2007

Optimal Layout of Multicast Groups Using Network Embedded Multicast Security in Ad Hoc Sensor Networks

Richard R. Brooks

Brijesh Pillai

Michele C. Weigle
Old Dominion University

Matthew Pirretti

Original Publication Citation

Brooks, R.R., Pillai, B., Weigle, M.C., & Pirretti, M. (2007). Optimal layout of multicast groups using network embedded multicast security in ad hoc sensor networks. *International Journal of Distributed Sensor Networks, 3*(3), 273-287. doi: 10.1080/15501320601062080

International Journal of Distributed Sensor Networks, 3: 273–287, 2007

ISSN: 1550-1329 print/1550-1477 online
DOI: 10.1080/15501320601062080

Taylor & Francis
Taylor & Francis Group

Optimal Layout of Multicast Groups Using Network Embedded Multicast Security in *Ad Hoc* Sensor Networks

R. R. BROOKS and BRIJESH PILLAI

Holcombe Department of Electrical and Computer Engineering

MICHELE C. WEIGLE

Department of Computer Science, Clemson University, Clemson, SC

MATTHEW PIRRETTI

Computer Science and Engineering Department, The Pennsylvania State University, University Park, PA

*This paper considers the security of sensor network applications. Our approach creates multicast regions that use symmetric key cryptography for communications. Each multicast region contains a single keyserver that is used to perform key management and maintain the integrity of a multicast region. Communications between two multicast regions is performed by nodes that belong to both regions. To ease the network management burden, it is desirable for the networks to self-organize into regions and dynamically select their keyservers. This paper shows how to determine the number of keyservers (*k*) to use and the size in the number of hops (*h*) of their multicast regions. We find that power consumption issues provide a natural trade-off that determines optimal values for these parameters. Analysis of one application shows an increase in system security with 70–80% less power overhead than existing security approaches.*

Keywords Sensor Networks; Key Management; *Ad Hoc* Networks

1. Introduction

Sensor nodes work in a distributed and cooperative manner to increase the lifetime of the network and maintain their sensing capability. Sensor networks rely on limited, non-renewable battery energy resources, so all aspects of their operation need to be as energy-efficient as possible. Though prior work [3, 4] claims that wireless communications dominate energy consumption in sensor networks, many sensor network applications [8, 9, 10, 11] have communications responsible for less than 20% of the total energy drain. Other researchers [19, 20, 21] have shown that encryption, decryption, and secure hashing are computation-intensive with a large energy overhead. AES encryption per bit of information consumes one-sixth the energy required to transmit a single bit using Bluetooth [19].

Address correspondence to R. R. Brooks, Holcombe Department of Electrical and Computer Engineering, Clemson University, PO Box 340915, Clemson, SC 29634-0915, Tel.: 864-656-0920, Fax: 864-656-5910. E-mail: rrb@acm.org

In this paper, we propose a new approach to sensor network security that uses multicast regions to manage cryptographic keys. We illustrate how this approach can greatly reduce the number of encryptions needed by the system to secure communications resulting in significant (~90%) reduction in the power budget needed to support security.

Our work builds on a network viability criterion for network connectivity and sensing coverage [22] that is a direct consequence of the network model in [12], where nodes with a fixed communications range are placed at random in the terrain. To be viable, a sensor network needs to have sufficient connectivity to guarantee that a unique component connects a quorum of nodes. It also needs enough connected nodes to be positioned so that they can detect events throughout the sensor field. These viability factors are satisfied if and only if the network possesses a unique giant component. The network is said to have a giant component if there exists a component whose size is on the order of the total number of nodes in the network. Absence of a giant component would result in failure of the viability criterion. The sensor network will not be able to detect every event and/or inform the user community.

A contribution of this paper is the use of multicast communications to secure sensor networks. A node transmits messages securely within a local multicast group by encrypting the message using a shared symmetric key. Each member of the multicast group reads the message by decrypting it locally. A packet is re-encrypted only when moving between different multicast regions. When data is shared within regions, approaches using multicast communication require fewer encryptions for secure message exchange, resulting in a net power savings. Each multicast region has a single keyserver that manages key distribution within the region. The tradeoff between the number of multicast regions and the size of each region becomes vital to maintaining minimum message transmission overhead and reducing power consumption due to computation, while at the same time ensuring security.

Another contribution of this paper is the development of a methodology that allows a sensor network to self-organize into secure multicast regions. The distribution of keyservers needs to be defined so as to maintain the giant component needed for system viability without incurring excessive overhead. We show how to find the number of keyservers (k) the network needs and the size in hops (h) of the multicast region they should manage. A larger overview of this multicast sensor network security scheme is available in [2].

The rest of this paper is organized as follows. Section 2 covers the criterion for network viability. It presents equations that predict phase changes in *ad hoc* networks. In Section 3, we combine results from Section 2 with the network maintenance protocol in [2] to find the keyserver distribution that minimizes system overhead. Section 4 explains how our approach can be integrated with the ColTraNe application described in [11]. Section 4 also shows how our approach results in fewer encryptions, less message traffic and lower power consumption when compared to current techniques. We conclude in Section 5 with future directions for research.

2. Predicting Phase Change in *Ad Hoc* Networks

Ad hoc networks with range-limited communications exhibit phase change phenomena like those found in random graph [13] and percolation [14] theories. Random graph theory is a branch of graph theory that assigns probability distributions to the existence of edges between vertices. Percolation theory, a branch of physics, studies fluid flows in random media. In these models, network behavior has two phases. In the first phase, the probability of connection between nodes is small and the network has a large number of isolated components. As the connection probability grows, the expected size of the largest component

grows logarithmically. In the second phase, the network is dominated by a unique giant component that contains most of the system nodes. There are still isolated holes in the network. The size of the largest hole shrinks logarithmically as the connection probability increases. The transition between these two phases, called the phase change, is extremely steep.

For random graphs, the curve of the maximum component size versus the edge probability takes the form $e^{-e^{a}}$. Above the phase change (percolation threshold) a single giant component of O(n) connects most of the sensor nodes [14], with at least one path connecting all the terrain's external boundaries. This property is self-similar; i.e., it is true for the system across all scales. Thus, for a surveillance sensor network with a giant component, targets of interest traversing the network will be detected by at least one node that can report the detection to the user community. System self-similarity implies that a target traversing any portion of the network is almost certain to be detected and reported to the user community.

This shows that the network is viable while it has a giant component. When there is no path between the terrain's external boundaries, the giant component is fractured into a set of isolated regions. For most observations, the network will not be able to report results to its user community. A fuller treatment of these issues is in [15].

It is vital for any application to predict this phase transition since it defines the viability of the network. Consider sensor networks with nodes either randomly placed [12], in a regular tessellation [14], or a weighted combination of the two. In this paper, we use the random graph models shown in Fig. 1 to model sensor networks.

Sensor nodes are vertices in a random graph structure. Edges between vertices represent either the likelihood of an active communications link, or detection of a target passing between nodes. In practice the edge probability distribution is the minimum of the two likelihoods, which is often the communications range. Nodes are placed at random in a geographic region for networks that fit the range-limited graph model in [12]. We integrate this approach with random key predistribution concepts from [24], so that an edge exists between two nodes only if they are within communications range of each other and share a common key. Instead of formally decomposing the graph definition into a set of Bernoulli probabilities to model the random process, we work using the tools of statistical physics [14] to derive a model that approximates system behavior [16]. Formally, we model the sensor network as a random graph $G = (V, E)$. The set of vertices V corresponds to the set of sensor nodes, and the elements of the set of edges E are communications links between the sensor nodes.

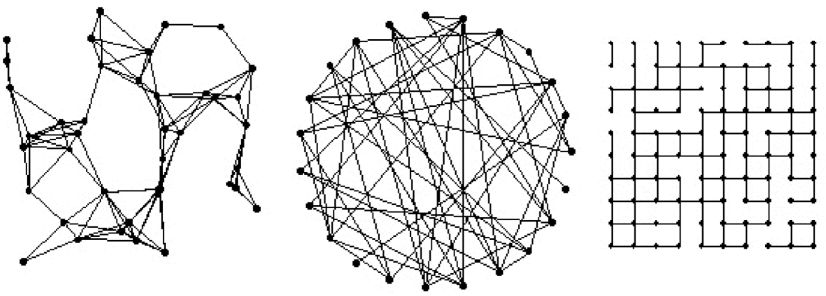

FIGURE 1 Graphs based on (left to right) range-limited model, Erdös-Rényi model and percolation theory (regular tessellation of nodes).

To analyze the graph, we use a probabilistic connectivity matrix M_h where each element *(i,j)* is the probability of a connection between nodes *i* and *j* in *h* hops. For this paper we assume the graph is undirected. The communication radius is denoted by *r*. A node can establish a connection with another node in a single hop only if both nodes fall within this communication radius. We normalize the value of *r* from [0 . . . 1]. For range-limited graphs, element (*i,j*) of the probabilistic connectivity matrix (probability of an edge connecting node *j* to node *i*) has the following value (see [16] for derivation):

$$(2c - c^2) \tag{1}$$

where *c* is a constant defined as:

$$c = \begin{cases} r^2 - \left(\frac{i}{n+1} - \frac{j}{n+1}\right)^2 & ;r^2 \geq \left(\frac{i}{n+1} - \frac{j}{n+1}\right)^2 \\ 0 & ;otherwise \end{cases}$$

The sensor network is only viable above the phase change where the network has a giant component. Before the phase change, the distribution of component sizes is such that most nodes are isolated and a small number of components of size up to O(log *n*) exist [13]. The magnitude of elements in the probabilistic connectivity matrix M_h, decrease as *h* increases. After the phase change, the number of isolated nodes and small components decreases dramatically. The single giant component of size O(*n*) emerges. Given the distribution of component sizes, on average, before (after) the phase change the number of nodes reachable within *h* hops will decrease (increase) with *h*. The likelihood of two nodes communicating with each other in *h* hops or less changes accordingly. This implies that the phase change should occur when $p_{ij}^{(h)} = p_{ij}^{(h+1)}$, *i.e.*, there exists an equal likelihood of a path between two nodes in *h* walks and a path between the same two nodes in *h + 1* walks.

Equation (2) looks for paths from node *i* to node *j* by considering paths passing through all possible intermediate nodes. Constraining diagonals in the connectivity matrix to zero removes consideration of a node as its own intermediate node.

$$p_{ij}^{(2)} = 1 - \prod_{\substack{l=1 \\ l \neq i \\ l \neq j}}^{n} (1 - p_{il}^{(1)} * p_{lj}^{(1)}) \tag{2}$$

where $p_{ij}^{(2)}$ is the probability a walk of two hops exists edge between nodes *i* and *j*;

$p_{ij}^{(1)}$ is the probability an edge exists between nodes *i* and *l*;

$p_{ij}^{(1)}$ is the probability an edge exists between nodes *l* and *j*.

Edge effects are an artifact of our model observed among nodes in the boundary of the field. To avoid edge effects, we consider nodes $i = \left\lceil \frac{n}{2} \right\rceil$ and $j = \left\lceil \frac{n}{2} \right\rceil + 1$ to find the phase change. Full derivations of these results are in [15] and [16].

We use the architecture described in [2] for sensor network organization. In this architecture, security is maintained by authenticating nodes when they join the network. Some sensor nodes are elected to be keyservers by using a secure election scheme. The local keyserver establishes session keys and manages group communication within the multicast group. Each of the k keyservers forms a multicast region by soliciting the membership of all nodes within h hops. Nodes served by more than one keyserver act as gateways between multicast regions. Direct communications is therefore possible between any two adjacent multicast regions when their keyservers are separated by $2h$-1 or fewer hops. The network of secure multicast regions should form a secure giant component overlaying the physical range-limited giant component to form a viable network of sensor nodes. Figure 2 shows a multicast communication topology. The large circles indicate partitioned multicast regions.

Erdös-Rényi [23] defined a graph topology where there is an equal probability an edge exists between any two vertices. In [2], we provide theorems that map the connectivity graph for multicast regions to an Erdös-Rényi topology. The likelihood a path of $2h$-1 hops exists between any two nodes chosen at random on the range-limited graph will be the same. We can therefore consider the keyserver connectivity graph as an Erdös-Rényi graph of k nodes, where k is the number of keyservers. This network of keyservers is modeled as an Erdös-Rényi graph overlaid on the *ad hoc* network. The phase change for the secure communications network occurs when

$$k = 2 + \frac{\log(1 - p_{ij}^{(2h-1)})}{\log(1 - (p_{ij}^{(2h-2)})^2)} \tag{3}$$

where k is the number of keyservers, the keyserver serves all nodes within h hops, and p_h is the probability of a walk of h or fewer hops existing between nodes with the labels $i = \left\lceil \frac{n}{2} \right\rceil$ and $j = \left\lceil \frac{n}{2} \right\rceil + 1$ from the *ad hoc* network model.

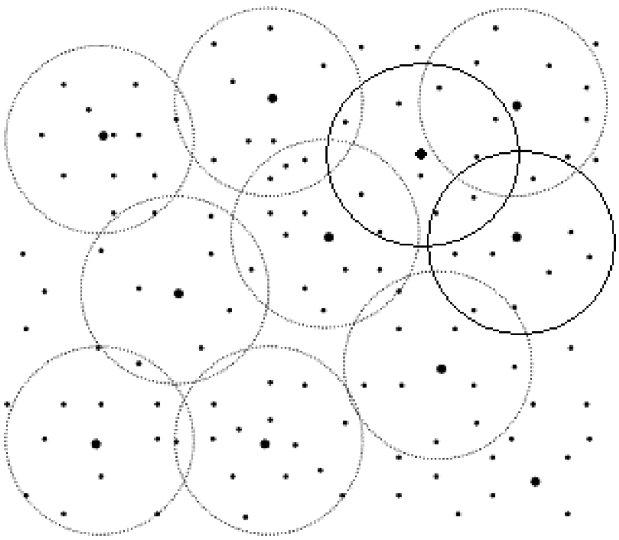

FIGURE 2 Multicast communication topology.

Proof

The phase change occurs when $p_{h+1} = p_h$. By applying equation (2) recursively, we find the likelihood of a walk of 2h-1 hops between nodes i and j:

$$p_{ij}^{(2h-1)} = 1 - \prod_{\substack{l=1 \\ l \neq i \\ l \neq j}}^{n} (1 - p_{il}^{(2h \cdot 2)} * p_{lj}^{(1)}) \tag{4}$$

Two keyservers can communicate if there is a walk of length 2h-1 or less between them. Since keyservers are placed at random on the *ad hoc* network, we have an Erdös-Rényi graph where any two keyservers can communicate with the probability defined in (4). The probability that any two multicast regions with keyservers k_1 and k_2 can communicate using an intermediary is therefore

$$p_{k_1 k_2}^{(2)} = 1 - \prod_{\substack{i,j=1 \\ i,j \neq k_1 \\ i,j \neq k_2}}^{k} (1 - p_{ij}^{(2h-1)} * p_{ij}^{(2h-1)}) \tag{5}$$

which simplifies to

$$p_{k_1 k_2}^{(2)} = 1 - (1 - (p_{ij}^{(2h-1)})^2)^{k-2} \tag{6}$$

so that the phase change occurs when

$$p_{ij}^{(2h-1)} = p_{k_1 k_2}^{(2)} = 1 - (1 - (p_{ij}^{(2h-1)})^2)^{k \cdot 2} \tag{7}$$

Taking the log of both sides and rearranging terms yields equation (3), which was the item to be proved. Q.E.D.

Simulations of our *ad hoc* model were run using MATLAB to verify these analytical predictions. The phase change predictions are shown in Figs. 3 and 4. For each simulation, the normalized value for the radius of communication r, was varied from 0.04 to 0.20. The radius r is normalized with respect to the dimensions of the area where the sensor network is deployed.

Figure 3 shows phase change for the *ad hoc* network with communication range r = 0.07. The filled circle is the predicted inflection point. Error bars for 95% confidence intervals are shown. The graphs show the mean of 35 repetitions. The approach predicts that the point of phase change is at 42 keyservers. At this point, 88% of the keyservers are in the same component.

Figure 4 shows failure to form a giant component. An *ad hoc* network of 1000 nodes with a range of 0.02 was simulated. The network viability criterion fails in this case. The size of the largest component keeps decreasing as more keyservers are added as shown in Fig. 4. With these conditions, 1236 keyservers would be needed to form a giant component from the analytical equation (3). Since this is more than the number of nodes (1000), the giant component cannot form and the network breaks. This agrees with the predictions made by our phase change analysis.

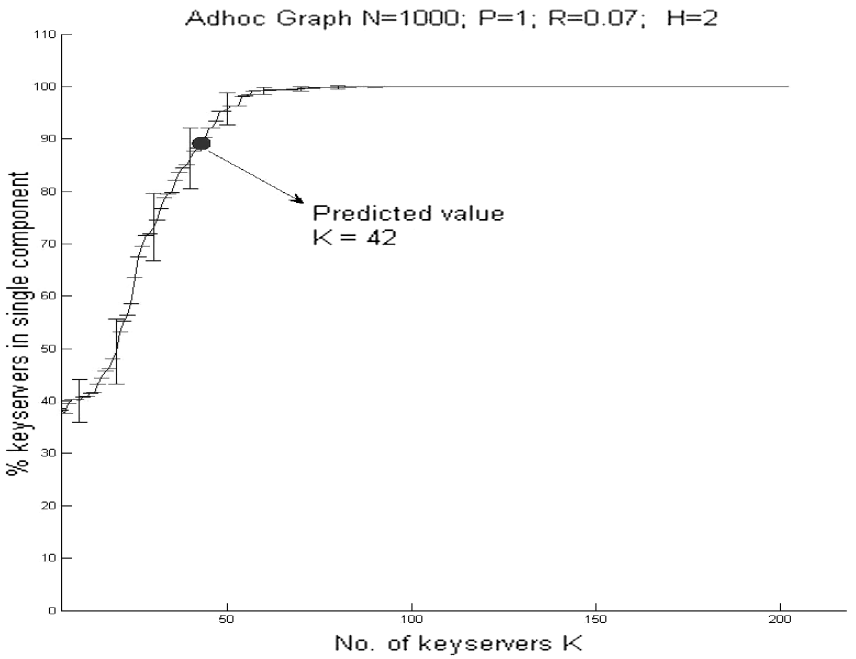

FIGURE 3 Percent of keyservers included in the giant component versus the number of keyservers in a network above the percolation threshold.

Simulations with the number of sensor nodes n being 100, 200, 500, 1000, and 3000 yielded similar results. The approximation achieved by this model is good, but not perfect. One reason for the deviations is the use of expected values in the derivation of the *ad hoc* network model. For graph instances with a small number of nodes the variance of the node positions is greater; and second order effects are possible. Using expected values also assumes independence between random variables. Independence may not strictly hold throughout the range-limited graph construction process. On the other hand, the predicted inflection point is close to the value found by the simulations.

For Erdös-Rényi graphs, mathematicians have determined that the phase change occurs when the number of edges is $E = n/2 + O(n^{2/3})$ [18]. Note that these results are asymptotic as the graph size approaches infinity and constant offsets are not considered in the O notation. Results from our approach are therefore consistent with the analysis in [18] and [13].

3. Determination of Multicast Parameters

In [2] we discussed the protocol for initializing and maintaining secure multicast regions by the sensor network. The total number of messages required to set up a single multicast region is $5(n_c–1)$, where n_c is the number of nodes in, or the size of, the multicast region. This overhead is a minimum when every node is a keyserver for its own multicast region, because the number of hops within the region is zero and the size of every multicast region is unity. Security is maintained in the network by a group agreement protocol which requires message overhead proportional to k^2 *and* n_c*;* k being the number of keyservers and n_c the number of sensor nodes in a multicast region. These security features favor a network with minimum keyservers to reduce the message overhead for group agreement. The number of sensor nodes in a multicast region is proportional to the number of hops h.

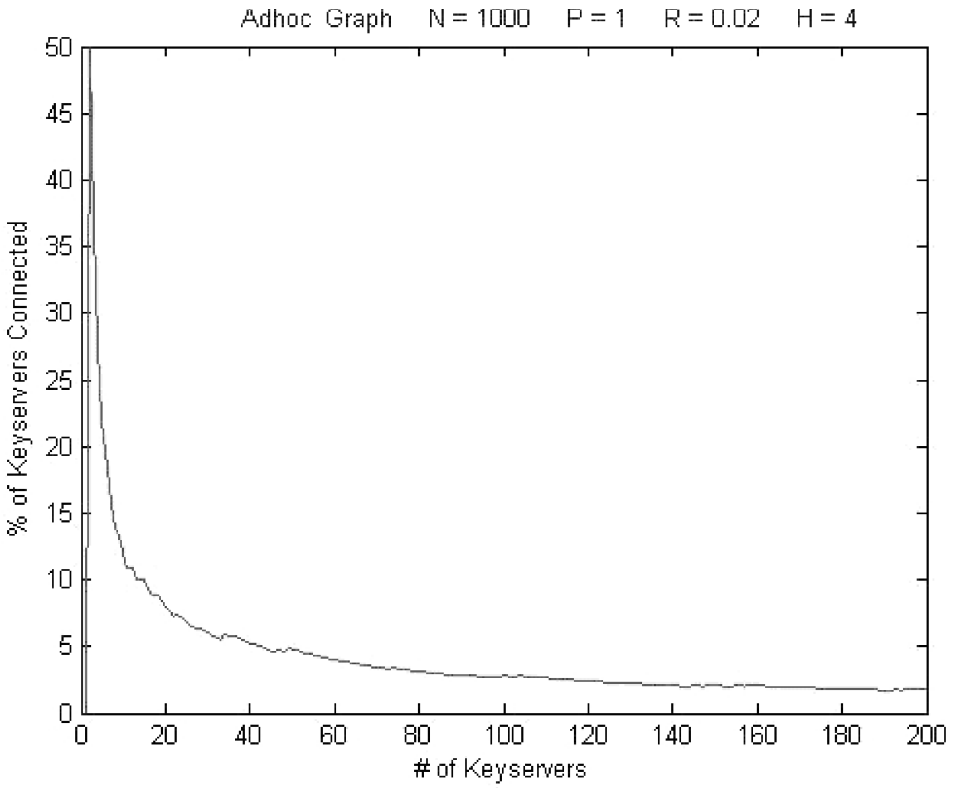

FIGURE 4 Percent of keyservers included in the giant component versus the number of keyservers in a network below the percolation threshold.

The multicast architecture and group agreement protocol thus setup a tradeoff between the number and size multicast regions, *viz. k* and *h* to minimize the message overhead.

We compute n_c as a function of *h*. The network of keyservers is modeled as an Erdös-Rényi random graph overlaid on a range-limited graph of sensor nodes. Every keyserver has an equal probability of communicating with another keyserver in the network within *2h* hops. Each node can communicate with all other sensor nodes physically located within its communications range in a single hop. The area covered by this range is $\pi^* r^2$. The mean field approximation *p*, the probability that any node is within range of a given node, is $\pi^* r^2 / A$, where *A* is the size of the field or region being surveyed. Each keyserver serves all nodes within *h* hops. The likelihood that a node is within *h* hops can therefore be estimated as $\pi^*(h^*r)^2 / A$. The communications range of a node is a circle of radius *r* around it. The network of numerous sensor nodes is laid within a bounded region called the field. The sensing area does not completely overlap with the field for nodes scattered at the edges. If the size of a multicast region is *h* hops, then the region outside the field would be within a radius of at most *(h*r)*. We compensate for edge effects that are an artifact of our model by inflating the area considered by a factor of *(h*r)*. Node placement follows a binomial distribution. The probability density function for the number of nodes within *h* hops becomes *phk*, where

$$phk = \binom{n}{k} P^k (1-P)^{(1-k)} \text{ and } \mathrm{P} = \pi^* (h*r)^2 / A*(1+h*r)^2 \tag{8}$$

The expected number of nodes within h hops of the keyserver becomes

$$n_c = \sum_{k-0}^{n\cdot 1} k.phk \tag{9}$$

In [2], we discuss a distributed key agreement protocol using counting Bloom filters to detect compromised nodes in the network. A Bloom filter [26] is an approximate representation of a set that supports membership queries. It is a vector of m bits. Initially all bits in the vector are set to 0. Each member of the set is hashed using h hash functions each with range [1 . . . m]. The bit corresponding to each hash value is set to 1. A bit might be set more than once. Counting Bloom filters [26] are an extension of this idea where each bit is replaced by a small counter. A round of key agreement needed to detect compromised nodes requires an exchange of $k(n_c–1) + 5(k^2–k)/2$ messages. The message overhead for the distributed key agreement protocol increases with the number of keyservers.

A Byzantine agreement protocol [2] ensures that cloned keyservers do not falsify information when exchanging the counting Bloom filters. This can be done by introducing redundancy and allowing nodes to be served by multiple keyservers. This redundancy also improves the accuracy of the key usage statistics reported by the Bloom filters. Thus both security measures have a message overhead of the order of k^2. To minimize traffic for the agreement protocol, we need to minimize the number of keyservers. Thus a trade-off exists between the number of keyservers and the number of hops. We use the results for predicting the phase change in the network from the previous section to find the values of k and h that minimize the overhead required to establish sensor network security.

$$\text{Total messages to set up } k \text{ multicasts} = k*(5(n_c-1)) \tag{10}$$

$$\text{Total messages for key agreement} = k(n_c-1)+5(k^2-k)/2 \tag{11}$$

The total number of messages for both is

$$Ms = k*(6*(n_c-1)+5*(k-1)/2) \tag{12}$$

The multicast group size n_c is a function of n, r, and h as shown in equations 8 and 9. Also, equation (3) shows that k is indirectly dependent on n, r, and h. Thus, the optimization problem of minimizing Ms, subject to k and h, can be solved using gradient descent or any numerical optimization algorithm [17]. Gradient descent is an iterative algorithm to find the local minima of a function that involves moving in the direction of the negative gradient from an initial estimate. We assume the initial point for $h = 1$.

Table 1 shows the results for a network of 100 nodes with a communication range of 0.2. It is clear from the data that, for this instance, the network can be established with a minimum number of messages with 4 legitimate keyservers and 2 hops from each keyserver. Cluster size estimates from [2] give a cluster size of 26 nodes for a network with 100 nodes with a communication radius of 0.2. Hence the optimal parameters for this network are to have at least 4 keyservers each with a cluster of all nodes within 2 hops of the keyserver.

However, assume that c nodes in the network are compromised. Since every node is equally likely to be elected as the keyserver (proved in [2]), the expected number of

TABLE 1 Number of messages necessary to establish a network for different network definition parameters

Nodes = 100; range = 0.2				
Hops in region	Number of Keyservers	Expected number of nodes in region	Keyservers needed to tolerate 25 clones	Messages
1	8	10.3	16	1492.8
2	4	25.8	8	1330.4
3	3	45.6	7	1978.2
4	3	56.3	7	2427.6

compromised keyservers is $\left\lceil \frac{c}{n} * k' \right\rceil$. We need k legitimate keyservers (from equation 7) to maintain the giant component so that the network is viable.

We pick k' keyservers to introduce redundancy required by the Byzantine agreement protocol [25]. The protocol discards $2*\tau$ (the τ largest and τ smallest) values where τ indicates the number of adversaries the network can tolerate as keyservers. Hence, an extra $2*\left\lceil \frac{c}{n} * k' \right\rceil$ keyservers are introduced. To tolerate c clones in the network of n nodes we pick k' keyservers such that

$$k' = 2*\left\lceil \frac{c}{n} * k' \right\rceil + k. \tag{13}$$

In the above example, to tolerate a network where 25 percent of the nodes are clones, we need to have 8 keyservers.

Figure 5 plots the total messages required to initialize the network versus the number of hops in a multicast region. In this graph, when $h = 5$ the number of nodes in a single multicast region is of the order of total nodes in the network. The number of multicast regions reduces with increasing region size. However, the number of messages neither

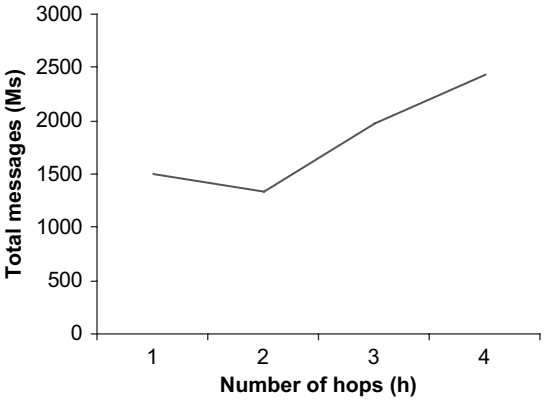

FIGURE 5 Plot of number of messages for multicast (communication range = 0.20) initialization for varying multicast size (number of hops).

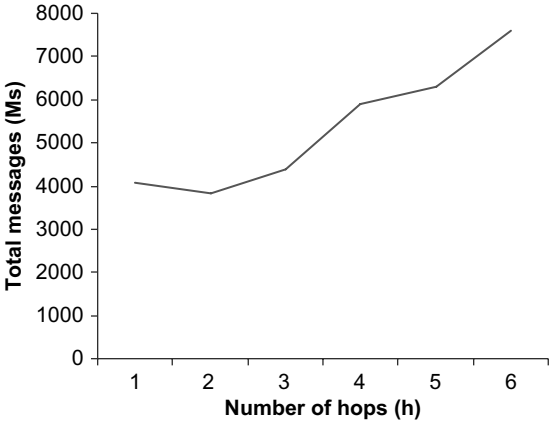

FIGURE 6 The number of messages required to initialize multicast (communication range = 0.15) regions versus the number of hops served.

decreases nor increases uniformly with region size. In this instance, use of $h = 1$ results in 16 multicast regions. The number of regions drops to 8 when h increases to 2. Larger values of h do not significantly reduce the number of regions, but result in additional overhead for key maintenance. This highlights the importance of calculating the minima, to reduce overhead and power consumption.

Similar results are shown in Fig. 6 where the communication radius r was reduced to 0.15. The total nodes in the network n remain at 100. The optimal configuration has 27 keyservers serving all nodes within 2 hops of the keyserver. The number of messages decreases as long as the multicast group size remains between 1 to 3 hops. The optimum occurs with 27 keyservers serving all nodes within 2 hops.

4. Application

Consider the field test of ColTraNe [11] conducted in November 2001. Military targets were tracked using a sensor network of 70 nodes. Each node broadcasts a closest point of approach (CPA) packet to all neighboring nodes when a target is detected. A dynamically chosen local clump head, i.e. the node with the highest intensity of a target signal, calculates target velocity and heading from the CPA data and forwards a tracking packet to nodes likely to detect the target in the future. The tracking was implemented using the Extended Kalman filter (EKF), lateral inhibition, and a combination of both. The number of tracking packets, CPA packets and inhibition packets are shown in Table 2 [11]. The numbers in parentheses indicate packet size in bytes.

The node layout for the field test has a maximum of 20 hops between nodes. On an average, each node had 4 to 5 nodes within a 1-hop radius and 12 nodes within a 2- hop

TABLE 2 Data transmission requirements [11] for tracking application in ColTraNe

	Tracking packets	CPA packets	Inhibition packets	Total bytes sent over the network
EKF	852 (296)	59 (40)	0 (56)	254552
Lateral Inhibition	217 (56)	59 (40)	130 (56)	21792
EKF & Lateral Inhibition	204 (296)	59 (40)	114 (56)	69128

radius. We assume our multicast topology will require 10 multicast regions where each region will contain nodes within a 2-hop radius.

The tracking packets are transmitted to all nodes that are in the direction of the target. The tracking packets contain sensitive information that has to remain secure from adversaries. An existing pair-wise key encryption technique would require a separate encryption for every recipient of the same packet. Hence, the existing implementation in ColTraNe would require one encryption/decryption for every packet, i.e. 852 encryptions/decryptions. Our approach would require one encryption/decryption per multicast region. Only one encryption will be required for every 12 nodes; hence the total encryptions required would be 71. When the tracking algorithm used lateral inhibition, only 18 encryptions would be required using our approach, as opposed to 217 with the existing encryption technique.

A CPA packet is transmitted only to nodes within the vicinity. The network topology for ColTraNe shows an average of 5 nodes lying in the vicinity of any node. Hence, 59 CPA packets indicate that 12 CPA events were generated during the tests. Our multicast communication would require only one encryption for every CPA event since all nodes in the vicinity would fall into a single multicast region. If the node generating the CPA event is common to two multicast regions, that CPA packet would require two encryptions. A worst-case estimate using our approach is 24 encryptions for all 59 CPA packets.

The lateral inhibition approach involved only selective nodes forwarding track information. Our approach would require as many encryptions as multicast regions assuming a worst case that each inhibition packet is forwarded to all nodes.

EKF requires 71 encryptions for tracking packets and 24 encryptions for CPA packets, totaling to 95 encryptions using our approach. Similarly, the lateral inhibition implementation would require 18 encryptions for tracking packets, 24 encryptions for CPA packets, and 12 encryptions for inhibition packets, adding up to 53 encryptions. Table 3 compares the total number of encryptions required by an existing point-to-point communication scheme and our approach for the ColTraNe application. The numbers within parentheses indicate the total number of bytes to be encrypted.

AES encryption on a MC68328 DragonBall consumes 0.000101 mJ/bit. The estimated power consumption is shown in Table 4. Energy consumption is much lower in our approach for secure transmission of tracking information.

TABLE 3 Number of encryptions required for secure transmission in ColTraNe

	Our approach	Point-to-point communication
EKF	95 (21976)	911 (254552)
Lateral Inhibition	53 (2584)	406 (21792)
EKF & Lateral Inhibition	51 (6552)	377 (69128)

TABLE 4 Power consumption comparison using AES encryption

	Our approach	Point-to-point communication
EKF	17.76 mJ	205.68 mJ
Lateral Inhibition	2.09 mJ	17.61 mJ
EKF & Lateral Inhibition	5.29 mJ	55.86 mJ

TABLE 5 Power consumption for security and communication

	Security using our approach	Communication using Bluetooth
EKF	17.76 mJ	203.64 mJ
Lateral Inhibition	2.09 mJ	17.43 mJ
EKF & Lateral Inhibition	5.29 mJ	55.30 mJ

Communications require from ~ 40 * 10^{-6} joules (GSM cellular phone) to 1 * 10^{-7} joules (Bluetooth for 10s of meters) per bit. Reception energy needs for GSM are 2*10^{-6} joules per bit and 10^{-7} joules per bit for Bluetooth. [27] Assuming Bluetooth communication, we predict the power consumption for transmission of the messages. Table 5 compares the energy consumption for security with that required for communication.

The computations suggest that a sensor node using a point-to-point encryption mechanism expends the same amount of energy for security as required for communication, whereas when using a multicast approach with the optimal number of keyservers and region size, the power requirements for maintaining security can be reduced to one-tenth.

5. Conclusion

Phase change analysis is extremely important in *ad hoc* sensor network design to maintain network viability. We develop equations to predict this phase change. We use the multicast approach explained in [2] for secure and efficient communication between sensor nodes in a random network. This paper derives equations on how to predict the number of keyservers required to maintain connectivity in the network without compromising on security. We find an optimum value for the size of each multicast (number of hops) to reduce message overhead for network initialization.

Further research could try to establish an optimal value for the communication radius to reduce power consumption and increase the network lifetime. In the future, we wish to implement our approach on actual test beds and expand this approach to counter other attack mechanisms.

About the Authors

Dr. Richard Brooks has a Ph.D. in Computer Science from Louisiana State University, and a B.A. in Mathematical Sciences from Johns Hopkins University. He is currently an associate professor of electrical computer engineering at Clemson University in Clemson, South Carolina. He was previously the head of the Distributed Systems Department of the Pennsylvania State University Applied Research Laboratory. Dr. Brooks was PI of the Mobile Ubiquitous Security Environment (MUSE) Project sponsored by ONR as a Critical Infrastructure Protection University Research Initiative (CIP/URI). He is author of *Disruptive Security Technologies with Mobile Code and Peer-to-peer Networking* from CRC Press. He was co-PI of a NIST project defining security standards for networked building control systems. He has had other research projects funded by ARO, ONR, and DARPA. His Ph.D. dissertation received an exemplary achievement certificate from the Louisiana State University graduate school. His current research concentrates on distributed strategic systems for network security and national defense. He has a broad professional background with computer systems and networks. This includes being technical director of Radio Free Europe's computer network for many years. His consulting clients

include the French stock exchange authority and the World Bank. While with the World Bank, he expanded their internal network to sub-Saharan Africa, Eastern Europe, and the Former Soviet Union.

Brijesh Pillai received his M. S. degree in Computer Engineering from the Holcombe Department of Electrical and Computer Engineering in Spring 2006. His thesis topic was *Network Embedded Support for Sensor Network Security*. It proposed solutions for countering cloning and Sybil attacks in sensor networks.

Michele Weigle is an Assistant Professor of Computer Science at Clemson University. She received her Ph.D. from the University of North Carolina at Chapel Hill in 2003. Her research interests include network protocol evaluation, network simulation and modeling, Internet congestion control, and mobile ad-hoc networks.

Matthew Pirretti is a Ph.D. candidate with the Computer Science and Engineering (CSE) Department of the Pennsylvania State University in University Park, PA. He has his B.S. and M.S. degrees in CSE from Penn State. His research interests include sensor network security issues.

References

1. S. S. Iyengar and R. R. Brooks, ed.'s, *Distributed Sensor Networks*. Boca Raton, FL: Chapman & Hall, 2005.
2. B. Pillai, "Network embedded support for sensor network security," Masters thesis, ECE Dept, Clemson University, May 2006.
3. G. J. Pottie, and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, May 2000.
4. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications*, vol. 40, no. 8, pp. 102–114, Aug. 2002.
5. E. Slavin, R. R. Brooks, and E. Keller, "A comparison of tracking algorithms using beamforming and CPA methods with an emphasis on resource consumption vs. performance," *PSU/ARL ESP MURI Technical Report*, 2002.
6. J. Chen and K. Yao, "Beamforming," in *Distributed Sensor Networks*, (ed.s) S. S. Iyengar and R. R. Brooks, Boca Raton, FL: Chapman & Hall, 2005.
7. S. Phoha and R. Brooks, *Emergent Surveillance Plexus MURI Annual Report*, The PSU Applied Research Laboratory, Report 2, DARPA & ARO (March 2003).
8. R. Brooks, C. Griffin, and D. S. Friedlander, "Self-organized distributed sensor network entity tracking," *International Journal of High Performance Computer Applications, special issue on Sensor Networks*, vol. 16, no. 3, pp. 207–220, Fall 2002.
9. R. R. Brooks, P. Ramanathan, and A. Sayeed, "Distributed target tracking and classification in sensor networks," in *Proceedings of the IEEE*, Invited Paper, vol. 91, no. 8, pp. 1163–1171, August 2003.
10. R. Brooks, Friedlander, E. Grele, C. Griffin, N. Jacobson, T. Kaiser, J. Koch, S. Phoha, J. Moore, and T. Reggio, "Distributed tracking and classification of Land vehicles by acoustic sensor networks," *Journal of Underwater Acoustics*, Classified Journal, Invited Paper, In Press, October 2003.
11. R. R. Brooks, D. Friedlander, J. Koch, and S. Phoha, "Tracking multiple targets with self-organizing distributed ground sensors," *Journal of Parallel and Distributed Computing Special Issue on Sensor Networks*, vol. 64, no. 7, pp. 874–884, August 2004.
12. Bhaskar Krishnamachari, Stephen B. Wicker, and Ramon Bejar, "Phase transition phenomena in wireless ad-hoc networks," *Symposium on Ad-Hoc Wireless Networks, GlobeCom2001, San Antonio, Texas, November 2001.*
13. B. Bollobás, *Random Graphs*, Cambridge, University Press, *Cambridge* 2001.
14. D. Stauffer, Aharony, *Introduction to Percolation Theory*, London, Taylor & Francis, 2001.
15. R. R. Brooks, "Random networks and percolation theory," Chapter 49. *Distributed Sensor Networks*, eds. S. S. Iyengar and R. R. Brooks, pp. 907–946, Boca Raton, FL: Chapman & Hall/CRC Press, 2005.

16. Brooks R, Rai S, Racunas S, Pillai B. "Mobile network analysis using probabilistic connectivity matrices", accepted for publication in *IEEE Transactions on Systems, Man, and Cybernetics—Part C* (2006).
17. http://documents.wolfram.com/v5/TheMathematicaBook/AdvancedMathematicsInMathematica/NumericalOperationsOnFunctions/3.9.8.html
18. S. Jensen, T. Luczak, A. Rucinski, *Random Graphs*, New York: John Wiley & Sons, 2000.
19. D. W. Carman, P. S. Kraus, and B. J. Matt, Constraints and Approaches for Distributed Sensor Network Security (Final), *NAI Labs Technical Report #00–010*, September 1, 2000.
20. N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha, "Analyzing the energy consumption of security protocols," *Proc. International Symposium on Low Power Electronics and Design*, pp. 30–35, 2003.
21. D. W. Carman, "Data security perspectives," in *Distributed Sensor Networks*, (ed.s) S. S. Iyengar and R. R. Brooks, Chapman and Hall, 2005.
22. R. R. Brooks, S. Amarnath, and H. Siddul, "On adaptation to extend the lifetime of surveillance sensor networks," *Innovations and Commercial Applications of Distributed Sensor Networks Symposium*, Bethesda, MD, October 2005.
23. P. Erdös and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst.Hung. Acad. Sci. 5*, 1960, 17–61.
24. L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41–47, Nov. 2002.
25. R. R. Brooks and S. S. Iyengar, Multi-Sensor Fusion: Fundamentals and Applications with Software, *NJ. Prentice – Hall PTR, 1998.*
26. R. R. Brooks, P. Y. Govindaraju, M. Pirretti, N. Vijaykrishnan & M. Kandemir, "On the detection of clones in sensor networks using random key predistribution," *IEEE Transactions on Systems Man and Cybernetics, Part C*, accepted for publication.
27. L. Doherty, B. A. Warneke, B. E. Boser, and K. S. J. Pister, "Energy and performance considerations for smart dust," *International Journal of Parallel and Distributed systems and Networks*, vol. 4, no. 3, pp 121–133, 2001.