Providing End-to-end Secure Communications in Wireless Sensor Networks

Wenjun Gu, Neelanjana Dutta, Sriram Chellappan and Xiaole Bai

***Abstract*—In many Wireless Sensor Networks (WSNs), providing end to end secure communications between sensors and the sink is important for secure network management. While there have been many works devoted to hop by hop secure communications, the issue of end to end secure communications is largely ignored. In this paper, we design an end to end secure communication protocol in randomly deployed WSNs. Specifically, our protocol is based on a methodology called differentiated key pre-distribution. The core idea is to distribute different number of keys to different sensors to enhance the resilience of certain links. This feature is leveraged during routing, where nodes route through those links with higher resilience. Using rigorous theoretical analysis, we derive an expression for the quality of end to end secure communications, and use it to determine optimum protocol parameters. Extensive performance evaluation illustrates that our solutions can provide highly secure communications between sensor nodes and the sink in randomly deployed WSNs. We also provide detailed discussion on a potential attack (i.e. biased node capturing attack) to our solutions, and propose several countermeasures to this attack.**

***Index Terms*—Sensor Networks, Security, Key Management**

I. Introduction

Many Wireless Sensor Networks (WSNs) are being envisaged in military, emergency and surveillance applications today, where sensor nodes need to send sensed data to the sink. In many applications under hostile environment, sensor nodes cannot be deployed deterministically and thus are randomly deployed into the field. An important requirement in network management of many mission critical applications is to secure end to end sensor networks data from being eavesdropped by the attacker. While there have been many works devoted to hop by hop secure communications in WSNs, the issue of end to end secure communications is largely ignored. This is mainly due to the fact that there exist two intuitive approaches to provide a high degree of end to end secure communications in WSNs:

- The first one is distributing a unique pairwise key into each sensor and the sink prior to deployment, and letting each sensor use this pairwise key to encrypt the communications with the sink;
- The second one is simply providing hop by hop secure communications between neighboring sensors in the network. It is in general believed that in this way end to end secure communications can naturally be achieved via hop by hop encryption/decryption.

Wenjun Gu is currently with Microsoft, USA - E-mail: wenjgu@microsoft.com. Neelanjana Dutta and Sriram Chellappan are with The Department of Computer Science at Missouri University of Science and Technology, Rolla, USA - Email: nd2n8@mail.mst.edu and chellaps@mst.edu. Xiaole Bai is with The Department of Computer and Information Science at University of Massachusetts, Dartmouth, USA - Email: xbai@umassd.edu.

The first approach has critical limitations in multi-hop WSNs since it precludes the possibility of intermediate sensors performing encryption/decryption along the path. This feature is necessary for interpreting and aggregating data at intermediate sensors to save energy (a critical requirement in WSNs), authenticating received data to defend against fake packets injection attack, denial of service attack etc. Hence in WSNs, we need to use hop by hop based encryption/decryption in providing end to end secure communications.

The second approach works well if all links in the network are highly resilient. However, it is very hard, if not impossible, to achieve high resilience for *all* the links in randomly deployed WSNs. This is due to the inherent resource limitation of sensor nodes, the nature of random deployment and the existence of malicious attacks. In fact, with random key pre-distribution (*RKP*) [1] based schemes, a majority of links in the network have low resilience under reasonable memory constraint and even under mild attack strength, which restricts the room for providing a high degree of end to end secure communications. In the following, we give an example to illustrate this fact.

In order to provide secure communications between neighboring nodes in randomly deployed WSNs, Random key pre-distribution (*RKP*) was proposed [1]. In its simplest version, each sensor is pre-distributed with k distinct keys randomly chosen from a large pool of K keys. After deployment, neighboring nodes use these pre-distributed keys to establish a pairwise key between themselves. Communications between neighboring sensors in each hop are encrypted/decrypted using these pairwise keys. A host of key management protocols have been proposed based on key pre-distribution [2], [3], [4], [5], [6], [7], etc., each one improving upon one or more features of the basic scheme.

The resilience of each hop (link) can be reflected by the number of shared pre-distributed keys in the link. It is known that under uniform key distribution, i.e. each sensor pre-distributed with equal number of keys, will achieve maximum average number of shared pre-distributed keys in each link. However, there is an inherent limitation in uniform key distribution as demonstrated in Fig. 1. In Fig. 1, we have 1000 nodes randomly deployed in a circular network with radius $500\ meters$, where $k = 40$, $K = 10000$ and communication range of each node is $100\ meters$. We can see that a majority of links have low resilience (i.e., small number of shared keys), while the percentage of links that are highly resilient is quite low. This clearly restricts the room for routing protocols to

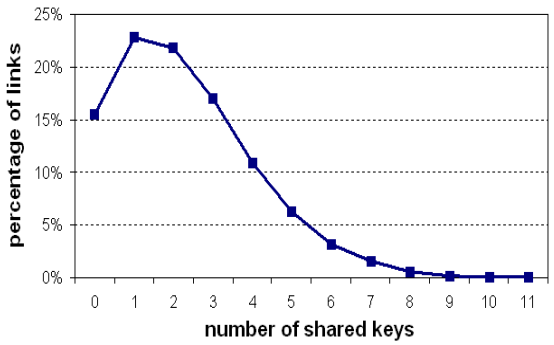

Fig. 1. Percentage of links with varying number of shared keys.

choose more resilient links during end to end communications. Installing more keys into each node is not always preferable since it enables the attacker to disclose more keys upon node captures, which could again compromise the link resilience.

Our Contributions: In this paper, we design an end to end secure communication protocol in randomly deployed WSNs. The contributions of our work are four-fold:

- We propose a methodology called *differentiated key pre-distribution* for end to end secure communications in randomly deployed WSNs. Our protocol is based on this methodology. The core idea of the methodology is to pre-distribute different number of keys to different nodes. By distributing more keys to some nodes, the links between those nodes tend to have much higher resilience than the link resilience under uniform key pre-distribution. These high resilient links are preferred during routing to enhance the end to end secure communications. For fairness in analysis, we keep the average number of keys per node in our scheme the same as that in uniform key pre-distribution. Assuming that the probability of node capture is the same for all nodes, the attack impact (i.e., expected number of unique keys disclosed under node capturing attack) remains the same in both uniform and our heterogeneous key distribution scheme, making performance comparisons fair. Furthermore, we also discuss impact of biased node capturing attacks in this paper.
- Based on the above methodology, we design a new end to end secure communication protocol in WSNs. In our protocol, links with high resilience are given preference compared with links with low resilience during routing to the sink node. Besides, we apply alternative path routing among high resilience links to achieve a good balance between end to end secure communications and network lifetime.
- We conduct a rigorous theoretical study on the proposed protocol, and determine the optimal way to differentially pre-distribute keys into nodes to maximize the value of P_{e2e}. All our results are further validated using extensive simulations, which confirm that end to end secure communications can be significantly improved by our solutions.
- We also conduct a thorough investigation on the possibilities of certain advanced attacks (called biased node capturing attacks) to our proposed scheme. We find that our scheme does not incur serious vulnerability to such advanced attacks. Nevertheless, to further minimize the vulnerability, we propose a variety of countermeasures against such attacks.

Paper Organization: The rest of the paper is organized as follows. In Section II, we discuss related works. We discuss our differentiated key pre-distribution methodology, and present our end to end secure communication protocol in Sections III and IV respectively. We study advanced attacks in Sections V, and finally conclude this paper in Section VI.

II. Background and Related Work

A. RKP based Schemes

In this section, we provide a brief background on random key pre-distribution (RKP) schemes, attack models and performance metrics in randomly deployed WSNs.

Basic RKP Scheme: A well accepted scheme for secure communications in randomly deployed WSNs is the random key pre-distribution (*RKP*) scheme [1]. There are two stages in the *RKP* scheme. At the *key pre-distribution* stage, each node is pre-distributed with k distinct keys randomly chosen from a large pool of K keys, and then nodes are randomly deployed. At the *pairwise key establishment* stage, each node first obtains its neighborhood information. If two neighbors share one or more pre-distributed keys, they can establish a pairwise key in between directly. To do so, one node can generate a random pairwise key and send it to its neighbor encrypted with their shared keys. For two neighbors that do not share pre-distributed key, they will use neighboring nodes, called *proxies*, to construct key paths for pairwise key establishment using the above process.

Variants of RKP Scheme: A host of protocol variants have been proposed based on the above idea of key pre-distribution in sensor networks. Networks targeted by existing literature include homogeneous sensor networks, heterogeneous sensor networks and also (more recently) mobile sensor networks. While some works focus primarily on extensions to the basic scheme, other works focus on more involved extensions. In this section, we provide a detailed description of well known and recent works in this area.

We first discuss important related work on key management in homogeneous sensor networks, where the number of keys distributed per node is the same, and the network topology is flat. In order to overcome the low resilience of the basic key management protocol in [1], in [2] and [7], it is proposed to use multiple key paths to enhance the resilience of the link between two nodes. However, exploiting the high resilience key paths for routing is not discussed. In [4] and [5], keys are pre-distributed according to some well known optimization designs, which helps increase chances of key sharing between

nodes. In [3] and [6], each sensor is pre-distributed with k key structures (vectors or polynomials) from a key structure pool, where each key structure has degree λ. In such schemes, no key structure is disclosed until at least λ+1 nodes pre-distributed with this key structure are captured. As such, the attacker has to disclose many nodes before a key structure is disclosed. When $\lambda = 0$, this scheme degrades to the basic *RKP* scheme. In [8], a similar technique is proposed called Triangle based key management, in which deployment information about expected locations of the nodes and polynomial based key predistribution are used. The idea is to divide the network into triangular cells, each of which is associated with a unique random bivariate polynomial. Each sensor node is provided with a set of polynomials belonging to the cells nearest to the nodes. Based on this structure, direct key and multi-hop (indirect) key establishment protocols are developed, along with mechanisms for key revocation and additions.

There are some existing works on key management in heterogeneous sensor networks. The work in [9] is most similar to ours. In [9], cluster heads in the network are distributed with many more keys than normal sensors. However, in the analysis of [9], cluster heads are assumed to be equipped with a fast encryption/deletion algorithm to protect their supplementary keys from compromise. The authors also admit that the successful capturing of a cluster head node can severely compromise the resilience of their scheme. The resilience degradation is our protocol is much more graceful under attacks. Secondly, the protocol proposed in [9] is only for pair-wise key establishment. We propose two resilience aware routing protocols: data centric and location centric protocols in this paper that further exploit the idea of differentiated keys among sensors. Also, the work in [9] only mentions biased node capture attacks, while we discuss biased node capture attacks and design countermeasures against them in this paper. In [10], a key management scheme is proposed, wherein sensor nodes are organized into multiple hierarchies using a tree structure. Three types of keys are provisioned initially. The keys are divided in different categories such as cluster key (shared among all members of the cluster), intermediate key (shared between a smaller subset of cluster members) and private key of each sensor (used to communicate with cluster head). In our scheme, we do not use different cluster head keys, but rather use same key pools that reduces complexity during key pre-distribution and subsequent pair-wise set-up. Also, the scheme in [10] suffers from poor resilience under node compromises and all keys have to be refreshed very quickly. In [11], a new protocol for cluster-head selection is proposed, and ideas are proposed for key additions and revocations under network dynamics via one-way hash functions. Trust protocols are also designed during initial cluster setup. Periodic use of hash functions for key revocation and refreshment however can be very energy consuming. In [12], a key management scheme is proposed for heterogeneous sensor networks wherein sensors use initial keys to securely establish trust with their peers. This happens under the assumption of a safe period when there are no attacks. Subsequently, energy efficient key management protocols are designed as a function of desired degree of secure communications.

In [13], more complicated security designs are envisioned for secure sensor networks like non-group confidentiality (nodes outside cluster should not be able to decrypt messages within cluster), forward confidentiality (compromised nodes should not have keys to decrypt future messages) and backward confidentiality (new nodes should not have keys to decrypt previous messages). The paper assumes a safe period when secrets are established. Techniques are proposed for key revocation under compromises, although it is not quite clear as to how compromises can be detected. A similar idea is also proposed in [14], for similar requirements except that bi-variate polynomials are used instead of symmetric keys. In [15], a robust and secure key management scheme for hierarchical sensor networks using self-healing mechanisms is proposed. The focus is on group key distribution, wherein the idea is a combination of reverse and forward hash chains. The paper also deals with attacks against individual nodes and group heads. The protocol ensures security of un-compromised nodes until a minimum number of nodes are compromised in the same group. In [16] also, protocols are proposed for group communications among sensors and cluster-heads. The approach uses symmetric keys throughout, and also proposes a protocol for cluster-head selection to maximize chances of key sharing with sensors in the group. Protocols are also designed for authentication of packets and nodes, and confidentiality for important packets in the network. Other works like [17], [18], [19] propose ideas for key management in hierarchical sensor networks emphasizing on aspects like key updation and revocation, energy efficiency, and mitigating attacks like guessing attacks, replay attacks, man-in-the-middle and denial of service attacks. Unfortunately, the downsides of such works are in the complexity in key pre-distribution and subsequent pairwise key set-up, which increases overhead. Furthermore, it is not quite clear as to how such schemes perform under biased node capture attacks. Nevertheless, we believe that the works referenced above are orthogonal to our contributions in this paper which focuses primarily on data confidentiality of end-to-end secure communications. Extending our techniques for incorporating additional security requirements within the network would an interesting area of future work.

In the recent past, there have been a host of orthogonal dimensions where random key pre-distribution has been adapted in sensor networks. While works in [20], [21], [22], [23], [24], [25], [26], [27] use deployment knowledge (i.e., partial knowledge of sensor locations in the network) to enhance pair-wise key set-up among nodes, exploiting traffic models in sensor networks for secure communications is studied in [28] based on the premise that traffic patterns are governed by network topologies which is exploited during key set-up and data delivery. There have also been works in sensor network key management where principles of Genetic Algorithms are used for key management [29], [30]. In [30], genetic algorithms are used to optimize memory usage, power control and computational security in sensor networks. In [29], genetic algorithms are used to optimize network performance from the perspective of operation cost, security and survivability.

More recently, an area of interest that has picked up in the sensor network community is mobile sensor networks [31],

[32], [33], [34]. There have been some recent efforts on key management in mobile sensor networks. In [35], a pair-wise key management in sensor networks is proposed where sensors can move from one network to another. If a sensor is allowed to roam from one network to another, it has to communicate with different base stations, and hence it should have shared keys with the new base stations. Attackers might fake as a new incoming sensor to a network to know the key of the network base station. So the base station needs authentication of incoming nodes. The paper proposes a secure authorization scheme and pair-wise key establishment between the roaming sensor and the new base stations that it needs to communicate with. In [36], security, integrity and authentication services in wireless sensor and actor networks are studied. Such networks consist of static sensors and resource rich actor nodes (that are also mobile), which are responsible for more sophisticated responses to sensed events without human intervention. The authors propose to divide the wireless sensor and actor network into upper and lower layers, with the lower layer using symmetric key management, and the upper layer using asymmetric key cryptography. Similarly in [37], a scheme is proposed for group based key management that can be used for both static and mobile sensor networks. The objective is to provide secure communication, data confidentiality using secure data aggregation and resilience. The network is divided into clusters and homomorphic encryption is used. The scheme supports inter and intra cluster roaming of sensors.

There have also been some recent works on applications of random key pre-distribution for real life sensor network missions. In [38], a sensor network key management is proposed for process control systems. This paper talks about secure key management in Process Control System or Supervisory Control and Data Acquisition system against node capture attack to secure both forward (future key secrecy to captured nodes) and backward (past key secrecy to new incoming nodes) secrecy. The paper proposes a scheme that updates shared symmetric keys between a node or group of nodes and network manager of PCA/SCADA. The work specially focuses on enhancing security in the remote fields which are the weakest components in SCADA. The attack model is node capture attack which might lead to software corruption, impersonation attack, future key disclosure etc. They use a hash key chain for group key update and pair-wise key update. In [39], a mathematical analysis of random key pre-distribution in sensor networks is attempted, from the perspective of network connectivity. The authors conduct a mathematical analysis to show that the number of communication links needed for a sensor to assure complete network connectivity is very large in real life applications. They do an analysis then showing that under node failures, faults and interference, the actual number of neighbors can be fewer in real-time sensor network operations. This work could potentially lead to optimizations in real-life sensor network missions.

Orthogonally, there are also works that address end-to-end secure communications in sensor networks without random key pre-distribution techniques. One particularly interesting work is [40], that primarily focuses on end-to-end data confidentiality by performing intermediate data aggregation via homomorphic encryption techniques. In their technique, each sensor can derive its own private key based on a master secret, which is only known to the sink. Then, all data from sensors is encrypted with keys of other sensors via a homomorphic encryption technique that allows aggregation on encrypted data hop by hop, which can then be recovered by the sink. The downside of such attempts is that while in-network aggregation is done, in-network processing (encryption/ decryption) of data cannot be directly accomplished. As we know, there are several advantages with in-network processing of data like in-network (and local) aggregation, localized verification of data trust and integrity, local filtering of malicious data etc. Furthermore, in applications where the sink needs to know individual data (and not just aggregates), the work in [40] has limited applicability. Our work is different in the sense that we are focusing on end-to-end secure communications via a combination of key management and routing techniques in the network, while still retaining the advantages of in-network aggregation.

Attack Models: In the standard attack model used in secure communications in WSNs [2], [1], [7], etc., the attacker launches two types of attacks. In *node capturing* attack, the attacker physically captures a certain percent of sensor nodes, and is able to disclose the pre-distributed and pairwise keys stored in those captured nodes. The sink node is assumed to be well protected and cannot be captured. In *link monitoring* attack, the attacker monitors all wireless links after deployment. Clearly, all communications of captured nodes are deciphered by the attacker. Furthermore, by combining the disclosed pre-distributed keys and messages recorded, the attacker can infer some pairwise keys between other uncaptured nodes. The attack model used in our paper is one where the attacker launches both *node capturing* and *link monitoring* attacks.

While some works like [41], [42] assume a safe period (no node capture) after deployment, this assumption may not be always realistic in practice, and this attack model is not widely adopted. Note that when multiple key paths are used to establish a pairwise key on a link, that pairwise key (link) is not compromised until all the key paths are compromised. The above attack model has a salient feature in that it is hard to detect the attacker. This is because the attacker only passively monitors traffic after nodes capture, and does not send out any traffic actively. In [43], a variety of active attack models are proposed against routing protocols, which require the attacker to actively send out fake/modified messages. Dealing with those attacks is orthogonal to our work, and thus is out of the scope in this paper.

Performance Metrics: Note that while our goal is end-to-end secure communications in sensor networks, we are (as pointed before) still interested in local in-network data processing among sensor nodes during network operation. Consequently (as in standard *RKP* schemes), we use two standard metrics: *connectivity* and *resilience* to evaluate our key management scheme. Connectivity is the probability that two physical neighbors can establish a pairwise key between them. Note that while the above definition refers to local connectivity and is the standard metric, one could also define global (end-to-end) connectivity as the probability that the entire network is securely connected, or as the number of

nodes in the largest connected component of the secure network. Global connectivity can be inferred by local connectivity [44], we focus only on local connectivity (henceforth called connectivity) in this paper. Resilience is the probability that a pairwise key (link) between two nodes is not compromised under attack.

B. Routing Protocols in WSNs

Routing in wireless sensor networks has some differences from that in traditional wired and wireless ad hoc networks due to resource constraints, faults/failures etc. There are two main paradigms of routing protocols in WSNs: location-centric routing and data-centric routing. Other paradigms include hierarchical routing and security aware routing.

Location-centric routing: Greedy Perimeter Stateless Routing (GPSR) [45] is a well known location centric routing protocol. In GPSR, beacon messages are broadcast by each node to inform its neighbors of its position. GPSR assumes that sensors can determine through separate means the location of the sink. Each node makes forwarding decisions based on the relative position of the sink and its neighbors. In general, the neighbor that is closest to the sink is chosen.

Data-centric routing: Directed diffusion [46] is the most well known data centric routing protocol, in which the sink sends queries to all nodes and waits for data from the nodes satisfying specific requirement (e.g., located in selected regions, sensing data meet certain criteria, etc). In order to create a query, an interest is defined using a list of attribute-value pairs such as name of objects, geographical area, etc. The interest is broadcast through the network, and used by each node to compare with the data received. The interest entry also contains several gradient fields. A gradient is a reply link to a neighbor from which the interest was received. By utilizing interests and gradients, paths are established between sensors and the sink. Several paths may be established, and one of them is selected by reinforcement.

Other Paradigms: Two other paradigms for routing in WSNs are hierarchical [47] and security aware routing [48], [49]. In the former, certain nodes are either pre-assigned or chosen at run time to be cluster heads. Routing takes place on two planes: sensor to sensor towards cluster head and cluster head to cluster head towards the sink. In this paper, we focus on flat networks. The issue of hierarchy is discussed in Section IV-D. In security aware routing [48], [49], nodes are expected to route packets in the presence of attacks like fake packets injection, selective forwarding, etc. Such works, while definitely important are orthogonal to our work here, which focuses on securing communications from node capture and eavesdropper.

III. Methodology

In this section, we will introduce our differentiated key pre-distribution methodology. In order to provide a high quality of end to end secure communications, it is clear that we should enhance the resilience of individual links in the network. An intuitive way to do so is to increase the number of keys pre-distributed into each node (k). When the number of shared keys in each link increases, resilience seems to increase since all shared keys have to be disclosed to compromise the link.

However, such a solution is counter-productive in reality. When k increases, more keys are disclosed per node capture. The compromise of only a small percent of nodes can disclose many more keys to the attacker, which compromises the resilience of links. We need an approach by which link resilience can be enhanced without the downside of disclosing more keys to the attacker. On the other side, the number of pre-distributed keys (k) is also subject to the memory constraint of sensor nodes.

In this paper, we propose a methodology called *differentiated key pre-distribution* to enhance the quality of end to end secure communications in randomly deployed WSNs. Our methodology is based on the observation that links in the network are not equally important with respect to secure communications. Only the links used for data transmission have impacts on security. The core idea of our methodology is to pre-distribute different number of keys to different nodes. We keep the average number of keys per node the same as that in uniform key pre-distribution, so that the attacker impact (e.g., average number of keys disclosed per node capture) remains the same. By distributing more keys to some nodes, the links between those nodes tend to have much higher resilience than the link resilience under uniform key pre-distribution. These high resilient links are preferred during routing to enhance the end to end secure communications.

We illustrate our methodology further using the example in Fig. 1, where 1000 sensors are deployed randomly in a WSN under the same scenario. We divide the 1000 nodes into two classes, with 200 nodes in the first class and 800 nodes in the second. We distribute $k_1 = 80$ keys in each first class node and distribute $k_2 = 30$ keys in each second class node. As such, the average number of keys per node here (40) is the same as that in Fig. 1, where k is the same for all nodes. In Table 1, we show the impacts of our methodology. We can see that when we apply our differentiated key pre-distribution for the above setting, the number of high resilient links (those with large number of shared keys) dramatically increases, with the cost that the number of low resilience links also increases. This is because compared with the link resilience in traditional RKP schemes with uniform key pre-distribution, the links between two first class nodes in our scheme tend to have higher resilience, while those between two second class nodes tend to have relatively lower resilience. When those high resilient links are preferred during routing path selection, the end to end security performance can be enhanced significantly.

In order to use this methodology to provide end to end secure communications between sensor nodes and the sink in randomly deployed WSNs, the following important questions need to be addressed:

- *How to determine the parameters in key pre-distribution?* We need to determine the number of node classes, the number of nodes in each class, and the number of keys distributed into nodes in each class. Determining the optimal values of these parameters needs a rigorous derivation of end to end secure communication perfor-

TABLE I
INCREASE OF THE NUMBER OF LINKS WITH DIFFERENT NUMBER OF SHARED KEYS UNDER DIFFERENTIATED KEY PRE-DISTRIBUTION

# of shared keys	0	1	2	3	4
# of links increase	54%	−8%	−20%	−29%	−19%
# of shared keys	5	6	7	8	>8
# of links increase	−2%	25%	56%	183%	475%

mance.

- *How to pre-distribute different number of keys into different classes of nodes?* An intuitive way is always choosing keys randomly from the key pool regardless of node class. Is there any better way to achieve higher resilience?
- *How to do routing given links having different resilience?* In this situation, the length of routing path and energy balancing are not the only factors to consider during routing path selection. Link resilience also plays a role. Care should be taken to make a good balance among these factors.
- *Will there be any new attack to our proposed scheme?* To achieve better security performance, nodes with more pre-distributed keys are likely to be selected to forward more traffic. The attacker may prefer choosing those nodes to capture. How can we defend against such attack?

IV. END-TO-END SECURE COMMUNICATION PROTOCOL

In this section, we will address the first three questions pointed out above. The last question will be addressed in Section V. In particular, we will present our end to end secure communication protocol based on the methodology above. Our protocol consists of two components: differentiated key management and resilience aware routing. Table 2 lists the parameters in protocol description and their notations. Theoretical analysis is then conducted to help determine optimal protocol parameters.

A. Differentiated Key Management

Our differentiated key management consists of two stages: *key pre-distribution* and *pairwise key establishment*. The main difference between our key management protocol and traditional RKP based key management protocols lies in the stage of key pre-distribution.

Key Pre-distribution: We study a network with N sensor nodes and one sink node. The N sensor nodes are divided into c classes, each of which has n_i ($1 \leq i \leq c$) nodes. We call the sensors in the i^{th} class as class i nodes. We then pre-distribute k_i ($1 \leq i \leq c$ and $k_1 \geq k_2 \geq \cdots \geq k_c$) unique keys chosen from a large key pool with size K into each class i node, detail of which will be discussed in the following. On the other side, the sink node is pre-distributed with all K keys in the key pool. After this procedure, the sink node is deployed strategically at certain position, while the N sensor nodes are deployed randomly in the network. The N sensor nodes will execute the following protocols for pairwise key establishment and routing.

We detail our key pre-distribution in the following. For each class 1 node, its k_1 unique keys are chosen randomly from the key pool. However we use a semi-random way to distribute keys into all other nodes to increase the chance of key sharing between these nodes and class 1 nodes. For a node in class i ($i > 1$), $\lfloor k_i/n_1 \rfloor$ keys are first chosen randomly from the distributed keys in each of $n_1 - (k_i - \lfloor k_i/n_1 \rfloor \cdot n_1)$ class 1 nodes, which are chosen randomly from all n_1 class 1 nodes. For the remaining $k_i - \lfloor k_i/n_1 \rfloor \cdot n_1$ class 1 nodes, $\lceil k_i/n_1 \rceil$ keys are chosen randomly from the distributed keys of each node. We define $\lfloor x \rfloor$ as the largest integer no more than x, and define $\lceil x \rceil$ as the smallest integer no less than x. If some of the chosen keys are the same, the redundant keys will be re-chosen until all k_i keys are distinct.

We illustrate with a simple example. Let $N = 100$, $c = 2$, $n_1 = 20$, $n_2 = 80$, $k_1 = 80$, $k_2 = 30$. The following is the key pre-distribution procedure. For each of the 20 nodes in class 1, we choose 80 distinct keys randomly from the key pool. For each of the 80 nodes in class 2, we choose 30 distinct keys as follows. We first randomly classify class 1 nodes into two types. Type A has 10 (i.e., $n_1 - (k_2 - \lfloor k_2/n_1 \rfloor \cdot n_1)$) nodes, and Type B has 10 (i.e., $k_2 - \lfloor k_2/n_1 \rfloor \cdot n_1$) nodes. Now, $\lfloor k_2/n_1 \rfloor = 1$ key is chosen randomly from the pre-distributed keys in each of the 10 Type A class 1 nodes above, and is distributed into the class 2 node under discussion. Then, $\lceil k_2/n_1 \rceil = 2$ keys are chosen randomly from the pre-distributed keys in each of the 10 Type B class 1 nodes above, and are distributed into the class 2 node under discussion. At this point, the class 2 node has 30 keys distributed. If these 30 keys are unique, key distribution is over. Otherwise, we redo the preceding 2 steps for the duplicate keys until all 30 keys are unique. Note that since $k_1 > k_2$, uniqueness can always be guaranteed.

By distributing keys in this way, we guarantee k_i unique keys are distributed, and the number of keys chosen from each class 1 node are balanced and differs by at most 1. The reason we pre-distribute keys for class i nodes in the above semi-random way instead of purely randomly is two folded. First, we can enhance the probability that a class i node shares key with a class 1 node. Second, we do not decrease the probability that a class i node shares key with a non-class 1 node. Both facts are confirmed by our simulation, and can help increase link resilience. Besides, pre-distributing keys for non-class 1 nodes in the above way will not decrease the *effective key space* much, which is defined as the number of keys in the key pool that are distributed in at least one sensor node. This is because when the values of n_1, k_1 and K are carefully chosen, the number of unique pre-distributed keys among the n_1 class 1 nodes is already close to K.

Pairwise Key Establishment: Once nodes are pre-distributed with keys and deployed, they start to discover their neighbors within their communication range r via local communication, and obtain the key IDs of their neighbors'

TABLE II
PROTOCOL PARAMETERS

Notation	Protocol parameter
S	network area ($= \pi R^2$)
r	communication range
N	number of nodes in the network
c	number of node classes
n_i	number of class i nodes ($1 \leq i \leq c$)
k_i	number of keys pre-distributed in class i node ($1 \leq i \leq c$)
K	number of keys in key pool
N_c	number of captured nodes

pre-distributed keys. With the above information, each node constructs all the one-hop and two-hop key paths to all its neighbors. If node i shares pre-distributed keys with a neighbor j, there is one direct key path with one hop between them. However, node i will also construct all the two-hop key paths with each of its neighbors, regardless of whether a one-hop key path has been constructed or not, to enhance the link resilience (the attacker has to compromise all key paths for a link between two nodes in order to compromise this link). Suppose node i wants to construct all two-hop key paths with node j now. To do so, node i sends a request to its neighbors, containing the node IDs of i and j. After a neighboring node m receives the request, it checks if it shares pre-distributed keys with node i and shares pre-distributed keys with node j. If both conditions are satisfied, node m sends a reply back to node i. In this way, a two-hop key path $i - m - j$ is constructed. If possible, other two-hop key paths are also constructed as above. After node i constructs all two-hop key paths to node j, node i will generate multiple random key shares, and transmit each key share on each key path. Key shares are encrypted/decrypted hop by hop by a combination (e.g., XOR) of all shared keys on that hop. Ultimately, the pairwise key between nodes i and j is a combination of all the key shares (e.g, XOR) transmitted. Nodes also estimate and store the number of protection keys for each link as follows. Assume there are p two-hop key paths between i and j, each with the help of proxy s_l $(1 \leq l \leq p)$, and denote $k(i,j)$ as the number of shared keys between i and j. The number of protections keys between i and j ($key(i,j)$) is,

$$key(i,j) = k(i,j) + \sum_{l=1}^{p} min(k(i,s_l), k(s_l,j)). \quad (1)$$

We calculate $key(i,j)$ in this manner because the resilience of a two-hop key path is mainly decided by the weaker link (the one with fewer shared keys). The larger the number of protection keys for a link, the more resilient is the link in general.

B. Resilience Aware Routing

In this section, we will describe how to incorporate our differentiated key pre-distribution with popular WSN routing protocols for end to end secure communications. We particularly focus on one popular location centric routing protocol and one popular data centric routing protocol. Incorporation with other routing paradigms is similar. The basic idea is to *tune* the routing protocols such that they consider link resilience as a metric during routing. In order to prevent overuse of a few nodes, we will let nodes choose several next hop nodes, and use one at each time to prolong network lifetime.

Extensions to location centric routing protocol: The location centric routing protocol we extend is GPSR [45]. In traditional GPSR, each node chooses a neighbor as the next hop that is closest to the sink. In order to achieve high end to end secure communications without compromising network lifetime, we extend GPSR protocol as follows. Each node i assigns a weight to all its secure neighbors (neighbors with which a pairwise key is established) that are closer to the sink than itself. We denote $U(i)$ as the set of node i's secure neighbors that are closer to the sink than itself, and recall $key(i,j)$ is the number of protection keys for the link between nodes i and j, we assign weight to each node j in set $U(i)$ as,

$$w_j = \frac{key(i,j)^\alpha}{\sum_{m \in U(i)} key(i,m)^\alpha}. \quad (2)$$

Here w_j is the probability that i chooses j as the forwarder. When $\alpha = 0$, all nodes in $U(i)$ are given equal priority regardless of link resilience. When α is positive, more resilient links are given higher priority. When α approaches infinity, only the most resilient links are chosen for routing. An intermediate value of α can be used to achieve a good balance between security and lifetime, which can be decided by security policy and other factors. For example, a large value of α can be chosen when high resilience is preferred and energy consumption imbalance is not a serious issue, while a small value of α can be chosen when energy is limited and energy consumption balance is critical. We will study the sensitivity of security and lifetime to α in Section IV-E.

Extensions to data centric routing protocol: In traditional minimum hop routing protocol [50], a variant of Directed Diffusion routing protocol, a node will choose a neighbor on the minimum hop path to the sink. We can extend this protocol in a similar way as above. During the next hop determination process, packets are forwarded only on the minimum hop secure paths. A secure path consists of links that have pairwise keys established. We denote the set of neighbors on the minimum hop secure path of node i by $U(i)$. Note that in a relatively dense network, there could be several minimum

hop secure paths between node i and the sink. Node i then assigns a weight w_j to each of its secure neighbors j in the set $U(i)$. The expression of w_j is given in (2).

C. Key Pre-distribution Parameters Determination

In this section, we discuss how to determine the design parameters in our proposed protocol above, namely number of node classes (c), number of nodes in each class (c_i) and number of keys in class i nodes (k_i). Our parameter determination is primarily based on the derivation of the end to end security metric P_{e2e}, which is detailed below.

1) Analysis Architecture: Recall that P_{e2e} is the probability that the sink can receive a message from a sensor without it being disclosed to the attacker. We focus on the extended GPSR described above. To simplify the exposition, we here assign an overwhelming weight (α approaches infinity) to the most secure neighbor such that all traffic will be forwarded by it. When there are more than one such nodes, the one closest to the sink is chosen. We assume the number of captured nodes (N_c) is known, which can be obtained from worst case estimation or historical experience.

Let $P_{e2e}(\ell)$ denote the resilience for a path from a node to the sink with geographic distance ℓ. Then P_{e2e} can be obtained as,

$$P_{e2e} = \frac{1}{\pi R^2} \int_{\ell=0}^{R} P_{e2e}(\ell) 2\pi \ell d\ell. \tag{3}$$

Here, we assume the network is of a disk shape centered at the sink with radius R. P_{e2e} for the networks with other shapes can be obtained accordingly.

For a specific ℓ, $P_{e2e}(\ell)$ can be expressed as,

$$P_{e2e}(\ell) = \sum_{h=1}^{\infty} P(h|\ell) P_{path}(h), \tag{4}$$

where $P(h|\ell)$ is the conditional probability that a node at distance ℓ can reach the sink with h hops, and $P_{path}(h)$ is the end to end resilience for a path with h hops.

In (4), $P_{path}(h)$ can be calculated following its definition as the probability that all nodes/links are uncaptured/uncompromised in the path with h hops. It is the multiplication of the probability that all h nodes on the path are uncaptured, the probability that the first $h-1$ hops between two nodes are uncompromised, and the probability that the last hop between a node and the sink is uncompromised. Then we have,

$$P_{path}(h) = \frac{\binom{N-h}{N_c}}{\binom{N}{N_c}} \cdot \left(P_{link}^{(1)}\right)^{h-1} \cdot P_{link}^{(2)}. \tag{5}$$

In the above, $P_{link}^{(1)}$ is the probability that a link between two uncaptured nodes is uncompromised, while $P_{link}^{(2)}$ is the probability that a link between an uncaptured node and the sink is uncompromised.

In (5), the value of $\frac{\binom{N-h}{N_c}}{\binom{N}{N_c}}$ clearly remains the same under our differentiated key pre-distribution. The value of $\left(P_{link}^{(1)}\right)^{h-1}$ is increased since the resilience of high resilient links, which are preferred for routing, is increased under differentiated key pre-distribution. Similarly, the value of $P_{link}^{(2)}$ is increased since the last hop node is more likely to be a node with large number of keys. Therefore, under differentiated key pre-distribution, the value of $P_{path}(h)$ increases, which leads to the increase of $P_{e2e}(\ell)$ in (4). In (4), the value of $P(h|\ell)$ tends to decrease for small h, and tends to increase for large h in our protocol. This is because in extending GPSR or min-hop protocol, we prefer high resilient links, which may not reside on the path with minimum hops. This will lead to the decrease of $P_{e2e}(\ell)$ in (4). However, such impact by $P(h|\ell)$ is dominated by that of $P_{path}(h)$ in general, as confirmed by extensive simulations in Section IV-E. Overall, the values of $P_{e2e}(\ell)$ and P_{e2e} increase under differentiated key pre-distribution, which shows the benefit of our methodology. The derivations of $P_{link}^{(1)}$, $P_{link}^{(2)}$ are given below. Due to space limitation, the derivations of $P(h|\ell)$ are skipped. In fact, our derivation of $P(h|\ell)$ is similar to the work in [51] and [52].

2) Derivation of q_i: Before we give the derivation for $P_{link}^{(1)}$ and $P_{link}^{(2)}$, we need to derive q_i, which is the probability that a node on the path between a sensor and the sink belongs to class i. The probability of a class i node appearing on the key path is different from the percentage of class i nodes in the network. This is because during our routing path selection, we prefer the nodes on the highly resilient links, and those nodes are more likely to be nodes with more keys pre-distributed. We will show the derivation of q_i in the following.

We first give some definitions. We define p_i as the percentage of class i nodes in the network, which is given by $p_i = n_i/N$. We define $P_{share}(i,j,\ell)$ as the probability that a class i node shares ℓ keys with a class j node, which can be given as,

$$P_{share}(i,j,\ell) = \frac{\binom{K}{\ell}\binom{K-\ell}{k_i-\ell}\binom{K-k_i}{k_j-\ell}}{\binom{K}{k_i}\binom{K}{k_j}}. \tag{6}$$

If we define $P_{prefer}(j,\ell)$ as the probability that a class j node is preferred to a class ℓ node during routing path selection, the expression of $P_{prefer}(j,\ell)$ can be given by,

$$\begin{aligned} P_{prefer}(j,\ell) \quad = \quad & \sum_{i=1}^{c} p_i \Big(\sum_{u=1}^{k_j} P_{share}(i,j,u) \\ & (\sum_{v=0}^{u-1} P_{share}(i,\ell,v)) \\ & + \frac{1}{2} \sum_{u=1}^{k_j} P_{share}(i,j,u) P_{share}(i,\ell,u) \Big). \end{aligned} \tag{7}$$

Finally, the expression of q_i is given by,

$$q_i = \binom{n_{nei} p_i}{1} \prod_{\ell=1}^{c} (P_{prefer}(i,\ell))^{n_{nei} p_\ell - f(\ell)}. \tag{8}$$

In (8), n_{nei} is the average number of physical neighbors of a node, which is given by $n_{nei} = N\frac{\pi r^2}{S}$. Besides, $f(\ell)$ equals 1 when $\ell = i$, and equals 0 otherwise.

3) Derivation of $P_{link}^{(1)}$ and $P_{link}^{(2)}$: Given the expressions of q_i derived above, we will derive the expressions for $P_{link}^{(1)}$ and $P_{link}^{(2)}$ in this section.

If we denote $P_{link}^{(1)}(i,j)$ as the resilience of the pairwise key between a class i node and a class j node, and denote $P_{link}^{(2)}(i)$ as the resilience of the pairwise key between a class i node and the sink, the expressions of $P_{link}^{(1)}$ and $P_{link}^{(2)}$ can be given by,

$$P_{link}^{(1)} = \sum_{i=1}^{c}\sum_{j=1}^{c} q_i q_j P_{link}^{(1)}(i,j), \tag{9}$$

$$P_{link}^{(2)} = \sum_{i=1}^{c} q_i P_{link}^{(2)}(i). \tag{10}$$

In the following, we will derive the expressions for $P_{link}^{(1)}(i,j)$ and $P_{link}^{(2)}(i)$.

We denote $P_{res}(i)$ as the probability that at least one of i unique pre-distributed keys is not disclosed to the attacker, denote $k_{avg}(i,j)$ as the average number of shared pre-distributed keys between a class i node and a class j node, denote $k_{avg}(i)$ as the average number of shared pre-distributed keys between a class i node and one of its physical neighbors, and denote $n_{path}(i,j)$ as the number of two-hop key paths between a class i node and a class j node. Thus, the expressions for $P_{link}^{(1)}(i,j)$ and $P_{link}^{(2)}(i)$ can be given by,

$$\begin{aligned} P_{link}^{(1)}(i,j) &= 1-\Big(1-P_{res}(k_{avg}(i,j))\Big) \\ &\quad \Big(1-(1-N_c/N)P_{res}(k_{avg}(i)) \\ &\quad P_{res}(k_{avg}(j))\Big)^{n_{path}(i,j)}, and \end{aligned} \tag{11}$$

$$P_{link}^{(2)}(i) = P_{res}(k_i). \tag{12}$$

In (11), $P_{res}(k_{avg}(i,j))$ is the probability that the direct one-hop key path between a class i node and a class j node is uncompromised, N_c/N is the probability that the proxy node on one of the two-hop key paths is captured, $P_{res}(k_{avg}(i))$ is the probability that the link on a two-hop key path between a class i node and the proxy is uncompromised, and $P_{res}(k_{avg}(j))$ is defined similarly. A two-hop key path is uncompromised if the proxy is uncaptured and both links are uncompromised. The pairwise key between a class i node and a class j node is uncompromised if at least one key path (either one-hop or two-hop key path) is uncompromised. In (12), the number of shared pre-distributed keys between a class i node and the sink is k_i since the sink is assumed to have all the K pre-distributed keys and will not be captured. In the following, we will derive the expressions for $P_{res}(i)$, $k_{avg}(i,j)$, $k_{avg}(i)$ and $n_{path}(i,j)$.

Given the number of captured nodes N_c, the average number of disclosed pre-distributed keys, denoted by K_{dis}, is given by,

$$K_{dis} = K(1-(1-\frac{k_{avg}}{K})^{N_c}), \tag{13}$$

where k_{avg} is the average number of keys pre-distributed in a node. The expression of k_{avg} is given by,

$$k_{avg} = \sum_{i=1}^{c} p_i k_i. \tag{14}$$

Given the expression of K_{dis} above, we are able to give the expression of $P_{res}(i)$ as,

$$P_{res}(i) = 1 - \frac{\binom{K-i}{K_{dis}-i}}{\binom{K}{K_{dis}}}. \tag{15}$$

Recall the expression of $P_{share}(i,j,\ell)$ in (6) above, the expressions of $k_{avg}(i,j)$ and $k_{avg}(i)$ can be given by,

$$k_{avg}(i,j) = \sum_{\ell=1}^{min\{k_i,k_j\}} \ell P_{share}(i,j,\ell), \tag{16}$$

$$k_{avg}(i) = \sum_{j=1}^{c} p_j k_{avg}(i,j). \tag{17}$$

Finally, the expression of $n_{path}(i,j)$ can be given by,

$$\begin{aligned} n_{path}(i,j) &= 0.5865 n_{nei} \sum_{\ell=1}^{c} p_\ell \\ &\quad (1-P_{share}(i,\ell,0))(1-P_{share}(j,\ell,0)), \end{aligned} \tag{18}$$

where $0.5865n_{nei}$ is the average number of common neighbors of two neighboring nodes [2].

Therefore, we have derived the expressions for $P_{link}^{(1)}$ and $P_{link}^{(2)}$ above.

Based on our derivation of P_{e2e}, we find that end to end security depends on several key pre-distribution parameters, that are, number of node classes (c), number of nodes in each class (c_i) and number of keys in class i nodes (k_i). The expression of P_{e2e} is clearly a nonlinear function of all the above parameters. Given the network parameters (R, r, N, N_c) and the memory constraint of sensors (maximum value of k), we can apply standard optimization tools on the above equations to obtain the optimal values for the design parameters.

The above analysis also has other usages. Consider a node i with geographic distance L to the sink. For any reference number δ $(0 < \delta < 1)$, we have,

$$P(\text{Resilience for the path from } i \text{ to sink} > \delta) \leq \frac{P_{e2e}(L)}{\delta},$$

where $P_{e2e}(L)$ is decided by (4). This bound follows directly by using Markov Inequality.

The above inequality illustrates a tradeoff between network size and end to end security performance. When network size (L) increases, $P_{e2e}(L)$ will decrease. Thus the upperbound value in the right-hand side will decrease. Given a required end to end resilience for any node in the network, there exists an upperbound for the network size, beyond which the resilience requirement cannot be achieved. Therefore, in a large scale network, we can deploy nodes in groups, each of which has a sink. In this way, we effectively decrease network size in each group, and subsequently can achieve the required end to end security performance.

D. Remarks

In the following, we will discuss the issues of empty set $U(i)$ (discussed in Section IV-B), possibility of longer hops, extending our solutions to hierarchical networks, and the possibility of applying public key cryptography.

In extending GPSR, a node i may find out that its set $U(i)$ is empty. In such case, node i can follow the right hand rule in [45] to choose a secure neighbor that is further away from the sink than i itself. If node i happens not to have any secure neighbor, it may increase its communication range, hoping to find some secure neighbors at the cost of power consumption. Applying such rules will eliminate loops and guarantee finding a secure path if it exists. Increasing communication range for more secure neighbors works for the extended minimum hop protocol as well.

We would like to point out that the number of path hops in our schemes could be larger than that in traditional GPSR or minimum hop routing schemes. This is because in our schemes, nodes choose neighbors considering both path length and link resilience, and thus could choose neighbors on a path with more hops. Besides, as mentioned above, a node may choose a secure neighbor that is further away from the sink than itself. Intuitively, a path with more hops tends to decrease path resilience as the chance of attacker compromising at least one hop is increased. However, in our schemes, the path resilience improvement via choosing highly resilient links overwhelms the negative effect of a little longer paths of a small percentage of nodes. Overall, the path resilience will be improved.

In this paper, we have focused on flat topologies. In some situations nodes could be deployed in hierarchies. The end to end routing here occurs in more than one plane, i.e., sensor to cluster head via multiple sensors, and cluster head to sink via multiple cluster heads. Our methodology and protocols are directly applicable in such hierarchical networks as well. The cluster heads can be chosen as class 1 nodes (provisioned with more keys), while other sensors can be chosen as class 2 nodes, class 3 nodes and so on depending on the number of levels in the hierarchy.

Recently, public key cryptography has been receiving attention in WSNs [53] [54] [55] [56], which can be used to establish pairwise keys between neighboring sensor nodes. However, public key cryptography based scheme involves high energy consumption due to its complicated computation, which may not be preferable for energy constrained sensor nodes. Based on experiments in [54], [55] and [56], energy consumption in public key cryptography based pairwise key establishment is about two to three orders of magnitude more than that in the symmetric key cryptography based scheme. In [53], special hardware is used to reduce the energy consumption of public key based operation, which clearly incurs extra cost. Considering that public key cryptography based pairwise key establishment is still too energy consuming for long lived operation of energy constrained sensors, random key pre-distribution based pairwise key establishment is still highly relevant and practical for large scale deployment of secure wireless sensor networks. Furthermore, some researchers in the sensor networks area are building their own customized sensor-mote prototypes [57], with lower memory, processing power, and battery life than commercial platforms. For such prototypes also, symmetric key cryptography techniques will have practical relevance.

E. Performance Evaluation

In this section, we present performance evaluation based on both analysis and simulation. The analysis is based on our discussions in Section IV-C. We first describe our simulation setup, and then report performance data and our observations.

1) Simulation Setup: We conduct our simulation using a self-made simulator in C. The network is circular with radius $500\ meters$, where 1000 nodes are uniformly deployed at random. The sink is at the center of the network. Unless otherwise specified, the default parameters are: $c = 2$, $n_1 = 200$, $n_2 = 800$, $k_1 = 80$, $k_2 = 30$, $k = 40$, $K = 10000$, $r = 100\ meters$, $\alpha = 1$ and $N_c = 50$ (for notation, please refer to Table 2 in Section IV). The default values of k_1, k_2 and k are chosen such that $k_1 n_1/(n_1+n_2)+k_2 n_2/(n_1+n_2) = k$, which means the average number of keys disclosed to the attacker is the same in our differentiated key pre-distribution and the original RKP scheme for the same number of captured nodes. Our communication model is one where sensors periodically transmit data to the sink. In the legend in all figures, *our GPSR* and *our minhop* refer to our protocols extending GPSR [45] and minimum hop [50] routing presented in Section IV-B respectively. The legends *GPSR* and *minhop* refer to the traditional GPSR and minimum hop routing protocols following the uniform key pre-distribution respectively. Each point in the simulation data is the average of 100 runs based on independent random seeds.

2) Sensitivity of P_{e2e} to Attack Intensity: In Fig. 2, we first compare our differentiated key pre-distribution with the traditional uniform key pre-distribution (for both GPSR and minimum hop routing protocols) under different number of captured nodes N_c. We find that while the performance of all schemes degrades with increasing N_c, our schemes are consistently better than those of traditional schemes. We also find that the improvement increases with larger values of N_c. This is because when the attacker captures more nodes, the resilience of highly resilient links in our schemes degrades at a much slower pace than those of the less resilient links in traditional schemes. Besides, we can also observe that the end to end security under minimum hop based protocols is better than their GPSR counterparts. This is because minimum hop based protocols always choose the path with minimum hops, while the GPSR based protocols may choose longer paths, which compromises end to end resilience. The cost though is the increased initial energy consumption in query flooding.

3) Sensitivity of P_{e2e} to Network Density: In Fig. 3, we compare our schemes and traditional schemes under different communication range r, which in turn corresponds to different network density (i.e., number of neighbors per node). When r is small, P_{e2e} is low due to both low connectivity (many nodes cannot find secure neighbors) and low resilience (fewer proxies resulting in fewer key paths for each link). When r increases, P_{e2e} increases correspondingly. For all values of r, our schemes performs consistently better.

4) Sensitivity of network lifetime to parameter α: Recall from Section IV-B that α is the *knob* that trades-off security with lifetime. In Fig. 4, we compare our schemes and the traditional schemes for different values of α. We define network lifetime as the time until when the first node has

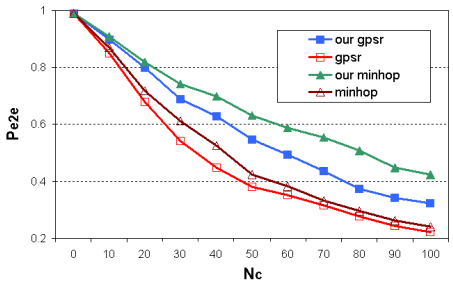

Fig. 2. Sensitivity of P_{e2e} to number of captured nodes N_c.

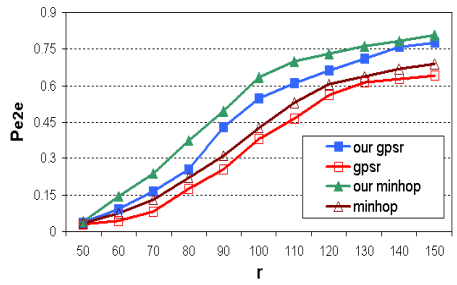

Fig. 3. Sensitivity of P_{e2e} to communication range r.

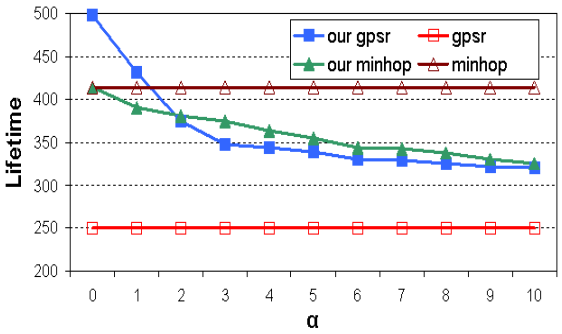

Fig. 4. Sensitivity of lifetime to parameter α.

used up its energy. Since traditional schemes do not have weight assignment, they are insensitive to α. The lifetime in our schemes decreases with larger values of α. This is because a larger value of α means more priority is given to links with high resilience, thereby draining the corresponding neighbors more rapidly.

We also observe that the extended GPSR has higher lifetime compared with extended minimum hop for smaller values of α, and the difference diminishes as α increases. This is because for smaller values of α, lifetime is mainly decided by total number of candidate forwarders of each node. In extended GPSR, each node usually can find more forwarders (secure neighbors closer to sink) than it can find in extended minimum hop protocol (secure neighbors on minimum hop secure path). When α increases, lifetime is mainly decided by the number of most secure neighbors of each node. This number is similar for both protocols, and hence they have similar lifetimes when α increases. We also observe that lifetime of traditional GPSR scheme is lower than that of traditional minimum hop scheme. This is because in traditional GPSR scheme, some nodes are so positioned that most of their nearby nodes will choose them as forwarders, which results in their energy being drained quickly. While in traditional minimum hop scheme, nodes are less likely to be the only one on the minimum hop path of most of their neighbors, and thus traffic is more balanced.

5) Sensitivity of P_{e2e} and network lifetime to number of class 1 nodes: In Figs. 5 and 6, we compare the traditional schemes, our schemes with default parameters, and our schemes with optimal parameters. The optimal parameters are obtained via our analysis in Section IV-C. The average number of keys pre-distributed per node is the same across all schemes for fairness of comparison. In Fig. 5, we find that traditional schemes are insensitive to n_1 since all nodes are given same number of keys. Our schemes achieve much better performance under intermediate values of n_1, while the performance of our schemes is close to that of traditional schemes for very small and very large values of n_1. This is because when n_1 approaches 0 or 1000, all nodes will be given same number of keys, and thus our schemes degrade to traditional schemes. In Fig. 6, we also observe that lifetime of the traditional schemes is insensitive to n_1 due to the same reason as above. The lifetime of our schemes increases with the value of n_1. The case when $n_1 = 0$ can be treated as the same as $n_1 = 1000$. This is because for small values of n_1, the class 1 nodes are given many keys initially, and so they tend to be used as forwarders much more frequently and the lifetime tends to be small. When n_1 increases, the number of keys given to class 1 nodes decreases, thus helping to distribute the load more evenly and improve network lifetime. Finally, in both figures, we observe that the optimal parameters derived by our analysis give better performance than the default parameters, which confirms the correctness of our analysis.

V. Biased Node Capturing Attack and its Countermeasures

Recall that in the traditional attack model discussed in Section II-A, the attacker is able to capture a certain percent of the nodes in the network. Such an attack is an unbiased one since the captured nodes are chosen at random. In this section, we will study a type of advanced attack model, denoted as *biased node capturing attack*, in which the attacker has bias in choosing nodes to capture, aiming to achieve higher attack impact.

A. Biased Node Capturing Attack

Simply put, biased node capturing attack is one in which the attacker attempts to capture some special nodes in the network. Typically, the capture of those nodes results in higher attack impact, and they are chosen with bias instead of randomly. The existence of such special nodes comes from the fact that the roles (or importance) of sensor nodes in the network are inherently different. In a multi-hop sensor network, the nodes near the sink are such special nodes, whose capture results

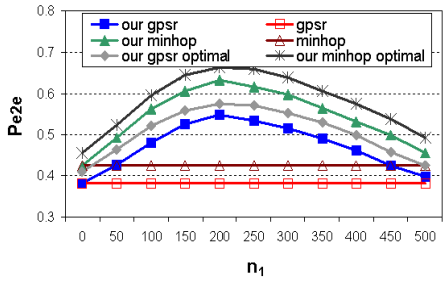

Fig. 5. Sensitivity of P_{e2e} to number of class 1 nodes n_1.

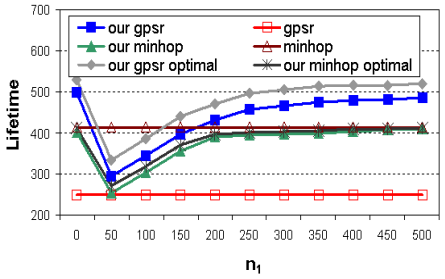

Fig. 6. Sensitivity of lifetime to number of class 1 nodes n_1.

in more secret information disclosed to the attacker. This is because a node near the sink generally forwards more traffic, and its capture results in more data being disclosed. Thus, the attacker can take advantage of the heterogeneity in topology to capture those important nodes near the sink.

In our differentiated key management, we also introduce a type of heterogeneity among the nodes in that different nodes have different number of pre-distributed keys. The capture of nodes with more pre-distributed keys tends to have higher attack impact due to the fact that more pre-distributed keys are disclosed. Thus, the attacker could also take advantage of the heterogeneity in the number of pre-distributed keys to achieve higher attack impact. Such an attack can be easily accomplished via identifying more resilient links (and hence nodes) via simple traffic analysis of monitored communication.

In Fig. 7, we show the impacts of three types of biased attacks and the unbiased one. The first biased attack is one that chooses to capture nodes closest to the sink, denoted as *topology* in Fig. 7. The second is one that chooses to capture nodes with the largest number of pre-distributed keys, denoted as *key* in Fig. 7. The third combines both strategies, in which the attacker first chooses nodes closest to the sink. When multiple nodes are at the same distance to the sink, tie is broken based on the number of keys pre-distributed. Such a combined attack is denoted as *topology+key* in Fig. 7.

In Fig. 7, we find that the biased attacks result in higher attack impact than the unbiased one. However, the attack impact caused by the biased attack based on topology alone is much more severe than that caused by the biased attack based on key alone. Besides, the impact of the combined attack is close to that caused by the biased attack based on topology alone. This is because the nodes close to the sink are generally those forwarding more traffic. The capture of a few such nodes results in a significant portion of the data being disclosed. On the other side, when nodes with more pre-distributed keys are captured, the overall number of pre-distributed keys disclosed is still a small portion of the key pool size. The above observations show that the attack impact caused by topology heterogeneity dominates that caused by key heterogeneity. Combining key heterogeneity with topology heterogeneity does not further degrade security much. Therefore, our differentiated key management technique does not introduce a lot of additional negative impacts when the biased attack based on network topology is already in place.

B. *Countermeasures*

In the following, we will discuss the countermeasures to the biased attacks discussed above.

Note that the biased attack based on topology naturally exists in multi-hop sensor networks, and thus is not introduced by our differentiated key management. Here we discuss its countermeasures briefly. One potential countermeasure is letting the nodes near the sink just forward the encrypted data without needing to decrypt it for aggregation. In this way, the capture of such nodes does not disclose the data forwarded. To do so, nodes with a certain number of hops away from the sink need to be pre-distributed with a unique pairwise key with the sink before node deployment. Such approach comes at the cost that no data aggregation is conducted near the sink. An alternative countermeasure is letting the sink node move around the network so the amount of forwarded traffic is balanced among the nodes in the network. Thus, the impact of such biased attack is alleviated.

As shown in Fig. 7 above, the biased attack based on the number of pre-distributed keys does cause higher attack impact although such impact is far less than that of the biased attack based on topology. One potential countermeasure is to use tamper resistant hardware for the nodes pre-distributed with more keys. Therefore, such nodes become more robust to the attack in that the attacker may not be able to obtain secret information in the captured node. Such idea is inspired by the work in [9] where some special nodes are assumed never to disclose their secret information after capture. Here, we relax such assumption and allow a certain probability of secret information in such nodes being disclosed. Such probability is denoted as P_{dis}.

In Fig. 8, we show the sensitivity of P_{e2e} to the value of P_{dis}, the probability that nodes with tamper resistant hardware having their secret information disclosed after being captured. All three curves in Fig. 8 are based on GPSR routing. Similar observations are made in minimum hop routing as well. We find that, when the tamper resistant hardware can achieve a disclosure probability less than 0.5, the security performance of our differentiated key management under biased attack based on the number of pre-distributed keys is better than that under unbiased attack. When P_{dis} is between 0.5 and 0.8, the performance under biased attack falls below that under

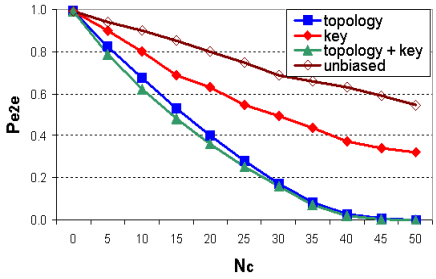

Fig. 7. Sensitivity of P_{e2e} to the strategy of biased attack.

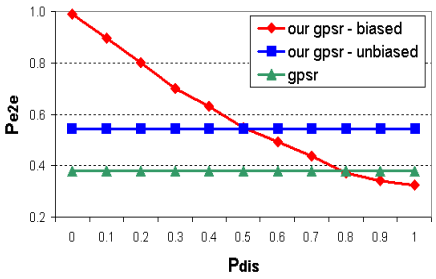

Fig. 8. Sensitivity of P_{e2e} to disclosure probability P_{dis}.

unbiased attack since more pre-distributed keys are disclosed. However, it is still better than that in traditional uniform key management. Only when P_{dis} becomes larger than 0.8 (ineffective tamper resistant hardware), the performance of differentiated key management under biased attack is worse than that in traditional uniform key management. In summary, with reasonably effective tamper resistant hardware ($P_{dis} < 0.5$), the performance of our differentiated key management will not degrade under biased attack. Such improvement comes at the cost of installing tamper resistant hardware. However, we only need to provide tamper resistant hardware for a small portion of the nodes, which is worthwhile considering the security performance improvement.

An alternative countermeasure to the biased attack based on the number of pre-distributed keys is one where camouflaging techniques are applied. Such camouflaging techniques aim to hide those nodes with more pre-distributed keys. During pairwise key establishment, each node i sends k_1 key IDs to its neighbors. Recall k_1 is maximum among all $k_i's$. If the number of keys distributed in node i is smaller than k_1, node i will append random dummy key IDs to ensure a homogeneous key ID list size. During the routing path selection and later communications, important nodes are more likely to receive/send more packets due to the high resilience of its links. Our alternative path routing can help alleviate this problem when traffic forwarding is shared by multiple neighbors. Besides, we can let nodes with low traffic burden send dummy traffic to maintain a homogeneous communication burden among all nodes. Based on the above camouflaging techniques, all nodes in our protocol tend to behave homogeneously. Therefore, the attacker cannot distinguish nodes with more keys from others. This approach comes at the cost of extra communication overhead.

VI. Conclusion

In this paper, we address the issue of providing end to end secure communications in randomly deployed wireless sensor networks. To the best of our knowledge, this is the first paper to study this topic. Our methodology is *differentiated key pre-distribution*, where the core idea is to distribute different number of keys to different sensors to enhance the resilience of certain links in the network. This feature is leveraged during routing, where nodes route through those links with higher resilience. We then present our end to end secure communication protocol based on the above methodology by extending well known location centric (GPSR) and data centric (minimum hop) routing protocols.

Using rigorous theoretical analysis, we derive the formulas to determine quality of end to end secure communications. We then use the analysis to determine optimum values for design parameters. Using extensive analytical data and simulations, we demonstrate the effectiveness of our solution in providing end to end secure communications, and prolonging network lifetime. We also provide a detailed discussion on biased node capturing attack to our solutions, and propose several countermeasures.

References

[1] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS)*, November 2002.

[2] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of IEEE Symposium on Research in Security and Privacy*, May 2003.

[3] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key predistribution scheme for wireless sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)*, October 2003.

[4] J. Lee and D. R. Stinson, "Deterministic key predistribution schemes for distributed sensor networks," in *Proceedings of the 11th workshop on Selected Areas in Cryptography (SAC)*, August 2004.

[5] J. Lee and D. R. Stinson, "A combinatorial approach to key predistribution for distributed sensor networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, March 2005.

[6] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)*, October 2003.

[7] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach," in *Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP)*, November 2003.

[8] H. Dai and H. Xu, "Triangle-based key management scheme for wireless sensor networks," *Frontiers of Electrical and Electronic Engineering in China*, vol. 4, no. 3, pp. 300–306, 2009.

[9] P. Traynor, H. Choi, G. Cao, S. Zhu, and T. L. Porta, "Establishing pair-wise keys in heterogeneous sensor networks," in *Proceedings of the 25th IEEE Conference on Computer Communications (INFOCOM)*, April 2006.

[10] A. Poornima and B. Amberker, "Tree-based key management scheme for heterogeneous sensor networks," in *16th IEEE International Conference on Networks (ICON)*, 2008.

[11] Y. Zhang, W. Yang, K. Kim, and M. Park, "An AVL Tree-Based Dynamic Key Management in Hierarchical Wireless Sensor Network," in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP)*, pp. 298–303, 2008.

[12] T. Landstra, M. Zawodniok, and S. Jagannathan, "Energy-efficient hybrid key management protocol for wireless sensor networks," in *IEEE Conference on Local Computer Networks (LCN)*, 2007.

[13] A. Poornima and B. Amberker, "Key Management Schemes for Secure Communication in Heterogeneous Sensor Networks," in *International Journal of Recent Trends in Engineering*, 2009.

[14] A. Das, "An unconditionally secure key management scheme for large-scale heterogeneous wireless sensor networks," in *First International Communication Systems and Networks and Workshops (COMSNETS)*, pp. 1–10, 2009.

[15] Y. Yang, J. Zhou, R. Deng, and F. Bao, "Hierarchical Self-healing Key Distribution for Heterogeneous Wireless Sensor Networks," *Security and Privacy in Communication Networks*, pp. 285–295, 2009.

[16] B. Tian, S. Han, and T. Dillon, "A Key Management Scheme for Heterogeneous Sensor Networks Using Keyed-Hash Chain ," in *5th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, 2010.

[17] J. Kim, J. Lee, , and K. Rim, "Energy Efficient Key Management Protocol in Wireless Sensor Networks," in *International Journal of Security and Its Applications*, 2010.

[18] M. Wen, Z. Yin, Y. Long, and Y. Wang, "An Adaptive Key Management Framework for the Wireless Mesh and Sensor Networks," *Wireless Sensor Network Journal*, 2010.

[19] H. Jen-Yan, I. Liao, and H. Tang, "A Forward Authentication Key Management Scheme for Heterogeneous Sensor Networks," *EURASIP Journal on Wireless Communications and Networking*, 2010.

[20] D. Liu, P. Ning, and W. Du, "Group-based key predistribution for wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 2, pp. 1–30, 2008.

[21] N. Canh, P. Truc, T. Hai, Y. Lee, and S. Lee, "Enhanced group-based key management scheme for wireless sensor networks using deployment knowledge," in *6th IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 1–5, 2009.

[22] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom)*, 2004.

[23] D. Liu and P. Ning, "Improving key predistribution with deployment knowledge in static sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 1, no. 2, pp. 204–239, 2005.

[24] N. Canh, Y. Lee, and S. Lee, "HGKM: A group-based key management scheme for sensor networks using deployment knowledge," in *6th Annual Communication Networks and Services Research Conference (CNSR)*, pp. 544–551, 2008.

[25] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 247–260, 2006.

[26] Y. Zhang, W. Liu, Y. Fang, and D. Wu, "Secure localization and authentication in ultra-wideband sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 4, pp. 829–835, 2006.

[27] K. Ren, W. Lou, and Y. Zhang, "LEDS: Providing location-aware end-to-end data security in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 585–598, May 2008.

[28] S. Patil, "Sensor Network Traffic-Adaptive Key Management Scheme," in *International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom)*, pp. 610–614, 2009.

[29] Y. Qian, K. Lu, B. Rong, and D. Tipper, "A design of optimal key management scheme for secure and survivable wireless sensor networks," *Security and Communication Networks*, vol. 1, no. 1, pp. 75–85, 2008.

[30] C. Wang, T. Hong, G. Horng, and W. Wang, "A ga-based key-management scheme in hierarchical wireless sensor networks," in *International Journal of Innovative Computing, Information and Control*, 2008.

[31] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, March 2004.

[32] S. Chellappan, W.Gu, X. Bai, B. Ma, D. Xuan, and K.Zhang, "Deploying wireless sensor networks under limited mobility constraints," in *IEEE Transactions on Mobile Computing (TMC), Vol. 6, No. 10*, October 2007.

[33] P. Andreou, D. Zeinalipour-Yazti, P. Chrysanthis, and G. Samaras, "In-network data acquisition and replication in mobile sensor networks," *Distributed and Parallel Databases*, pp. 1–26, 2011.

[34] C. Wang, P. Ramanathan, and K. Saluja, "Modeling latencylifetime trade-off for target detection in mobile sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 7, no. 1, pp. 1–24, 2010.

[35] S. Choi, V. Sarangan, and S. Trost, "Key management in wireless sensor networks with inter-network sensor roaming," in *33rd IEEE Conference on Local Computer Networks (LCN)*, 2008.

[36] Y. Lee and S. Lee, "A New Efficient Key Management Protocol for Wireless Sensor and Actor Networks," *Arxiv preprint arXiv:0912.0580*, 2009.

[37] K. Kifayat, M. Merabti, Q. Shi, and D. Llewellyn-Jones, "Group-based key management for Mobile Sensor Networks," in *IEEE Sarnoff Symposium*, pp. 1–5, 2010.

[38] H. Alzaid, D. Park, J. Nieto, C. Boyd, and E. Foo, "A forward and backward secure key management in wireless sensor networks for PCS/SCADA," *Sensor Systems and Software*, pp. 66–82, 2010.

[39] R. Silva, N. Pereira, and M. Nunes, "Probabilistic key management practical concerns in wireless sensor networks," *Journal of Networks*, vol. 3, no. 2, 2008.

[40] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *The Second Annual International Conference on Mobile and Ubiquitous Systems Networking and Services (MobiQuitous)*, pp. 109–117, 2005.

[41] B. Dutertre, S. Cheung, and J. Levy, "Lightweight key management in wireless sensor networks by leveraging initial trust," *SRI International, SDL Technical Report SRI-SDL-04-02*, 2004.

[42] Z. Sencun, S. Sanjeev, and J. Sushi, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," in *Proceedings of ACM Conference on Computing and Communication Security (CCS)*, pp. 62–72, 2003.

[43] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, vol. 1, pp. 293–315, September 2003.

[44] J. Spencer, "The strange logic of random graphs," 2001.

[45] B. Karp and H. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MOBICOM)*, August 2000.

[46] C. Intanagonviwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MOBICOM)*, August 2000.

[47] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks (leach)," in *Proceedings of the 33rd Hawaii International Conference on Systems Science (HICSS)*, January 2000.

[48] S. Tanachaiwiwat, P. Dave, R. Bhindwale, and A. Helmy, "Secure locations: Routing on trust and isolating compromised sensors in location-aware sensor networks," in *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (Sensys)*, November 2003.

[49] A. D. Wood, L. Fang, J. A. Stankovic, and T. He, "Sigf: A family of configurable, secure routing protocols for wireless sensor networks," in *Proceedings of the 4th ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, October 2006.

[50] C. Schugers and M. Srivastava, "Energy efficient routing in wireless sensor networks," in *Proceedings of Milcom*, October 2001.

[51] S. A. G. Chandler, "Calculation of number of relay hops required in randomly located radio network," *Electronics Letters*, vol. 25, pp. 1669–1671, November 1989.

[52] Y. C. Cheng and T. G. Robertazzi, "Critical connectivity phenomena in multihop radio models," *IEEE Transactions on Communications*, vol. 37, pp. 770–777, July 1989.

[53] L. Batina, N. Mentens, K. Sakiyama, B. Preneel, and I. Verbauwhede, "Low-cost elliptic curve cryptography for wireless sensor networks," in *Proceedings of the 3rd European Workshop on Security in Ad-hoc and Sensor Networks (ESAS)*, September 2006.

[54] A. Liu, P. Kampanakis, and P. Ning, "Tinyecc: Elliptic curve cryptography for sensor networks (version 0.3)," in *http://discovery.csc.ncsu.edu/software/TinyECC/*, February 2007.

[55] K. Piotrowski, P. Langendoerfer, and S. Peter, "How public key cryptography influences wireless sensor node lifetime," in *Proceedings of the 4th ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, October 2006.

[56] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "Tinypk: Securing sensor networks with public key technology," in *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (Sensys)*, October 2004.

[57] "https://sites.google.com/a/mst.edu/missouri-snt-motes/home,"