sensor network is set up with nodes fairly close together and they all start out with the same battery levels and duty cycle rate (keeps the send rates across the network equal). Thus, each node can infer anomalies is their neighbor by comparing its own send rate to its neighbor's send rate. If a difference exists, the node notifies (“Tattles”) the gateway that its neighbor is compromised. To be clear, the legitimate power usage in our approach causes all nodes to duty-cycle at the same rate, but nodes that have been compromised will duty-cycle too fast and be detected by their neighbors. The gateway then uses its detection logic to confirm the compromise (two reports of compromise), after which a broadcast message is sent to notify all nodes to avoid the compromised node when routing packets.

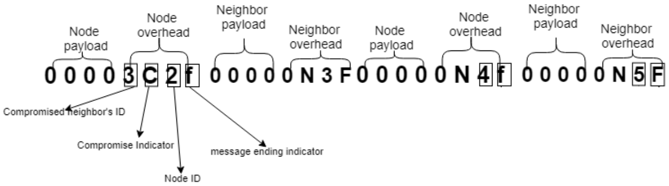


Fig. 2. Packet Layout: Indicating Node Compromise

7) Threat Models: Usually, a threat to an MBAN will consist of a device or node that can communicate with nodes inside the network to execute active attacks. Passive attacks may also be performed where the adversary listens in on the network and steals the patient's personal health information, but these are outside of our scope. The adversary must be in proximity to the nodes to be able to communicate with them, unlike the gateway which can be accessed through other means. We considered three threat models: (1) node capture attack, (2) denial of service attack, and (3) replay attack. Because the power dissipation for a wireless sensor node is directly proportional to the square of the distance between the nodes in question, any active attack will cause the battery power level of legitimate nodes to dissipate, the further away the attacker the greater the dissipation. Some attacks are harder to wage than others. The node capture attack is likely the hardest, because the attacker must gain complete control of the node. Next is the replay attack since the attacker must capture packets from legitimate communications first, then the attacker must transmit these captured packets to the target node. Finally, the DoS attack is likely the easiest, because an attacker only needs to repeatedly transmit packets to the node with the goal of forcing the node to exhaust all of its battery life on receiving packets from the attacker or responding back. This could be easy because of the limited resource nature of MBANs, and they likely do not have any mechanisms in place to ignore excessive requests to connect to it.

B. Experimental Setup

We implemented our MBAN IDS in the wireless sensor network operating system Contiki (with simulated hardware nodes using Cooja) [12] [13] [14]. Our experimental testbed is comprised of six simulated hardware nodes. Five (nodes 2 - 5) of the six are arranged in a MBAN representation as illustrated in Figure 1, while node 6 is set apart and used as an adversary node. Every node is positioned equidistant from its neighbor and starts out with the same battery power level. Each node sends a packet once per round. Energy dissipates every round based on the First Order Radio Model (Equations 1 and 2) for sending and receiving packets. The gateway (node 1) has unlimited power (wired power source), and thus its energy source does not dissipate like other nodes. Packets are sent sequentially from node 5 all the way to the gateway (i.e., static routing, 5 - 4 - 3 - 2 - 1). As power levels decrease, nodes start to duty cycle based on Table 1. Each node is capable of detecting a compromise except node 5, which has no neighbors.

C. Experimental Procedure

At the start of the simulation, every node receives announcements from all of its neighbors in the network which

TABLE II. Comparison With Other Methods

Method	Description	Simulation	Threat Model
”Tattle Tale”	Anomalous Power Detection	Contiki	Active Attacks
Ma et al. [5]	Agent Based, Two-Hop	NS2	Routing Attacks
Hai et al. [6]	Major Voting, Two-Hop	Castalia	Routing Attacks
Sampangi et al. [7]	Encryption	Java Code	Key Attacks

is used to build a neighbor list. The nodes then begin to route messages through their nearest neighbor. Since all nodes start out with equal battery, their power levels decrease similarly based on Equation 1 and the nodes duty cycle at the same time based on Table 1. We then introduce the threat, node 6 (Figure 1).

1) Node Subversion in Contiki: To simulate a node capture attack, node 6 must authenticate itself by sending a greeting packet to node 3. Node 3 will realize it received a message from outside the network and will send back a packet asking for a password. This extra communication causes nodes 3’s power level to decrease, causing it to duty cycle faster than the other nodes. Node 2 detects that node 3’s send rate is different from its own send rate, so it reports its suspicion of trouble to the gateway using the packet structure described in Figure 2. After the gateway receives this report twice from node 2, it concludes that node 3 is compromised and sends a broadcast message to all nodes, except the compromised node, to ignore node 3. As a result of this broadcast, node 4 updates its routing and sends packets to node 2 instead of node 3. Node 3 still continues to send packets, but they are ignored.

2) Replay Attack in Contiki: To simulate a replay attack, node 6 replays old packets to node 4. Nodes 4’s power level decreases causing it to duty cycle faster than the other nodes. Node 3 detects this by noticing node 4’s send rate is different from its own send rate. It reports this to the gateway. After the gateway receives this report twice from node 3, it concludes that node 4 is indeed compromised and sends a broadcast message to all nodes, except the compromised node, to ignore node 4. The structure of this broadcast message can be seen in Figure 6. As a result of this broadcast, node 5 has to update its routing and send packets to node 3 instead of node 4. Node 4 still continues to send packets, but they are ignored.

3) Denial of Service in Contiki: To simulate a denial of service attack, node 6 sends multiple packets to node 5. As node 5 continues to receive multiple packets in a single round, its battery decreases and eventually dies. Node 4 notices that it is no longer receiving packets from node 5 (i.e. Node 5’s send rate is 0), and after three rounds, it concludes that node 5 is compromised and sends a compromise message informing the gateway. Once the gateway confirms the compromise, it sends a broadcast message to all other nodes to ignore node 5.

V. Results and Discussion

A. Detection of Replay Attacks in MBANs

Replay attacks introduce additional and unanticipated packets into the MBAN. When a node receives a packet twice in one round (when it is really only expecting one), one of which is coming from a node farther away and outside the network, its battery level will be significantly affected. This causes the compromised node to eventually duty cycle faster, meaning that eventually its send rate becomes lower than the send rate of its neighbors. If a replay attack is limited, it may be more difficult to detect. We have determined that a replay attack must occur continuously for at least two rounds (at least 2 packets) for our method to detect it with in the next two rounds. Otherwise, the attack will take longer to detect.

B. Detection of Captured Nodes in MBANs

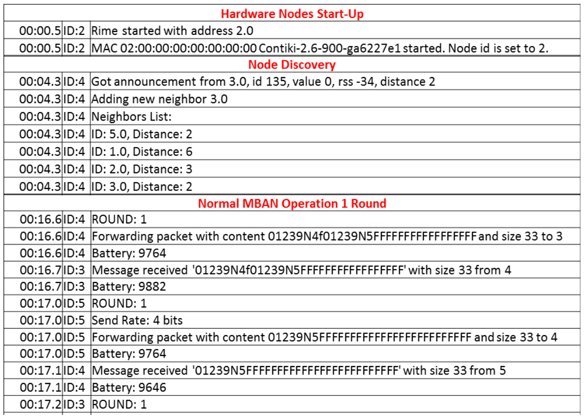
To capture a node, an attacker will need to connect to the target node in some way. Any extra communication outside of legitimate operations (receiving and sending one packet per round) will eventually be detected. During the process of the attacker capturing the node (via remote communications), the node’s battery level drops significantly if the attacking node’s distance is greater than the distance to the legitimate node’s closest neighbor. Depending on the number of packets required to capture the node, this activity could drop the legitimate node’s duty cycle more than one level, and thus making the attack very easy to detect within two rounds. In Figures 3 and 4 very detailed output is given from our actual Contiki implementation. In Figure 3 the hardware-based nature of our implementation is very evident. Our hardware simulated nodes start-up and discover their neighbors just like real nodes would. Also, the packet layout we defined earlier can be seen here operationally. Also the power dissipation due to sending and receiving packets are clear as well. Finally in Figure 4 all of the steps from node compromise, to inference of the the threat to ”Tattle” to the gateway to the gateway notifying all nodes to avoid the compromised node can be seen.

C. Detection of DoS Attacks in MBANs

The nature of a DoS attack is to exhaust the node. Here we only consider a quick DoS, which completely exhausts the node before it can send a packet; otherwise, this detection becomes similar to the replay or node capture attack, because the node has been slightly dissipated of power. We implemented the quick DoS attack by sending one very long packet to the target node in one round thats completely exhausts it. Once the compromised node is completely exhausted, it has a battery level of 0, and is unable to receive or forward messages, and thus has a send rate of 0. Although the compromised node’s neighbor is not receiving any packets, it can detect that the send rate is different than its own since it has no send rate to compare to its own. To detect this kind of attack, at least four rounds are needed for the gateway to confirm the compromise. The neighbor must not receive a packet for three rounds to verify that its neighbor is indeed unresponsive and has not just fallen prone to a routing error for one or two rounds. In the third round, the neighbor node will “Tattle” on the compromised node and inform the gateway. After the gateway receives the node compromise information for the second time in the fourth round, it will inform all the nodes of the unresponsive node.

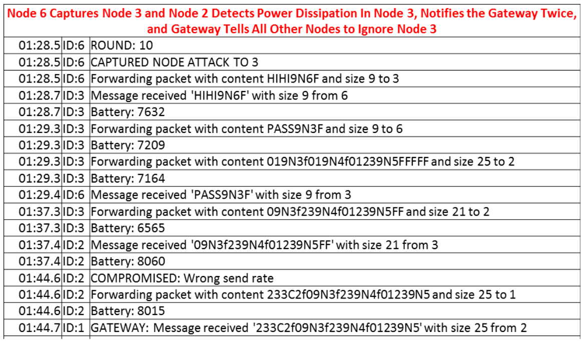
D. Comparing ”Tattle Tale” Security to Other Methods

Overall, we believe that the strength of our work is our implementation in Contiki, which demonstrates the behavior of real hardware. As compared to the above mentioned methods (Table 2), even though they thoroughly simulated the performance of their methods and offered metrics as evidence,



Hardware Nodes Start-Up		
00:00.5	ID:2	Rime started with address 2.0
00:00.5	ID:2	MAC 02:00:00:00:00:00:00:00 Contiki-2.6-900-ga6227e1 started. Node id is set to 2.
Node Discovery		
00:04.3	ID:4	Got announcement from 3.0, id 135, value 0, rss -34, distance 2
00:04.3	ID:4	Adding new neighbor 3.0
00:04.3	ID:4	Neighbors List:
00:04.3	ID:4	ID: 5.0, Distance: 2
00:04.3	ID:4	ID: 1.0, Distance: 6
00:04.3	ID:4	ID: 2.0, Distance: 3
00:04.3	ID:4	ID: 3.0, Distance: 2
Normal MBAN Operation 1 Round		
00:16.6	ID:4	ROUND: 1
00:16.6	ID:4	Forwarding packet with content 01239N4f01239N5FFFFFFFFFFFFFFFFFF and size 33 to 3
00:16.6	ID:4	Battery: 9764
00:16.7	ID:3	Message received '01239N4f01239N5FFFFFFFFFFFFFFFFFF' with size 33 from 4
00:16.7	ID:3	Battery: 9882
00:17.0	ID:5	ROUND: 1
00:17.0	ID:5	Send Rate: 4 bits
00:17.0	ID:5	Forwarding packet with content 01239N5FFFFFFFFFFFFFFFFFFFFFFFFFF and size 33 to 4
00:17.0	ID:5	Battery: 9764
00:17.1	ID:4	Message received '01239N5FFFFFFFFFFFFFFFFFFFFFFFFFF' with size 33 from 5
00:17.1	ID:4	Battery: 9646
00:17.2	ID:3	ROUND: 1

Fig. 3. Node Start-up, Discovery, and Normal Operation Excerpt From Contiki



Node 6 Captures Node 3 and Node 2 Detects Power Dissipation In Node 3, Notifies the Gateway Twice, and Gateway Tells All Other Nodes to Ignore Node 3		
01:28.5	ID:6	ROUND: 10
01:28.5	ID:6	CAPTURED NODE ATTACK TO 3
01:28.5	ID:6	Forwarding packet with content HIHI9N6F and size 9 to 3
01:28.7	ID:3	Message received 'HIHI9N6F' with size 9 from 6
01:28.7	ID:3	Battery: 7632
01:29.3	ID:3	Forwarding packet with content PASS9N3F and size 9 to 6
01:29.3	ID:3	Battery: 7209
01:29.3	ID:3	Forwarding packet with content 019N3f019N4f01239N5FFFFF and size 25 to 2
01:29.3	ID:3	Battery: 7164
01:29.4	ID:6	Message received 'PASS9N3F' with size 9 from 3
01:37.3	ID:3	Forwarding packet with content 09N3f239N4f01239N5FF and size 21 to 2
01:37.3	ID:3	Battery: 6565
01:37.4	ID:2	Message received '09N3f239N4f01239N5FF' with size 21 from 3
01:37.4	ID:2	Battery: 8060
01:44.6	ID:2	COMPROMISED: Wrong send rate
01:44.6	ID:2	Forwarding packet with content 233C2f09N3f239N4f01239N5 and size 25 to 1
01:44.6	ID:2	Battery: 8015
01:44.7	ID:1	GATEWAY: Message received '233C2f09N3f239N4f01239N5' with size 25 from 2

Fig. 4. Node Compromise, Inference, Tattle, and Broadcast Excerpt From Contiki

there is no substitute for a hardware implementation. Our implementation is closer to working hardware than any of the mentioned works. We have demonstrated in detailed the normal operation and detection capability of our model.

VI. Limitations

Our method is not without limitations. There are likely classes of attacks that are active, but do not significantly impact the battery power level of the MBAN. These attacks would have to take place from attackers that are closer to the target node than their legitimate neighbors. We denote these at Micro Attacks because of their necessarily closeness in proximity to the target. These types of attacks are feasible, but are likely improbable and thus out of scope. We focus on very practical MBAN threats in this paper. Also, passive attacks, such as eavesdropping, are not detectable by this method, because these types of attacks do not cause power dissipation in a node. However, this type of attack could be thwarted by using our method in conjunction with a lightweight obfuscation algorithm applied to the packets to confuse eavesdroppers. In addition, currently our security mechanism does not handle multiple attacks at once.

VII. Summary and Future Work

We simulated the routing and detection mechanisms of an MBAN using the hardware operating system Contiki [12] [13] [14]. The simulation demonstrates that common MBAN attacks dissipate MBAN battery power levels, which can be detected by a disturbance in the send rates of node packets. These attacks include (but are not limited to): captured node, replay, and denial of service attacks. Future work will experiment with detection in a randomized environment with nodes at different distances apart and without sequential placement. Energy-aware routing can be added to ensure battery levels still remain as expected. Other lightweight security measures, like checking the ID of incoming packets and their frequency per round, can also be examined to improve the efficiency of the “Tattle Tale Security” framework.

References

[1] Sharma, Sukhwinder, et al. Issues and Challenges in Wireless Sensor Networks. 2013 International Conference on Machine Intelligence and Research Advancement, 2013, pp. 5861., doi:10.1109/icmira.2013.18.

[2] Chandramouli, J. M., et al. Using Network Traffic to Infer Compromised Neighbors in Wireless Sensor Nodes. In IEEE Annual Consumer Communications and Networking Conference (CCNC), 2017, pp. 10221023., doi:10.1109/ccnc.2017.7983279.

[3] Chris Karlof, David Wagner, Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures, AdHoc Networks (elsevier), Page: 299-302, 2003.

[4] Ghamari, Mohammad, et al. ”A survey on wireless body area networks for ehealthcare systems in residential environments.” Sensors 16.6 (2016): 831.

[5] Ma, Jianqing, et al. SAID: A Self-Adaptive Intrusion Detection System in Wireless Sensor Networks. Information Security Applications Lecture Notes in Computer Science, pp. 6073.

[6] Huh, Eui-Nam, and Tran Hong. Lightweight Intrusion Detection for Wireless Sensor Networks. Intrusion Detection Systems, 2011, doi:10.5772/14849.

[7] Sampangi, Raghav V. A Security Suite for Wireless Body Area Networks. International Journal of Network Security and Its Applications, vol. 4, no. 1, 2012, pp. 97116.

[8] zderya H.Y., Erdl H., Kaykolu T., Ylmaz A.., Kaya . (2017) Wireless Body Area Network Studies for Telemedicine Applications Using IEEE 802.15.6 Standard. In: Badnjevic A. (eds) CMBEBIH 2017. IFMBE Proceedings, vol 62. Springer, Singapore

[9] Jones, VM, Mei, H, Broens, THF, Widya, IA and Peuscher, J 2007, Context Aware Body Area Networks for Telemedicine. in 8th Pacific Rim Conference on Multimedia, Lecture Notes in Computer Science 4810, no. 67310A, vol. 4810, Springer Verlag, pp. 590-599.

[10] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy efficient communication protocol for wireless microsensor networks, In the proceedings of the 33rd Hawaii International Conference on system sciences-HICSS2000.

[11] G. Thamilarasu and A. Odesile, ”Securing Wireless Body Area Networks:Challenges, Review and Recommendations,” IEEE International Conference on Computational Intelligence and Computing Research, 2016.

[12] Z. He and X Bai, ”A Wearable Wireless Body Area Network for Human Activity Recognition,” IEEE International Conference on Ubiquitous and Future Networks, 2014.

[13] I. Ciabattoni, A. Freddi, S. Longhi, A. Monteriu, I. Pepa, and M. Prist, ”An Open and Modular Hardware Node for Wireless Sensor and Body Area Networks,” Journal of Sensors, Volume 2016, Article ID 2978073, Hindawi Publishing, 2016.

[14] R. Jacobsen, F. Hansen, J. Madsen, H. Karstoft, P. Mikkelsen, T. Skogberg, E. Rasmussen, C. Andersen, M. Alroe, and T. Toftegaard, ”A Modular Platform for Wireless Body Area Network Research and Real-life Experiments,” Internal Journal of Advances in Networks and Services, Vol 34, 2011.

[15] S. Alam and D. De, ”Analysis of Security Threats in Wireless Sensor Network,” International Journal of Wireless Mobile Networks, Vol 6, 2014.