OPSENET: A Security-Enabled Routing Scheme for a System of Optical Sensor Networks

Unoma Ndili Okorafor and Deepa Kundur
Department of Electrical & Computer Engineering
Texas A & M University, College Station, Texas 77843-3124
Email: unondili, deepa@ece.tamu.edu

***Abstract*— In this paper we introduce OPSENET, a novel and efficient protocol that facilitates secure routing in directional optical sensor networks. We show that even though the uni-directionality of links in an optical sensor network (OSN) complicates the design of efficient routing, link directionality actually helps security in our network setup. In particular, we leverage the naturally-occurring clustering that results from passive (bi-directional) communication of select cluster head nodes with the base station, to improve overall network performance. This paper presents two main contributions: (1) We introduce OPSENET, a novel secure cluster-based routing algorithm for base station circuit discovery in OSNs. In order to support the efficient utilization of a nodes' resources, we employ symmetric cryptography in the design of OPSENET, using efficient one-way hash functions and pre-deployed keying. OPSENET achieves base station broadcast authentication, per-hop authentication, and cluster group secrecy, without requiring any time synchronization. (2) We analyze the relevance of traditional routing attacks on OSNs, and show that OPSENET is robust against uncoordinated (non-smart) insider routing attacks, amongst other compromises. An important performance metric of OPSENET is its low byte overhead, and graceful degradation with the number of compromised nodes in the network. To the best of our knowledge, this is the first paper to consider secure routing in an OSN network scenario.**

I. Introduction

Optical sensor networks (OSNs) are a subclass of sensor networks consisting of nodes that communicate via free space optical (FSO) links [1]. Recently, OSNs have attracted attention [2]–[5] owing to the advantages that optical communication offers over conventional RF networks, including cost savings due to lower power consumption, smaller sized nodes, inexpensive components including light emitting diodes (LEDs), photo-detectors and Positive Intrinsic Negative (PIN) devices, and the use of passive transmitters such as corner cube retro-reflectors (CCR) [5]. Owing to its dramatic spatial re-use of bandwidth, FSO avoids interference with existing communication infrastructure and does not require expensive and delayed government licensing. This enables easy and quick network deployment and reconfigurability. Additionally, OSNs realize ultra-high bandwidths which can benefit real-time multimedia and visual sensor network applications [28]. Well designed FSO systems are also eye and skin safe.

The main limitation of FSO includes the reduced bit rates encountered in some adverse atmospheric conditions as fog; physical obstructions that may cause temporary interruptions; and the direct line-of-sight requirement between a sender and a receiver. Discussion on the first two limitations is beyond the scope of this paper, since we are primarily concerned with the networking implications on OSNs. However, it is worth mentioning that a network design that shortens FSO link distances and adds network redundancies helps to counter the adverse effect of atmospheric conditions. In addition, we may consider indoor FSO deployment scenarios, in which atmospheric conditions and temporary physical obstructions have very little effect. An example of such an application is an underground tunnel with visual/multimedia control and monitoring systems, that require the high bandwidth and flexibility of reconfigurability (e.g. using mobile sensor nodes) offered by FSO links.

In this work, we will focus on the line-of-sight limitation, which has direct impacts on the network layers of an OSN, as it implies that all network links are uni-directional. That is, even though a node can talk to a neighboring node, he may not directly hear this neighbor. This directed communication results in a network that has been modeled as a directed random scaled sector graph [2]. This model has also been studied for directional RF networks [26], [27].

Routing under the uni-directional OSN paradigm is challenging to design, and a secure one even more so. It is well known though, that incorporating security mechanisms after the design of protocols is often non-trivial and superficial at best [6]. It is therefore beneficial to consider security objectives at the initial design of our routing scheme. Because of the importance of routing in networks, an attack in the network layer can completely cripple the OSN, in spite of best efforts aimed at securing other OSI layers of the network. In addition, the vulnerability of sensor nodes to physical capture and tampering, combined with the collaborative nature of communication in a sensor network, makes network layer protection mechanisms even more crucial [7].

As with any wireless medium, unsecured FSO is susceptible to outsider routing attacks as data replay, identity theft, or injection of unauthorized bits into the network. Worse, an enemy that is able to compromise an authentic network node, may easily launch more serious insider attacks, by extracting keying and security information from the compromised node, and then acting as an authentic network participant [6]. Such insider routing attacks include selectively forwarding or falsifying routing control packets, creating routing loops, sinkhole, wormhole, acknowledgement spoofing, sybil [9], denial of

1-4244-0425-8/06/$20.00 ©2006 IEEE

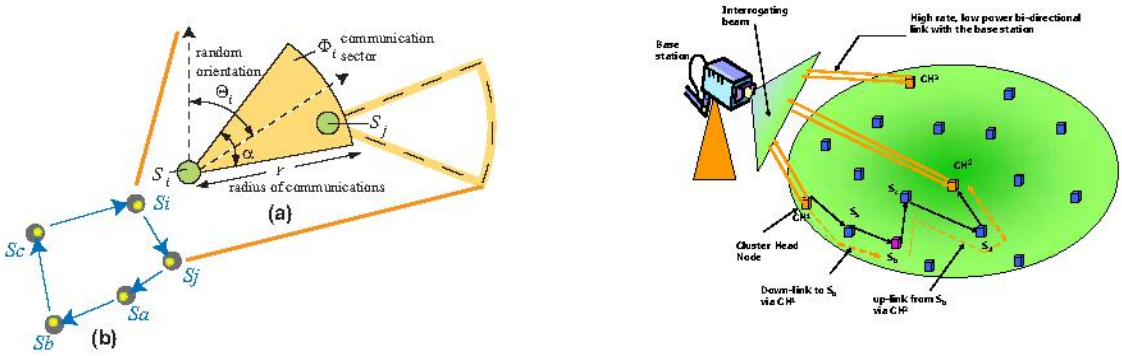

Fig. 1. (a) Node S_j can only hear node S_i if it falls into S_i's communication section. (b) However S_j talks to S_i via the back channel $S_j \rightarrow S_a \rightarrow S_b \rightarrow S_c \rightarrow S_i$. (c) A hierarchical network structure for OSN with CH nodes that are (randomly) oriented with a direct line-of-sight path to the base station. Cluster heads have a low power, high rate, bi-directional link with the base station. Other nodes establish up link and downlink paths with the base station via the cluster heads.

message [10], and HELLO flood attacks. These attacks are intended to create a denial of service (DoS) [11] in the network.

In this paper, we focus on the problem of secure routing in an OSN. We show that even though the uni-directionality of links in an OSN complicates the design of an efficient routing protocol, link directionality helps security in our network setup. This is because vulnerabilities in routing schemes that use reverse paths from parent nodes back to the base station (BS) become irrelevant in a directed OSN. In particular, for OSNs, a node does not know a priori neighbors that hear him, since reverse-path routing and link layer acknowledgement are non-trivial in uni-directional networks. Therefore, flooding-based attacks are generally more difficult in oriented and directional networks. We leverage the naturally-occurring clustering that results from passive (bi-directional) communication of select cluster head (CH) nodes with the BS, to improve overall network and security performance.

This paper presents two main contributions: First, we introduce OPSENET, a novel secure cluster-based routing algorithm for *base station circuit* (BS-circuit) discovery in OSNs. The *BS-circuit* consists of a downlink and up-link path between each node and the BS through a CH. In order to support the efficient utilization of a nodes' resources, we employ symmetric cryptography in the design of OPSENET, using efficient one-way hash functions and pre-deployed keying. OPSENET achieves base station broadcast authentication, per-hop authentication, and cluster group secrecy, without requiring time synchronization in the network. Second, we analyze the relevance of traditional routing attacks on OSNs, and quantify the relevance of such attacks on OPSENET. We show that OPSENET is robust against uncoordinated (non-smart) [7], [8] insider routing attacks. Our analysis show that the byte overhead of OPSENET is comparable with non-secure routing in an OSN, and the network gracefully degrades with the number of compromised nodes in the network. To the best of our knowledge, this is the first paper to consider secure routing in an OSN network setup.

In Section 2, we present our preliminaries, assumptions and network setup. Section 3 introduces the OPSENET protocols, while Section 4 analyzes the security and network performance of OPSENET. In Section 5 we present our attack analysis. Section 6 discusses our simulations and results, Section 7 briefly presents some related work. Finally, we present concluding remarks in Section 8.

II. Preliminaries

A. Network Setup

Flat Topology: Consider an OSN in which n nodes labeled as $S_i : i = 1, 2, \cdots n$, are densely and randomly deployed in a given area. All nodes are equipped with an optical transreceiver consisting of photo-detectors and a semi-conductor laser (e.g. eye-safe 1550 nm wavelength laser), which has a given communication range r. Each node S_i has a random position (x_i, y_i) and random direction Θ_i, and can orient its transmitting laser within a contiguous angular scanning region $\frac{-\alpha}{2} + \Theta_i \leq \Phi_i \leq \frac{+\alpha}{2} + \Theta_i$. Following the model in [2], [3] and as depicted in Figure 1(a), each node S_i can send data over a randomly oriented sector Φ_i of α degrees, for a fixed angle $0 < \alpha < 2\pi$, with α attaining values up to $\frac{2\pi}{9}$ [3]. The receiving photo-detector is omni-directional and thus receives data from any direction. As shown in Figure 1(a), for node S_j to hear S_i, we must have that:

$$d(S_i, S_j) \leq r \quad \text{and} \quad (x_j, y_j) \in \Phi_i$$

where $d(S_i, S_j)$ is the Euclidean distance between S_i and S_j. In this setup, S_i may directly talk to S_j (denoted as $S_i \rightarrow S_j$); however, S_j can only talk to S_i via a multi-hop back-channel or reverse route, with other nodes in the network acting as routers along the path. In the example of Figure 1(b) this reverse route is: $S_j \rightarrow S_a \rightarrow S_b \rightarrow S_c \rightarrow S_i$. This OSN (or directed RF sensor system) has been modeled by Diaz et. al. [2] as a *random scaled sector graph*. The following definition is derived from [2].

Definition II.I *For any natural n and fixed angle α, let $\boldsymbol{S} = (S_i)_{1 \leq i \leq n}$ be a sequence of* independently and uniformly distributed *(i.u.d.) random coordinate of points in $[0,1]^2$, a sequence $\Theta = (\Theta_i)_{1 \leq i \leq n}$ of i.u.d. angles, and $(r_i)_{1 \leq i \leq n}$ a sequence of numbers (x_i, y_i) in $[0,1]$. Let $\chi_n = \{X_1, \cdots X_n\}$ and $\Theta_n = \{\Theta_1, \cdots \Theta_n\}$. We call the graph $G_\alpha(\chi_n, \Theta_n, r_n)$ the* random scaled sector graph *with n nodes.*

We assume throughout this paper, that r is the same for each node, and verifies network connectivity constraints [2], [18] as $r \geq \sqrt{\frac{c.\log n}{n}}$ where c is a constant dependent on α. We define S_i's forward neighborhood $FNeb(S_i)$ as the set of nodes that S_i can talk to, i.e., $FNeb(S_i) = \{S_k\}, \forall k$ such that $(x_k, y_k) \in \Phi_i$ and $d(S_i, S_k) \leq r$. Nodes in $FNeb(S_i)$ are called S_i's *successors*. Similarly, S_i's *predecessors* are nodes in its backward neighborhood defined as $BNeb(S_i) := \{S_h\}, \forall h$ such that $(x_i, y_i) \in \Phi_h$ and $d(S_i, S_h) \leq r$. We define *outlier nodes* as nodes who have either $FNeb$ or $BNeb$ (or both) as empty sets. The communication in the flat topology is by scanning of an active laser beam.

Hierarchical Topology: Passive transmitters such as CCR [5] are especially attractive for OSNs because of their small size, ease of operation and extremely low power consumption (less than 1 nJ/bit). A CCR is a simple optical device that reflects incident light back to it source, and when used to modulate an interrogating beam from the BS yields huge energy savings compared to an active laser. We assume that all nodes are equipped with a CCR, and after random deployment, a fraction of nodes (practical tests have shown about 10% [3]) will be oriented so as to have a direct line-of-sight, and hence a passive bi-directional communication with the BS (See figure 1(c). We designate any such node S_i as a cluster head (CH), denoted as S_i^*. CHs forward data to the BS without adversely depleting their energy resources. This is a huge advantage since we can leverage cluster-based routing without the energy considerations associated with CH nodes. This network setup leads naturally to a hierarchical structure in which nodes aim to discover the 'closest' CH through which they can route their data to the BS (uplink), or receive data or broadcasts from the BS (downlink). It is well known that hierarchy improves overall network performance [6], [12], [13], and the implicit clustering ensures that nodes closer to the BS do not bear excessive routing loads, leading to their early demise. In this paper, we denote P_{CH} as the probability that a node is a CH.

Network Reconfigurability: The OSN setup makes it conceivably trivial to mitigate attacks based on traffic pattern analysis by dynamically and periodically changing the set of CHs, (and hence network topology). This network reconfiguration can be achieved with the BS simply moving to a different location and re-initiating the OPSENET protocol. Unlike other cluster-based routing schemes in the literature [14] in which CHs elect themselves based on some criteria, the CHs in our OSN scenario are *discovered* based on their relative orientation to the BS. This 'involuntary' election of CHs based on orientation helps thwarts attacker models that target and disable CH nodes since a malicious node cannot arbitrarily elect itself to a CH position.

The OSN network layer has been studied to a limited extent in [2]. The authors have proposed a localization and (orientation) synchronization scheme, in addition to two main routing protocols for uplink and downlink path discovery for each node: `simple-bro` algorithm for broadcasting (downlink), and the `simple-gather` algorithm for gathering (uplink). Our work differs substantially from [2] in three important respects:

1) We have combined gathering (uplink) and broadcasting (downlink) path discovery into one protocol, thereby saving network energy and byte overhead, and increasing overall network efficiency.
2) We have incorporated security considerations in the design of our OSN routing protocol. As stated earlier, it is vital to consider security implications during the initial design of routing schemes. In addition, we discuss route maintenance using acknowledgement packets.
3) We make no attempt to fix a node's scanning laser in a given orientation during routing setup. Instead, we assume that a node may broadcast to any node in its $FNeb$ by scanning over its communication sector, or may send data to any one of its successors by appropriately orienting its laser in that neighbor's direction. Even though this may (arguably) expend more energy for laser scanning, it results in a more flexible network structure for efficient routing. Comparing asymptotic energy usage for both scenarios is a focus of our future research efforts.

We show byte overhead comparison between OPSENET, non-secure OPSENET, and the `simple-bro/simple-gather` algorithm in Section 6.

Assumptions: Our assumptions include: all nodes are homogeneous, stationary, incapable of asymmetric cryptography and equipped with optical communication hardware including CCRs. The BS is not resource-constrained, and forms part of the trusted infrastructure. A node may be attacked even before route establishment, and if a node is compromised, its keying and security primitives become available to the attacker, making it possible for an attacker to control the node in an arbitrary way [15]. Finally, an attacker is constrained to similar hardware limitations as the network nodes.

Figure 2 shows three sample OSN node graphs (flat topology) with (a) $n = 70$, (b) $n = 200$ and (c) $n = 500$, respectively. $\alpha = 40^0$ and r is computed from n to ensure network connectivity with high probability. Also, as n increases, r decreases to avoid the problem of "over-connectedness." [18]. For each sample network, the number of outlier nodes is $9, 8$, and 5 respectively. We see that network connectivity increases with network size. In our simulations, outlier nodes

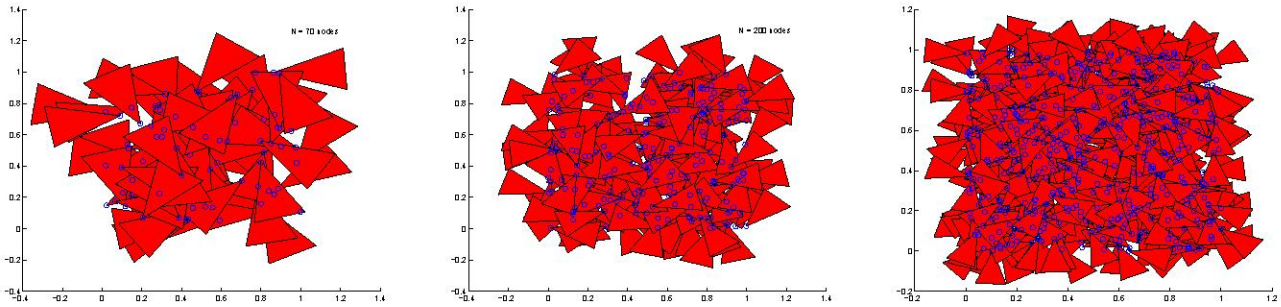

Fig. 2. Sample network graph for a random sector graph with (a) $N = 70$ nodes. (b) $N = 200$ nodes, and (c) $N = 500$ nodes. Radius of communication r is a function of n and α. The blue circles represent nodes while the red regions correspond to communication regions for the associated node at the emanating vertex.

are ignored. In practical deployment, since this is a small fraction for large networks, such nodes may be assumed "lost" without affecting network functionality.

III. OPSENET

A. Problem Formulation and Design Goals

Different from ad hoc networks in which communication is peer-to-peer, most of the communication in a sensor network is uplink: from the BS to nodes, or downlink: from the nodes to the BS. Therefore routing in a sensor network should be *data-centric* (query or event-based), and consider prevalent patterns of either node to BS, BS to nodes, or local neighborhood communication [6]. In addition, due to their sensing objective, random deployment, and hardware limitations, sensor network routing should also be: (1) energy efficient, (2) scalable, (3) capable of in-network processing (4) hierarchical, and (5) security aware.

Security objectives of a routing protocol must ensure that fabricated routing signals cannot be injected into the network, routing messages cannot be maliciously altered, routing loops cannot be formed, and routes cannot be maliciously redirected [16]. We believe that a hierarchical topology that provides per hop, broadcast and message authentication with link layer acknowledgements is well on its way to achieving many of these objectives. We have designed OPSENET with these characteristics and initial objectives in mind.

B. Introduction to OPSENET

OPSENET is a security aware *BS-circuit* discovery algorithm that provides per hop authentication, as well as assures nodes of the origin and integrity of routing signals. *We define a* BS-circuit *as a sequence of unidirectional links between network nodes, with a path leading away from, and back to the BS, thus forming a directed loop*. From a given node's point of view, the BS-circuit reveals its uplink and downlink paths to and from the BS respectively. Nodes on the same BS-circuit are called a cluster, and each node will usually be in more than one cluster. We assume that the network is strongly connected [2], [18] and every non-outlier node in the network is contained in at least one cluster. OPSENET uses one-way key chains and individual keys to defend against unauthorized participation and spoofed, altered or replayed route signals. We use the following notation to describe our security protocols: $A|B$ denotes concatenation of message A with message B. $E_K(M)$ and $D_K(M)$ respectively denote the encryption and decryption of message M with key K, while $MAC_K(M)$ is the message authentication code (MAC) of M with key K.

C. Initialization and Key Setup

As similarly proposed in [19], before network deployment, the BS pre-computes and stores W length-q one-way *key chains* by successively applying a known one-way function F to randomly generated keys K_q^w, for $1 \leq w \leq W$, so that $F(K_j^w) = F(K_{j+1}^w)$ for $j = 1 \cdots q$. Due to the nature of forward functions, future keys cannot be computed from previous keys. However, it is trivial to verify that a future key once revealed was derived from a previous key, by simply applying F one or more times to the past keys. Keys in a chain are revealed by the BS in the reverse order from which they were generated. The last key K_1^w in each key chain is known as the *commitment* to that chain, and the set of all W commitments form the *key pool* stored at the BS.

The key chains are used for network and cluster broadcast authentication as well as secure in-network processing. Each node S_i is pre-deployed with an *individual key* K_{S_i} shared with the BS, a counter C_i initialized to a random value, also known to the BS, and a fixed number ξ of randomly selected commitments from the key pool, that form the node's key ring $KR(S_i)$. All key rings contain the compulsory commitment K_1^1 for network-wide broadcast. The other commitments in $KR(S_i)$ are statistically selected so that with high probability (w.h.p.), nodes on a BS-circuit share at least one commitment. This shared BS-circuit key chain is used for establishing cluster group keys to support in-network processing (if needed).

The idea of key chains for broadcast authentication in sensor networks was introduced as μ-TESLA by Perrig et al [19]. However, in contrast to μ-TESLA, our setup does not require any time synchronization in the network, because routing signals are signed by individual nodes and terminated by the BS, and the directionality of links does not allow for reverse path spoofing attacks. This advantage is significant, since time synchronization is often difficult to achieve in practice, and is the main drawback for μ-TESLA [19].

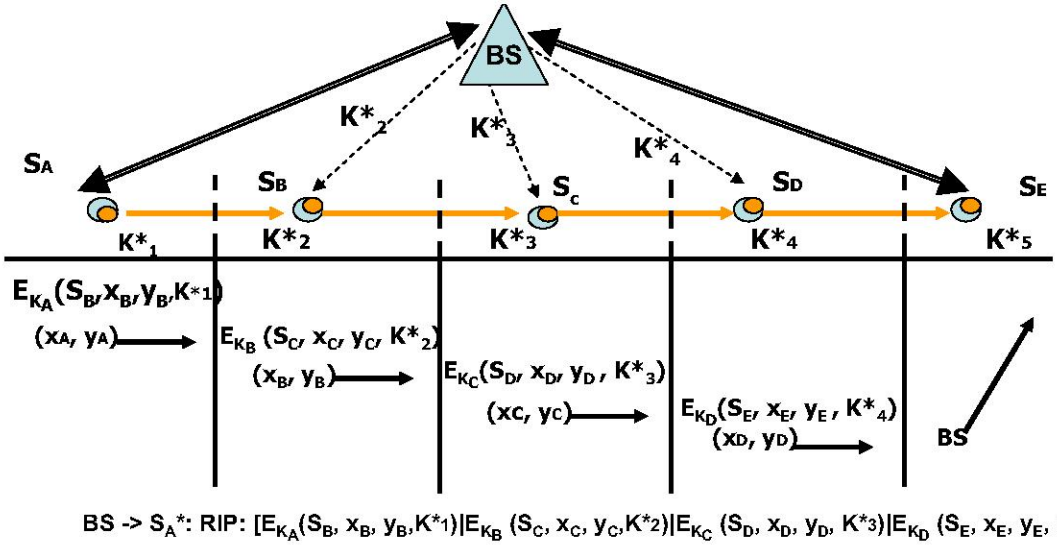

Fig. 3. A simple circuit $S_A^* \rightarrow S_B \rightarrow S_C \rightarrow S_D \rightarrow S_E^*$ to illustrate how nodes securely extract only the needed local successors information (ID and location), as well as future keys of a common key chain used for in-network cluster group processing.

Another contribution of our work is the concept of key rings composed of commitments to ξ randomly selected key-chains [17], useful for establishing cluster keys for secure cluster group activities. OPSENET achieves security using individual, network wide and group keys. As suggested in [12], this key differentiation is necessary and useful in supporting various applications. Pair-wise keys have not been considered here because they are not required for the communication pattern we have assumed.

D. OPSENET Base Station Circuit Discovery Algorithm

After key pre-distribution and deployment, the BS floods the network, and waits for a response from CHs, in order to establish authenticated communication with them (using for example a challenge and response protocol [19]). From this point on a CH is assumed to be trusted. BS-circuit discovery is initiated when the BS broadcasts circuit discovery packets (CDP) via the CHs. The format of a new CDP packet is $[HT = 0|K_2^1|E_{K_1^1}(nonce)]$, where HT is the *hops traversed* field and the nonce is a randomly generated bitstream used to achieve data freshness. Note that all initial CDP packets are deployed with the same nonce sequence from the BS, therefore the BS does not have to keep track of individual CDP packets sent along different routes. A different nonce is used for each BS update or broadcast. Upon receiving a CDP packet from the BS, a CH S_i^* verifies that $F(K_2^1) = K_1^1$. If so, he decrypts the nonce using the pre-stored K_1^1, XOR's the nonce with his counter value, increments the HT field by one, and signs (re-encrypts) the (incremented) nonce using his individual key. He then appends a MAC of his information (identity S_i^* and position x_i, y_i) and the modified nonce to the CDP packet, before broadcasting to all his successors. Note that we have chosen to use the XOR function since it does not expand byte overhead of the algorithm. After the first two hops through S_i^* and S_j, the CDP packet looks thus: $[HT = 1|K_2^1|E_{K_1^1}(nonce \oplus C_i)||MAC_{K_{S_i}}(S_i^*|(x_i, y_i))]$ and $[HT = 2|K_2^1|E_{K_1^1}(nonce \oplus C_i)||MAC_{K_{S_i}}(S_i^*|(x_i, y_i))||MAC_{K_{S_j}}(S_j|(x_j, y_j))]$, respectively, and so on. Individually signed nonces provide per hop authentication on the routes, while the MAC's ensure that malicious nodes cannot tamper with a previous nodes personal information entry, (i.e., identity of position).

To avoid routing loops, each non-CH node in receipt of a CDP first examines the sequence of appended MAC IDs to ensure that it has not received this particular CDP from the same route of predecessors. If it has not processed the CDP previously, the node processes the CDP in the same manner as a CH, decrypting and signing the nonce appropriately before broadcasting it. A node simply drops the CDP if it has 'seen' this packet, if it does not verify, or if its $HT > \delta$, where δ is a pre-defined constant used to avoid excessively long circuits. When a CDP with $1 < HT \leq \delta$ reaches a CH, its route discovery task is terminated by the CH who closes the BS-circuit and forwards the packet back to the BS. Our simulations show that all network-connected nodes find multiple BS-circuits when δ is reasonably bounded [2], [18].

Base Station Processing: From each CDP, the BS extracts the sequence of nodes on a BS-circuit, and can authenticate all the nodes on this route using their counters and individually signed nonce. Any CDP that does not verify is discarded and route maintenance and intrusion detection schemes initiated for this route. The BS uses the location information in the CDP to construct an approximate network topology, which as we show in Section 5, is useful in defending against Sybil attacks [9].

Once the BS has validated a BS-circuit, he constructs a route information packet (RIP) consisting of the uplink next hop information for each node on the BS-circuit.

TABLE I

IN-NETWORK PROCESSING FOR SAMPLE NETWORK OF FIGURE 3 ILLUSTRATING SUCCESSIVE DECRYPTION AND ENCRYPTION OF AGGREGATED DATA $\mathcal{D}_*$ FOR NODE S_*.

$\mathcal{D}_\mathcal{A}$ at S_A, K_1^*	$\mathcal{D}_\mathcal{B}$ at S_B, K_2^*	$\mathcal{D}_\mathcal{C}$ at S_C, K_3^*	$\mathcal{D}_\mathcal{D}$ at S_D, K_4^*	$\mathcal{D}_\mathcal{E}$ at S_E, K_5^*
$E_{K_1^*}(\mathcal{D}_\mathcal{A})$	$K_1^* = F(K_2^*)$	$K_2^* = F(K_3^*)$	$K_3^* = F(K_4^*)$	$K_4^* = F(K_5^*)$
	$D_{K_1^*}(E_{K_1^*}(\mathcal{D}_\mathcal{A}))$	$D_{K_2^*}(E_{K_2^*}(\bigwedge(\mathcal{D}_\mathcal{A}, \mathcal{D}_\mathcal{B})))$	$D_{K_3^*}(E_{K_3^*}(\bigwedge(\mathcal{D}_\mathcal{A}\mathcal{D}_\mathcal{B}\mathcal{D}_\mathcal{C})))$	$D_{K_4^*}(E_{K_4^*}(\bigwedge(\mathcal{D}_\mathcal{A}, \mathcal{D}_\mathcal{B}, \mathcal{D}_\mathcal{C}, \mathcal{D}_\mathcal{D})))$
	$E_{K_2^*}(\bigwedge(\mathcal{D}_\mathcal{A}, \mathcal{D}_\mathcal{B}))$	$E_{K_3^*}(\bigwedge(\mathcal{D}_\mathcal{A}, \mathcal{D}_\mathcal{B}, \mathcal{D}_\mathcal{C}))$	$E_{K_4^*}(\bigwedge(\mathcal{D}_\mathcal{A}, \mathcal{D}_\mathcal{B}, \mathcal{D}_\mathcal{C}, \mathcal{D}_\mathcal{D}))$	$E_{K_5^*}(\bigwedge(\mathcal{D}_\mathcal{A}, \mathcal{D}_\mathcal{B}, \mathcal{D}_\mathcal{C}, \mathcal{D}_\mathcal{D}, \mathcal{D}_\mathcal{E}))$

The BS authenticates and successively encrypts the RIP with each nodes' individual key, so that (unnecessary) future routing information is not revealed to the nodes. This prevents a malicious node from altering routing information of subsequent nodes in the circuit. Each node only extracts his uplink successor's ID and location from the RIP. Consider for example, the BS-circuit $S_A^* \rightarrow S_B \rightarrow S_C \rightarrow S_D \rightarrow S_E^*$ of Figure 3, in which the CDP has successfully returned to the base station via the CH S_{E*}. The BS then constructs and sends the following RIP to S_A^*: $[E_{K_A}(S_B, x_B, y_B)|E_{K_B}(S_C, (x_C, y_C))|E_{K_C}(S_D, x_D, y_D)|$ $E_{K_D}(S_E, x_E, y_E)]$.

Each node progressively extracts, decrypts and caches his own part of the routing information from the RIP, and forwards the packet to the next successor on the BS-circuit using the location information from the RIP. Note that S_E^* does not need any successor information since he simply returns the RIP to the BS, which then knows that routing update on this BS-circuit is complete. The RIP proves authentic since only the BS could have generated correctly encrypted data with each node's individual key. Position information in the RIP is used by nodes to correctly orient their laser in a fixed direction to communicate with a given successor. Therefore it is to a malicious nodes' disadvantage to misrepresent his location data, since he will no longer receive packets from a predecessor, unless he intentionally aims to achieve a blackhole DoS attack. Our route maintenance scheme helps to detect and recover from such blackholes.

Cluster Key Chain for In-Network Processing: For applications that require in-network processing, our scheme supports cluster-level data aggregation. The BS knowing the key ring of all nodes, can select and reveal a key ring common to all nodes in a BS-circuit. In addition, the BS reveals future keys to subsequent nodes on the circuit, in order to enable successive decryption and encryption of aggregated data along a circuit. A simple illustration of this scheme is summarized in Table 1 and Figure 3. Consider nodes $S_i, i \in (A, B \cdots E)$ on the BS-circuit shown in Figure 3. Each node S_i has sensor reading $\mathcal{D}_i$ destined for the BS, and we aim to achieve in-network data aggregation, such as averaging (denoted $\bigwedge(.,.)$) along this cluster. We assume all nodes on this circuit possess a common key chain commitment denoted K_1^* [17]. The BS uses the segments of the RIP to reveal individual members of the key chain K_2^* to S_B, K_3^* to S_C, and so on, where $K_{v+1}^* = F(K_v^*)$. The RIP would then be S_A^*: $[E_{K_A}(S_B, x_B, y_B, K_1^*)|E_{K_B}(S_C, x_C, y_C, K_2^*)|$ $E_{K_C}(S_D, x_D, y_D, K_3^*)|$ $E_{K_D}(S_E, x_E, y_E, K_4^*)]$.

S_A forwards his data encrypted with his cluster key to S_B as: $E_{K_1^*}(\mathcal{D}_A)$. S_B can easily derive K_1^* from his knowledge of K_2^* as $K_1^* = F(K_2^*)$, and can therefore decrypt S_A's reading and aggregate with his own reading before encrypting with his cluster key K_2^* as $E_{K_2^*}(\bigwedge(\mathcal{D}_A, \mathcal{D}_B)$, where $\bigwedge$ denotes the aggregation function. This aggregation model (due to directional communication) is structured such that a *common* cluster key is not needed. Instead, aggregation is possible in succession along a routing path. This is in contrast to other aggregation models using omnidirectional communication, in which many different aggregation pairs are possible.

E. Route Maintenance

Route Maintenance aims to discover malfunctioning, dead or subverted nodes along the circuit. In OPSENET, route maintenance is achieved by leveraging the naturally occurring path diversity in the network; each node is with high probability, part of more than one cluster and hence can exploit multiple paths to the BS in order to alert the network of possible malicious behavior. As shown in [18], each node with $O(\log n)$ predecessors and successor should discover the same order of up-links and down-links within OPSENET. If we assume that the BS sends (low-rate) acknowledgement packets (ACK) for received data packets, route maintenance is initiated if a node does not receive ACKs for data packets sent within a given time frame.

In this process, a node sends route maintenance request (RMReq) to the BS via a different (randomly selected) successors uplink path. Once a BS authenticates a RMReq, he initiates route maintenance by querying each successor node on the up-link circuit to be tested. He does this by requesting multi-path 'multi-cast' returns of his route maintenance query (RMQuery) packet, similar to flooding. Each node appends his node statistics and time stamp on the RMQuery packet so the RMQuery accumulates circuit information. Validated circuits whose RMQuery packets return to the BS are updated as functional. Since flooding is robust, this process will easily discover 'bad' nodes in the BS-circuit. Even though it is conceivable that stealthy malicious nodes may choose to behave properly when processing RMQueries, time stamping RMQuery packets may assist the BS in knowing which nodes are running malicious code. Since a malicious node will have to load authentic code in memory to correctly process data,

time delayed in processing the RMQuery may expose the node as fraudulent. This time stamping idea has been explored in [20] for remote entity verification.

IV. Security and Network Analysis

Unauthorized participation: OPSENET assures that unauthorized alien nodes cannot participate in route establishment using per hop authentication. Only authentic network participants are assumed to know K_1^1 used to decrypt the nonce, and secure individual keys K_{S_i} used to sign the nonce are known only to individual authentic network nodes. It is conceivable that an attacker can compromise one node, steal and distribute its individual key to alien nodes to enable them participate in routing. We discuss this attack further and its countermeasures in Section V, under identity replication attack.

Spoofed routing Signaling: Since only the BS knows future keys used to authenticate routing signals, no other entity can spoof, fabricate or initiate route signalling. Time synchronization is not required in the revelation of future keys since routing signals either expire or are terminated by the BS.

Alteration of Routing Messages: There are three fields a malicious node may alter in the routing signal in order to confuse the network. (1) A node may hope to decrement the HT field in order to appear as having a shorter route to the BS and possibly achieve a sinkhole. However as discussed in the next section, this attack is not relevant in an OSN. Besides, if the HT count number does not confirm the number of appended signed nonce and counter fields from each node the CDP has passed through, then the CDP is rejected by the BS. Unless a malicious node deletes traces of a previous nodes entry, this attack is ineffective. However, deleting a previous nodes entry is non-trivial since signing of the nonce XORed with the counters is cumulative, and so a malicious node cannot tell which was a previous nodes counter or what the original nonce from the BS was. (2) If a malicious node tampers with the second field in a CDP revealing the next key in the key chain, then the CDP packet will not verify to its successors who in effect will drop the packet. This is similar to a node dropping a routing packet and leads to a blackhole, which may or may not be detected by the BS. This allows us to reduce the effect of malicious nodes to blackhole attacks, which is attractive because it results in a DoS which can be easily detected and addressed with route maintenance approaches. (3) For the same reasons as stated in (1), malicious nodes cannot trivially alter the nonce except if he is a CH. In any case, a manipulated nonce would not verify at the BS who would then initiate intrusion detection for this circuit.

In addition, since nodes decrypt, increment (by their counter value) and re-encrypt the nonce, if a malicious node arbitrarily deletes a previous nodes entry into the CDP packet, the computation on the final nonce will not verify at the BS. Furthermore, each nodes' signature (using individual keys) on the nonce prevents a malicious node from arbitrarily inserting false IDs in the CDP to lengthen a route, since he cannot manufacture an authentic individual key.

Key Freshness: Every era, the BS securely broadcasts the first commitment of a new network key chain $K_1^{1'}$, and every node increments its counter by 1. Nodes that are suspected to be malicious are excluded from receiving this updated key, and therefore excluded from participating in network protocols. If a suspected node proves his innocence (e.g., by running code verification [20]), updated keys are revealed to him so he rejoins the network. This process ensures that bad nodes are periodically weeded from routing in the network, and results in key freshness useful for preventing cryptanalysis.

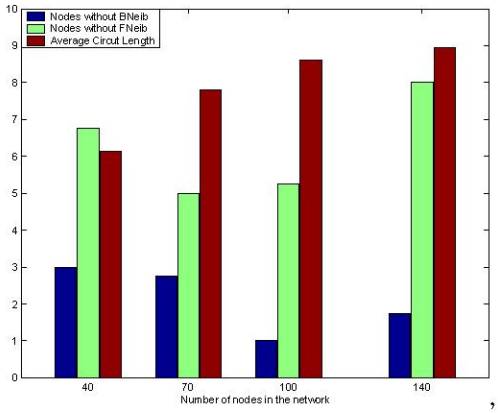

Fig. 4. A bar graph showing the number of nodes with no node in their BNeb and FNeb, as well as the average length of the BS-circuit lengths discovered, with $\delta = 10$, for $n = 40, 70, 100$ and 140.

V. Attack Analysis

We have shown that OPSENET prevents unauthorized participation, thereby preventing outsider attacks in which the attacker has no special access to the network. In this section, we focus on insider attacks, in which an attacker has gained access to the resources of a node, and can launch an attack as an authentic network participant. Insider attacks are more difficult to detect and prevent, and often, the best we can hope for is a graceful degradation of network performance as the number of compromised nodes grow. The two main insider routing disruptions we consider are sinkholes and blackholes. Various attacks have been shown to accomplish these [6].

Sinkhole Attacks: A sinkhole involves a malicious node striving to illegally attract traffic through itself by giving other nodes the impression that a high quality route exist through him to the BS. Once this is accomplished, the bad node can then launch selective forwarding, spoofing, packet altering or eavesdropping attacks. Insider sinkhole attacks we consider for the OSN scenario include Replay, Sybil and Wormhole attacks.

Replay Attacks: In a replay attack, a node caches prior authentic routing signals and replays them at a later time or a different location, in order to achieve a sinkhole. It is easy to see how OPSENET prevents replay attacks with the use of the nonce, incremental counter and key chain. Also, updating K_1^1 as mentioned for key freshness, will assist in mitigating this attack.

TABLE II

NUMBER OF VARIOUS LENGTH CIRCUITS FOUND FOR A NETWORK WITH $n = 250, \delta = 10$, AND $P_{CH} = 0.1$

BS-circuit length	2	3	4	5	6	7	8	9	10
Number of circuits	5	5	16	20	25	36	116	197	569

Wormhole Attacks: A wormhole is a powerful attack which involves two colluding nodes, in which one of the nodes tunnels packets through a low latency link to the other node in another path of the network, from which it easily launches a replay attack [21]. The aim of a wormhole in conventional RF networks is to achieve a sinkhole (and accompanying attacks) by making the wormhole node appear to have an efficient path back to the BS. The implicit assumption in this attack is that all links are bi-directional and routes are formed by reversing paths from which routing signals are received. This assumption does not hold in an OSN, thereby invalidating the effect of this attack. The authors in [22], propose directional antennas for detecting wormhole attacks, however they assume bi-directionality of network links. For OSNs the use of directionality is more suited to detecting and mitigating Sybil attacks.

Sybil Attack: In a Sybil attack [9], a single node presents multiple (false) identities for the purpose of confusing the routing scheme and leading to a possible sinkhole or blackhole. In OSNs, a malicious node trying to misrepresent location information may be identified by the BS as he constructs the network topology and validates nodes' location relative to other nodes. In addition, a node that successfully misrepresents its location effectively cuts himself off from the network since his predecessors will incorrectly orient their laser in the direction obtained from the RIP for that node.

Blackhole Attack: A blackhole involves a malicious node illegally attracting traffic to a non-existent route so that packets attempting to traverse such hops are not received by any node, and have to be dropped. An example of such an attack is the HELLO flood attack. Link layer acknowledgements and authentication help to mitigate against blackholes [6].

HELLO flood Attack: In this attack, nodes broadcast HELLO packets to announce themselves to their neighbors. In an OSN with directional lasers, the effect of HELLO flood attacks is irrelevant since this attack assumes link bi-directionality which implies reverse path routing. For an OSN with uni-directional links, this is not the premise on which routing circuits are built. Rather, lap-top class HELLO flooding [6] may be used as a jamming attack, most effective when the attacker floods a high powered laser beam in all directions using multiple optical transmitters. We do not address this laptop class of attack since it is in the physical layer, and we deal with a scenario in which an attacker is restricted to nodes of similar capabilities as the network nodes.

Identity Replication Attack: Identity replication, in which the same identity is used many times in multiple locations can be performed and defended against independently of the Sybil attack [9]. By registering each node's identity and location, the BS would detect that the same identity exists in multiple locations. Another approach is for the BS to centrally count the number of connections each node has, and revoke nodes with more connections than allowable maximum, thus countering node replication.

TABLE III

SAMPLE BS-CIRCUITS FOR CH_{61}; CH NODES ARE BOLD-FACED.

61* → **14***
61* → **8***
61* → 1 → 16 → **12***
61* → 1 → 16 → 11 → **12***
61* → 1 → 16 → 11 → **20***
61* → 1 → 16 → 11 → 3 → 2 → 7 → 40 → **20***
61* → 1 → 16 → 11 → 3 → 2 → 7 → 40 → 52 → **35***
61* → 1 → 16 → 11 → 3 → 2 → 7 → 40 → 52 → **8***
61* → 1 → 16 → 11 → 3 → 2 → 7 → 40 → 52 → **35***
61* → 1 → 16 → 11 → 3 → 2 → 7 → 40 → 52 → **65***
61* → 1 → 16 → 11 → 3 → 2 → 7 → 40 → 36 → **12***
61* → 1 → 16 → 11 → 3 → 2 → 7 → 40 → 36 → **20***
61* → 1 → 16 → 11 → 3 → 2 → 7 → 40 → 36 → **8***
61* → 1 → 16 → 11 → 3 → 2 → 7 → 40 → 36 → **35***
61* → 1 → 16 → 11 → 3 → 2 → **8***
61* → 1 → 16 → 11 → 3 → 2 → 21 → **12***
61* → 1 → 16 → 11 → 3 → 2 → 21 → 29 → 34 → **12***
61* → 1 → 16 → 11 → 3 → 2 → 21 → 64 → **12***
61* → 1 → 16 → 11 → 3 → 2 → 21 → 64 → 55 → **6***
61* → 1 → 16 → 11 → 3 → 2 → 21 → 64 → 55 → **20***
61* → 1 → 16 → 11 → 3 → 2 → 21 → 64 → 55 → **12***
61* → 1 → 16 → 11 → 3 → 2 → 21 → 64 → 55 → **8***
61* → 1 → 16 → 11 → 3 → 2 → 21 → 64 → 55 → **12***
61* → 1 → 16 → 11 → 3 → 2 → 30 → **20***
61* → 1 → 16 → 11 → 3 → 2 → 30 → **35***
61* → 1 → 16 → 11 → 3 → 51 → **14***
61* → 1 → 16 → 11 → 3 → 51 → 53 → **8***
61* → 1 → 16 → 11 → 3 → 51 → 53 → **35***
61* → 1 → 16 → 11 → 3 → 51 → 53 → 55 → **65***
61* → 1 → 16 → 11 → **46***
61* → 43 → **20***
61* → 43 → **35***
61* → 67 → 18 → **14***
61* → 67 → 18 → 25 → 28 → **8***
61* → 67 → 18 → 25 → 28 → 39 → 9 → 22 → 31 → **54***
61* → 67 → 18 → 25 → 28 → 39 → 50 → **54***
61* → 67 → 42 → **10***
61* → 67 → 42 → 60 → 49 → 17 → **8***
61* → 67 → 42 → 60 → 49 → 17 → 13 → **6***
61* → 67 → 42 → 60 → 49 → 17 → 13 → **8***
61* → 67 → 42 → 60 → 49 → 17 → 13 → **14***
61* → 67 → 42 → 60 → 49 → 17 → 13 → 38 → **8***
61* → 67 → 42 → 60 → 49 → 17 → 13 → **65***

VI. SIMULATION-BASED PERFORMANCE ANALYSIS

To evaluate the performance of our algorithm, we have developed `OPSENET` [23], a simulation test-bed written in C# and `matlab`, for specifically testing network and link layer algorithms in OSNs. Our simulation scenario involves n nodes deployed in a 1km $\times$1km area. Each node has a communication radius $r(n, \alpha)$ where r, n and α are parameters of the system. We use $\alpha = 40^o, \delta = 10$, and $P_{CH} = 0.1$ in all simulations, unless stated otherwise.

For our simulations, outlier nodes are ignored. Figure 4 is a bar graph that depicts the average percentage of outlier nodes (nodes without $FNEb$ and $BNeb$ denoted as $FNeb(\phi)$ and $BNeb(\phi)$, respectively), for a given network size. We see

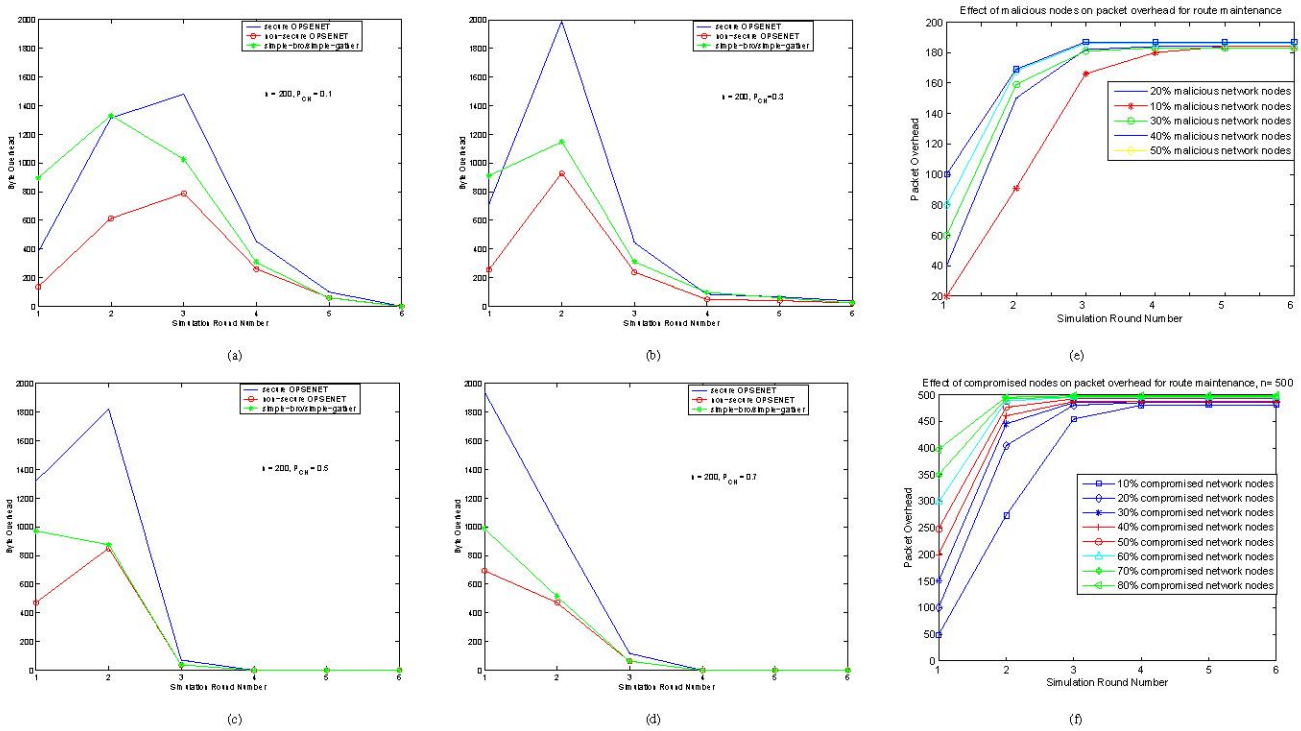

Fig. 5. (a-d) A comparative plot of byte overhead versus number of rounds of simulation for OPSENET, non-secure OPSENET and simple-bro/simple-gather algorithm, for network size of 200 nodes, (a) $P_{CH} = 0.1$ (b) $P_{CH} = 0.3$ (c) $P_{CH} = 0.5$, (d) $P_{CH} = 0.7$. (e-f) Graceful Degradation of OPSENET by measuring route maintenance overhead packets in the presence of compromised Nodes (ranging from 10% to 80% for (e) $n = 200$ and (f) $n = 500$

that as network size increases, the percentage of disconnected nodes decreases, so that for a densely deployed network, outlier nodes can be ignored. Also, it is interesting to note that in all our simulations, we have found $|FNeb(\phi)| \geq |BNeb(\phi)|$ consistently. This means that to further improve connectivity, we may look for ways to incorporate nodes who can hear, but cannot talk to a neighbor. We address this issue as a thrust of future research. Figure 4 depicts the average BS-circuit length, which as expected, increases with network size, with all other network parameters fixed. Table V shows the number of length-2 through length-10 BS-circuits found for a sample network deployment with 250 nodes and $P_{CH} = 0.1$.

A. Performance Analysis of OPSENET

We show a sequence of BS-circuits for a given CH obtained with a sample `OPSENET` network configuration of $n = 100$ nodes, labeled $(S_1 \cdots S_{100})$, and $P_{CH} = 0.1$. The set of CHs is: S_6 S_8 S_{10} S_{12} S_{14} S_{20} S_{35} S_{46} S_{54} S_{61} and S_{65}; nodes with no FNeb are: S_5 S_{45} S_{57} S_{68} S_{70}, while those with no BNeb is an empty set. The BS-Circuits found for CH_{61} in this example are shown in Table III. The average BS-circuit length is 8.2. We evaluate the following performance metrics:

1) Byte Overhead: This is the overall bytes generated by the protocol, which may be used to evaluate the energy requirement of the protocols. For analyzing byte overhead, we make the following assumptions: 64-bit keys and 8-bit nonce and counters are used, while the HT field is $\lceil \log \delta \rceil$ bits. A nodes' ID and position coordinates are each represented with $\lceil \log n \rceil$ bits. This means that for secure OPSENET, a new CDP packet is $\lceil \log \delta \rceil + 64 + 8$-bits, and accumulates $3\lceil \log n \rceil + 8$ for each hop traversed. Similarly, a new RIP packet from the BS is $3H\lceil \log n \rceil$ bits, where H is the number of hops of a given BS-circuit. As described, the RIP packet is decremented by $3\lceil \log n \rceil$ bits for each hop traversed. We assume that the encryption function used does not add extra redundant bits.

For non-secure OPSENET, we assume keying and nonce information are not transmitted, and compute the byte overhead accordingly. We compare secure and non-secure OPSENET with the combined byte overhead of `simple-bro/simple-gather` algorithms [2], since they both collectively achieve what non-secure OPSENET is built to do, i.e. uplink/downlink path discovery. Figures 5(a - d) show a comparative plot of byte overhead versus number of rounds of simulation for the three algorithms, for a network size of $n = 200$, while varying P_{CH} as $0.1, 0.3, 0.5$, and 0.7, respectively. We note from the plots that non-secure OPSENET always performs better than the other two algorithms. We observe that for fewer CHs, secure-OPSENET starts with a lower byte overhead than the `simple-bro/simple-gather`, but overtakes it as the number of rounds and/or P_{CH} increases. For all the protocols, byte overhead ramps up to a maximum, and then declines until zero. This is easily explained, since initially messages are broadcast, and explode as $O(\log n)$. However, as the network is saturated with the message, nodes who have "seen" a CDP

packet before drop them, the number of messages in the network decrease until all nodes in the network have received the packet and there are no more packets to be sent. Also, as expected, as P_{CH} increases, all the algorithms converge faster because, the higher the percentage of CHs in the network, the shorter the lengths of BS-circuits.

2) Graceful Degradation: Route maintenance overhead in the presence of compromised nodes: This measures the number of multicast packets sent in response to the BS's RM-Query during route maintenance for a percentage of randomly malicious network nodes. This metric enables us evaluate the degradation of network performance due to maintenance overhead in the presence of compromised nodes. Figure 5(e-f) depicts graphs of the packet overhead versus number of rounds of simulation for varying percentage of malicious nodes of a network (from 10% to 50%), with $n = 200$ and 500. We note the graceful degradation of the network performance as number of compromised nodes increase. We observe as expected that packet overhead is bounded by n.

VII. Related Work

A number of efficient and practical routing protocols have been proposed for sensor networks [24]. Many of these protocols attempt to minimize energy usage for the routing protocols while maximizing network life time. Protocols such as TinyOS, MCFA, SPINS, GEAR, SAR, Rumor Routing and Directed Diffusion [24] use planar multi-hop design with various optimization considerations including data aggregations, data dissemination latency and energy dissipation. Some other protocols such as LEACH [14] have considered a hierarchical design in their communication model. A few of these protocols have considered security, however they assume the links in the network are bi-directional, and therefore build routing trees via reverse path algorithms through parent nodes to the BS. This is the first paper to consider secure routing in a purely directional optical sensor network modeled as a random sector graph.

VIII. Conclusions

In this paper, we presented the design and evaluation of OPSENET, a novel efficient and secure routing for directed optical sensor networks that are modeled as random sector graphs. We base the design of OPSENET on symmetric cryptography using one-way hash chain and pre-deployed keying. We showed that OPSENET provides broadcast and per hop authentication, and is robust to non-smart attacker.

Acknowledgement

We would like to thank Kyle Marshall and Jim Griffin for the wonderful coding/debugging of the OPSENET simulator.

References

[1] J. Llorca, A. Desai, U. Vishkin, C. Davis, and S. Milner, Reconfigurable optical wireless sensor networks, In *Proc. SPIE vol. 5237, Optics in Atmospheric Propagation and Adaptive Systems VI*, J. D. Gonglewski and K. Stein, Eds., Barcelona, Spain, February 2004, pp. 136146.

[2] J. Diaz, J. Petit, and M. Serna, A random graph model for optical networks of sensors, In *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, pp. 186196, July- September 2003.

[3] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for smart dust. In *Proc. ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, Washington, August 1999, pp. 271278.

[4] J. Hill, R. Szewczyk, A. Woo, S. Holler, D. Culler and K. Pister. System architecture directions for networked sensors. *Proceedings of the ninth international conference on Architectural support for programming languages and operating systems*, pages 93–104, 2000.

[5] S. Teramoto and T. Ohtsuki. Optical wireless sensor network system using corner cube retroreflectors. 2005. *EURASIP Journal of Applied Signal Processing* 1, 39-44. (Mar. 2005).

[6] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *Proc. IEEE International Workshop on Sensor Network Protocols and Applications*, pp 113–127, May 2003.

[7] H. Chan and A. Perrig Security and Privacy in Sensor Networks . In *IEEE Computer Magazine*, October 2003.

[8] R. Anderson and A. Perrig. Key infection: Smart trust for smart dust. In *Proceedings of 12th IEEE International Conference on Network Protocols (ICNP 04)*, October 2004.

[9] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: Analysis and defenses. In *Proc. Symposium on Information Processing in Sensor Networks*, pp 259–268, Berkeley, California, 2004.

[10] J. McCune, E. Shi, A. Perrig and M. Reiter. Detection of Denial-of-Message Attacks on Sensor Network Broadcasts. In *Pro. of the IEEE Symposium on Security and Privacy*, May, 2005.

[11] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. *IEEE Computer Society Press*, 35(10):54–62, 2002. Los Alamitos, CA.

[12] S. Zhu, S. Setia and S. Jajodia. LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks. *10th ACM Conf. on Computer and Comm. Security (CCS '03)*, Washington D.C., Oct. 2003.

[13] M. Bohge and W. Trappe, An authentication framework for hierarchical ad hoc sensor networks, In *WiSe 03: Proc. of the 2003 ACM workshop on Wireless security*, New York, USA, 2003, pp. 7987, ACM Press.

[14] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocols for wireless microsensor networks. *Proc. Hawaaian Int'l Conf. on Systems Science*, January 2000.

[15] L. Buttyàn and J.-P. Hubaux. Report on a working session on security in wireless ad hoc networks. *ACM Mobile Computing and Communications Review*, 6(4), pp 1–17, November 2002.

[16] K. Sanzgiri, B. Dahill, B. Levine, and E. Belding-Royer. A secure routing protocol for ad hoc networks. In *In International Conference on Network Protocols (ICNP)*, Paris, France, November 2002.

[17] H. Chan, A. Perrig, and D. Song. Random Key Pre-Distribution Schemes. In *In IEEE Symposium on Security and Privacy.*, 2003.

[18] U. Okorafor and D. Kundur, Efficient Routing Protocols for a Free Space Optical Sensor Network, In *Proc. IEEE Int. Conference on Mobile Ad Hoc and Sensor Systems*, Washington, D.C., November 2005.

[19] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In *Proc. ACM Int. Conference on Mobile Computing and Networking*, pp 189–199, July 2001.

[20] A. Seshadri and L. Doorn, and P. Khosla and A. Perrig. SWATT: SoftWare-based ATTestation for Embedded Devices. In *2004 IEEE Symposium on Security and Privacy*.

[21] E. Shi and A. Perrig. Designing Secure Sensor Networks. In *IEEE Wireless Communications*, Vol 11(6), Dec 2004.

[22] L. Hu and D. Evans. Using Directional Antennas to Prevent Wormhole Attacks. In *Network and Distributed System Security Symposium*, San Diego, February 2004.

[23] OPSENET: Optical Sensor Network Simulator, *http://opsenet.tamu.edu*.

[24] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Comm.*, 11(6), pp 6–28, Dec 2004.

[25] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8), pp102–114, August 2002.

[26] C. Alvarez, J. Diaz, J. Petit, J. Rolim and M. Serna. Efficient and reliable high level communication in randomly deployed wireless sensor networks *MOBIWAC-2004*.

[27] G. Arzhantseva, J. Diaz, J. Petit, J.Rolim M. Serna: Broadcasting on network sensors communicating through directional antennaes. *International Workshop on Ambient Intelligencer (2003)*.

[28] W. Luh and D. Kundur Distributed Privacy for Visual Sensor Networks via Markov Shares, *Proc. 2nd IEEE Workshop on Dependability and Security in Sensor Networks and Systems*, Columbia, Maryland, April 2006.