Secure and Efficient Multicast in Wireless Sensor Networks Allowing Ad hoc Group Formation

Kui Ren, *Member, IEEE,* Wenjing Lou, *Senior Member, IEEE,* Bo Zhu, *Member, IEEE,* and Sushil Jajodia, *Senior Member, IEEE*

***Abstract*—Multicast security is one of the most important security services in wireless sensor networks (WSNs) since it enables a sink to multicast messages to sensors in a secure manner. While multicast authentication has widely been addressed in the literature, the problem of multicast encryption still remains open in WSNs. In this paper, we propose a multicast encryption scheme called *global-partition, local-diffusion* (GPLD) that focuses on scheme efficiency and supports various multicast group semantics. GPLD partitions sensors into a series of elementary groups using their location and class information and accordingly builds a location-class-aware symmetric key management framework. Furthermore, the scheme leverages the fact that sensors are both end receivers and routers, which effectively minimizes global (sink-to-sensor) group key distribution and rekeying traffic while supporting various multicast group semantics. The efficiency and security properties of GPLD are justified through both analysis and simulations.**

***Index Terms*—Efficiency, encryption, location, multicast security, rekey traffic, sensor networks.**

I. Introduction

MULTICAST communication from a sink to sensors is of great importance in a wireless sensor network (WSN), as it enables the sink to disseminate query and control messages to the sensors and, thus, efficiently operate the WSN. Multicast security is, hence, one of the most important security services in WSNs [1], [5], [19], [22]. Recently, many schemes have been proposed to address the problem of multicast authentication in WSNs [19], [22]. These schemes aim at providing efficient authentication solutions for the multicast traffic and, hence, ensure message authenticity and prevent message fabrication and alteration attacks. While multicast authentication has extensively been studied, there has been very little work addressing the problem of multicast encryption in the context of WSNs. Multicast encryption is orthogonal to multicast authentication; it provides message confidentiality and ensures that the message content can only be recovered by the intended receivers. The demand for multicast encryption is twofold. First, it ensures message confidentiality and privacy. For example, the query message regarding the health status of patients should always be kept confidential from people other than the responsible doctors/nurses in the case of a health-oriented WSN, such as CodeBlue [13]. Second, it minimizes the security risk (i.e., information leakage and key compromise) resulting from sensor compromise, which is unavoidable when the WSN is deployed in hostile environments. Hence, the problem of multicast encryption has to be addressed before multicast services can be deployed in practice.

Designing an applicable multicast encryption scheme for WSNs is challenging. On the one hand, multicast services in WSNs have various semantics and are inherently multigroup oriented. On the other hand, WSNs usually have a large network size, and sensors are resource constrained and subject to potential compromise when deployed in hostile environments. These factors pose drastic efficiency and security requirements on the design of multicast encryption schemes.

In this paper, the problem of multicast encryption in WSNs is addressed. We aim at providing message confidentiality for the multicast traffic from a sink to the sensors. We approach the problem by first classifying the multicast group semantics in WSNs. We then propose our scheme called *global-partition, local-diffusion* (GPLD). GPLD focuses on scheme efficiency and its support to various multicast group semantics. GPLD partitions sensors into a series of elementary groups using their location and class information and accordingly builds a location-class-aware symmetric key management framework. Further leveraging the fact that sensors are both end receivers and routers, GPLD develops a novel multicast encryption technique called *global-partition, local-diffusion*. This technique effectively minimizes global (sink-to-sensor) group key distribution and rekeying traffic while maintaining its support to various multicast group semantics. The efficiency and security properties of GPLD are justified through both analyses and simulations.

This paper makes three contributions.

1) We analyze and classify the multicast group semantics that are inherently demanded by WSNs.

Manuscript received December 7, 2007; revised May 10, 2008 and June 26, 2008. First published August 12, 2008; current version published April 22, 2009. This work was supported in part by Educational and Research Initiative Fund, Illinois Institute of Technology, by the National Science Foundation under Grant CNS-0626601, Grant CNS-0716306, Grant CNS-0831963, and Grant CNS-0831628, by the National Sciences and Engineering Research Council of Canada, by the National Science Foundation under Grant CT-0716567, Grant CT-0627493, and Grant IIS-0430402, and by the Air Force Office of Scientific Research under Grant FA9550-07-1-0527. The review of this paper was coordinated by Dr. L. Chen.

K. Ren is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616-3793 USA (e-mail: kren@ece.iit.edu.).

W. Lou is with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609-2280 USA (e-mail: wjlou@ece.wpi.edu.).

B. Zhu is with Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC H3G 2W1, Canada (e-mail: zhubo@ciise.concordia.ca.).

S. Jajodia is with Center for Secure Information Systems, George Mason University, Fairfax, VA 22030 USA (e-mail: jajodia@gmu.edu.).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2008.2003961

2) To the best of our knowledge, we propose the first scheme (i.e., GPLD), which supports various multicast group semantics, as well as dynamic changing and simultaneous formation of multicast groups.
3) Our scheme can achieve scheme efficiency and meet the resource-constrained nature of WSNs. More specifically, the communication overhead of our scheme is only slightly higher than the theoretical lower bound. As a tradeoff, the storage requirement per node is significantly reduced from *exponential* (with respect to the number of sensors in the network) to *linear* (with respect to the number of neighboring nodes and the number of levels of the hierarchical structure employed in our scheme, which is usually a small value that is less than 10).

The remainder of this paper is organized as follows. In Section II, we discuss multicast group semantics in WSNs. Section III presents related work. In Sections IV and V, we introduce our proposed scheme in detail. Sections VI and VII present the security and performance analysis of the proposed scheme, respectively. Finally, Section VIII concludes this paper.

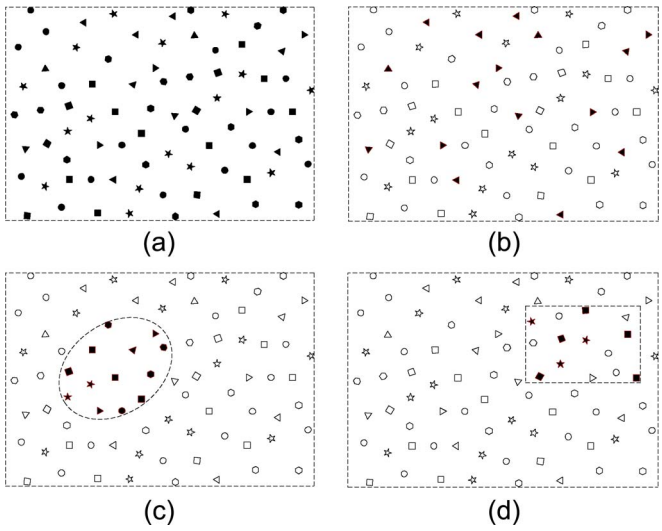

Fig. 1. Multicast group semantics in WSNs, with the solid symbols denoting the intended recipients of the multicast messages in each case. (a) Broadcast. (b) Class-based multicast. (c) Location-based multicast. (d) Location-class-based multicast.

II. Multicast Group Semantics in WSNs

Consider a military application where a large number of sensors with different functionalities are deployed in the strategic field to detect and identify the presence of critical events of interest, as shown in Fig. 1, where each symbol denotes a different sensor class with a different functionality, such as image sensors, acoustic sensors, and actuators.[1] Different classes of sensors are used for different purposes. For example, image sensors may be used to identify enemy tanks and soldiers, acoustic sensors may be used to detect other targets based on acoustic signals, and actuators may launch certain actions such as activating the preinstalled military devices upon the command from the sink. At the same time, all sensors also collaborate with each other and form a multihop wireless network to support network communications.

As WSNs are inherently location aware and function specific, multicast group semantics from the sink to the sensors can be classified into four most common categories, as shown in Fig. 1.

1) Broadcast, where all network sensors are the intended recipients of multicast messages, i.e., *recipient sensors* [Fig. 1(a)];
2) Class-based multicast, where only the sensors of a certain class are the *recipient sensors* [see Fig. 1(b)];
3) Location-based multicast, where the sink may multicast to groups of sensors, subject to certain dynamic spatial constraints [see Fig. 1(c)]. Since sensors are always deployed in a discrete manner at a certain density, we can easily express the location constraints of sensor groups as a few basic geometric shapes, which can efficiently be described using simple mathematical representations. In Fig. 1(c), the *recipient sensors* are the sensors located inside the elliptic area;
4) Location-class-based multicast, where the sink may also multicast messages to groups of sensors, subject to both spatial constraints and class requirements [see Fig. 1(d)]. In Fig. 1(d), the *recipient sensors* are the sensors of classes "$\star$" and "$\square$" that are located inside the rectangular area.

Depending on different applications, more sophisticated semantics may exist, but these four categories are certainly the most common categories and suffice in most scenarios. Therefore, any multicast encryption scheme designed for WSNs has to support (at least) these multicast group semantics.

III. Related Work

A multicast encryption problem has extensively been addressed in the context of wired networks and ad hoc networks. Here, we introduce some typical schemes that are closely related to this work.

1) Group Key Distribution Schemes: The logical key hierarchy (LKH) model was first introduced in [26] to address secure multicast for the Internet. For each group, LKH maintains a key tree, which is used for group key update and distribution. The root of the key tree is the group key used for encrypting data in multicast and is shared by all users. The leaf nodes of the key tree are the keys shared only between the individual users and the key distribution center (KDC), whereas the intermediate nodes are the auxiliary key encryption keys used to facilitate the distribution of the root key, i.e., the group key. Of all these keys, each user stores the keys from its leaf node all the way up to the root of the key tree. As a result, when a user joins/leaves the group, all the keys on its path (i.e., from its leaf node to the root node of the key tree) have to be changed and redistributed to maintain backward/forward data confidentiality. Various schemes such as OFT [2], ELK [18], and Seclor [11] are later proposed to further optimize rekeying

[1] In this paper, we do not distinguish sensors from actuators.

overhead. Group key distribution schemes are unsuitable for WSNs, because they are inherently single group oriented. For a single group, these schemes require a storage overhead of $\mathcal{O}(\log N)$ keys, and to revoke a single user, KDC has to send the rekeying message containing $\mathcal{O}(\log N)$ keys, where N is the group size. However, in WSNs, there may exist a large number of ad hoc and dynamic groups due to its abundant multicast group semantics. Thus, it is highly inefficient, if not impossible, for these schemes to support multicast encryption in WSNs.

2) Broadcast Encryption Schemes: First introduced in [14], broadcast encryption schemes enable a centralized server to securely multicast messages to a dynamically changing subset of users of a group. In [14], an efficient broadcast encryption scheme called SD was proposed based on a subset-cover framework. In contrast to group key distribution schemes, SD is stateless. That is, a user receiving only the current rekeying message can recover the group key used for the current session based on his initial configuration, even if he missed previous rekeying operations. In addition, unlike group key distribution schemes, SD allows multiuser revocation at a time. SD is, by far, the most efficient broadcast encryption scheme in terms of rekeying message size, which is $1.25r$ keys on average and is bounded by $2r - 1$ keys, where r is the number of group users excluded from the recipients of the current session. SD further requires a storage overhead of $\mathcal{O}(\log^2 N)$ keys at each user. When applied to WSNs, SD is still highly inefficient. For example, consider a multicast session in a WSN that consists of 10 000 sensors. If the sink wants to multicast a subset of sensors, for example, 8000 of them, the size of the rekeying message for this session is 2500 keys on average, and such rekeying messages are broadcast to the whole WSN. Obviously, this is impractical in WSNs.

3) Other Multicast Encryption Schemes: In [34], GKMPAN was proposed to address secure multicast in ad hoc networks. GKMPAN assumes that all nodes in an ad hoc network are predistributed with a certain number of keys m randomly out of a big pool of l keys, which are used to update group keys. If a node is compromised, the key server first determines a noncompromised key, which is the most common among the remaining members of the group. Then, the key server broadcasts a new group key encrypted with the chosen noncompromised key. Consequently, nodes that have this key can independently decrypt the group key. These nodes further reencrypt the new group key with another noncompromised key and forward it to those neighbors yet to obtain it. In this way, the new group key is propagated to all the members in a hop-by-hop fashion. However, GKMPAN is vulnerable to the selective node compromise attack. Compromising as low as 50 out of 1 000 000 nodes could be sufficient to break the whole scheme, given $m = 100$ and $l = 5000$. This attack is possible, because the attacker can derive which keys are carried by which nodes simply based on the nodes' identification (ID) and hence can selectively compromise those nodes carrying no keys in common. Additionally, GKMPAN only supports the single-multicast-group scenario. Hence, it is inapplicable to WSNs.

In [20], LKH wireless (LKHW) was proposed, which directly applies the LKH technique into WSNs while using directed diffusion [9] to support membership management. LKHW only considers the single-group case and also suffers from many attacks. There are also two other group key rekeying schemes proposed for WSNs. The scheme proposed in [4] aims to maintain a network-wide group key in the presence of node compromise, and the scheme in [29] provides an approach to renew group keys for multigroups. In [28], a ciphertext-policy attribute-based encryption technique is explored to specify a multicast group via member attributes for efficient group description. None of the aforementioned schemes supports the multicast group semantics discussed in Section II.

IV. GPLD: Setup

A. System Assumptions and Design Goals

1) Network Model: In this paper, we consider a large-scale WSN that monitors a vast terrain of interest via a large number of static sensors of different functionalities. We assume that the WSN is densely deployed and always well connected; sensors of each class are also interconnected among themselves. We further assume that the approximate estimation on the size and shape of the terrain of interest is known *a priori*. Without loss of generality, we assume that the terrain is square in shape. In WSN, there exists a sink, which is the data collection center equipped with sufficient computation and storage capabilities. We assume that all sensors can receive the messages from the sink since the WSN is well connected. We assume that the communications among the sensors are bidirectional. We do not address the reliability issue of the message delivery [17] since it is orthogonal to this work. In this paper, the sink is the centralized authority that is responsible for the key management tasks to ensure multicast security. We assume that the sensors are classified into several different classes based on their functionalities and are resource constrained in terms of computation, communication, and storage capabilities. The sensors are also not tamper resistant.

2) Threat Model: We assume that the WSN is deployed in hostile environments. The attackers do not only eavesdrop on all the network communications but are also able to compromise a small number of sensors to obtain the contents of the messages multicast by the sink. On the other hand, we also assume that compromised sensors can be detected in a timely manner, and no new sensors are compromised before the current rekeying operation is completed. We do not specify the particular mechanisms that detect compromised sensors, as it is orthogonal to this paper. However, schemes such as watchdog [31] can be well suited for this purpose. We note that before compromised sensors are detected, no key management scheme is able to prevent information from being leaked to the adversary through compromised sensors. However, an effective key management scheme can always exclude the detected compromised sensors from the WSN so that no further damage can occur. Furthermore, we assume that the sink is always secure and has a secure mechanism (e.g., μTESLA [19]) to authenticate its multicast messages to all sensors. In addition, we do not consider denial of service attacks against multicast messages, as they are also out of the scope of this paper.

3) Design Goals: GPLD is designed to achieve the following goals: 1) Support the multicast group semantics discussed

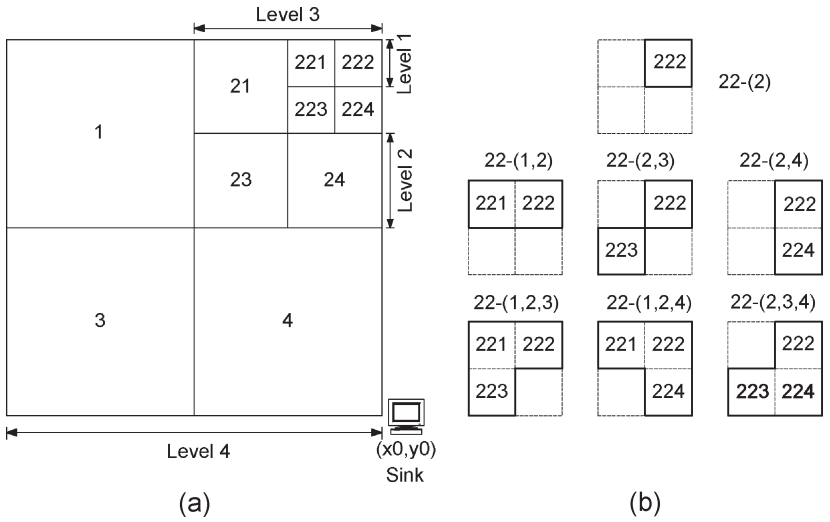

Fig. 2. (a) Virtual grid system partitioning the sensor field using a quad-tree approach ($L = 4$). (b) Seven level-1 location-based elementary groups to which all sensors located at cell 222 belong and their group IDs.

in Section II, 2) provide an efficient group key distribution mechanism to support ad hoc group formation, and 3) provide an efficient rekeying mechanism to support group membership dynamics.

B. Global-Partition, Local-Diffusion Technique

The performance of secure multicast schemes is determined by its group key distribution and/or rekeying operation overhead, as well as the storage and computation overhead. In most existing schemes, it is always the central authority's sole responsibility to deliver the keying materials to each individual group member whenever required; group members are all end hosts, which neither have the responsibility nor are possible for such tasks.[2] However, for secure multicast in WSNs, the sensors are both group members and routers; any multicast message sent by the sink has to be relayed by intermediate sensors before reaching all the target recipients. Consequently, it is possible as well as convenient for sensors to diffuse the group key obtained to other members of the same group in their vicinities. The sink could thus reduce the length of the keying materials it broadcasts to the whole WSN.

Based on this key observation, we develop a *global-partition, local-diffusion* technique, which provides highly efficient group key distribution and rekeying operations. On the one hand, the proposed technique partitions the sensors into a series of predefined elementary groups based on their location and class information. According to this partition, the proposed technique further assigns a common group key encryption key (GKEK) to each elementary group and preloads each sensor with the GKEKs corresponding to all the elementary groups to which it belongs. The proposed technique can hence efficiently support dynamic group formation by utilizing elementary groups and the corresponding GKEKs. These GKEKs can be used to efficiently and securely deliver the fresh group keys to the members of the dynamically formed groups. On the other hand, the proposed technique further avoids a large portion of global (sink-to-sensor) keying material traffic by carrying the minimum number of GKEKs. However, it still guarantees that all the group members obtain the group keys by allowing efficient local (sensor-to-neighbor-sensor) key diffusion.

[2]One group member might not even be aware of the existence of other group members.

C. Grid and Elementary Group Setup

1) Grid Setup: Before network deployment, the network planner prepares a *geographic virtual grid system* for the targeted terrain [21], which partitions the terrain into multilevel cells of different sizes, following a quad-tree approach. Such a grid is described through three parameters, i.e., $\langle (x_0, y_0), L, len \rangle$. (x_0, y_0) is a reference point of the grid, which is usually set as the location of the sink for convenience; L is the number of levels of the corresponding quad-tree; and len is the side length of the lowest level cells. Note that the sensors in the same lowest level cell are always within the direct communication range of each other. Fig. 2(a) shows an example of such a grid, where the quad-tree has four levels, i.e., $L = 4$, and level 1 is the lowest level. Each cell in the grid is uniquely indexed based on its position; a level-i cell is uniquely indexed by $L - i$ digits, with each digit ranging from one to four. Particularly, the level-L cell refers to the whole WSN and is indexed by 0. In the example, cell 222 denotes a level-1 cell located at the top right corner of its belonging level-2 cell, this level-2 cell is located at the top right corner of its own belonging level-3 cell, etc. In our definition, if a sensor is located at a certain cell, we call this cell the *home cell* of that sensor. Clearly, every sensor has one *home cell* at each level.

2) Elementary Groups: GPLD further defines six kinds of elementary sensor groups based on the grid concept.

1) Network-wide group: Sensors from the level-L cell form a network-wide sensor group.
2) Individual groups: Each sensor is itself an elementary group by definition.

3) Neighbor-pair groups: Each pair of immediate neighbor sensors forms such a group.
4) Class-based groups: The sensors of each different class form a class-based group.
5) Location-based groups: For every four level-i ($i \in [1, L-1]$) cells constituting a level-$(i+1)$ cell, the sensors from each possible combination of these four level-i cells form a location-based group.
6) Location-class-based groups: Within each location-based group, the sensors of each different class form a location-class-based group.

Here, the network-wide group is the largest group, an individual group is the smallest, and a neighbor-pair group is the second smallest. Furthermore, we say that one elementary group is larger than another if the former contains more level-1 cells than the latter, and a location-based group is said to be larger than a location-class-based group containing the same number of level-1 cells.

3) Group ID: Each of these elementary groups is uniquely indexed in GPLD to facilitate the subsequent scheme operations. For the network-wide group, the group ID is set as ('all'). For an individual group corresponding a sensor S_u, the group ID is set as $(\mathtt{sink}, \mathtt{u})$. For a neighbor-pair group between two sensors S_u and S_v, the group ID is set as $(\mathtt{u}, \mathtt{v})$, supposing $u < v$ is in its binary expression. For a class-based group corresponding to C_j, the group ID is set as $(\mathtt{C_j})$. For a location-based group at level i, the group ID is set as the ID(s) of the corresponding cell(s) at level i, with the common prefix suppressed. An example is shown in Fig. 2. For a location-based group at level 1 consisting of cells 222 and 223, we have its group ID as $(22-(2,3))$. Lastly, for a location-class-based group regarding C_j, its group ID is composed of C_j and the ID of the corresponding location-based group from which it derives. For a location-class-based group regarding C_j at level 1 consisting of cells 222 and 223, we have its group ID as $(22-(2,3), \mathtt{C_j})$. This indexing approach allows one to directly compare the size of different groups from their group IDs and support efficient location-based message forwarding, as will be shown shortly.

D. Key Setup

GPLD initializes each sensor with the GKEKs corresponding to the elementary groups it belongs to during the bootstrapping phase. GPLD adopts a robot-assisted network bootstrapping technique [33]. We assume that a group of mobile robots is dispatched to sweep across the whole sensor field along preplanned routes after the deployment of sensors. Mobile robots have Global Positioning System capabilities, as well as more powerful computation and communication capacities than ordinary sensors. The leading robot is also equipped with the network master secret key K. The robots securely localize every sensor using the secure localization protocol given in [3]. For a sensor S_u of class C_j with its level-1 *home cell* as $a_{L-1}\cdots a_i \cdots a_1 (a_i \in \{1,2,3,4\}, i = 1, \ldots, L-1)$, six GKEKs corresponding to the elementary groups to which it belongs are loaded.

1) *Broadcast key* (BCK): Corresponding to ('all'), $\mathtt{BCK} = H(K|0|K)$, where "|" denotes a concatenation operation and $H()$ denotes a cryptographically secure hash function such as SHA-1 [15].
2) *Individual key* (IDK): Corresponding to $(\mathtt{sink}, \mathtt{u})$, $\mathtt{IDK} = H(K|\mathtt{u}|K)$. IDK is known only to S_u and the sink.
3) A set of *pairwise keys* (PWKs): For every pair between S_u and its immediate neighbors, there is a PWK. Corresponding to $(\mathtt{u}, \mathtt{v})$ formed by S_u and a neighbor S_v, $\mathtt{PWK}_{u,v} = H(K|\mathtt{u}|\mathtt{v}|K)$, assuming $\mathtt{u} < \mathtt{v}$.
4) *Class key* (CLK): Corresponding to $(\mathtt{C_j})$, $\mathtt{CLK} = H(K|C_j|K)$.
5) A set of *location-aware keys* (LAKs): At each level, S_u belongs to all the groups that involve S_u's *home cell* at that level. There are a total of seven such groups at each level. The corresponding group IDs and LAKs at level i are

$$\mathtt{a}_{L-1}\cdots\mathtt{a}_{i+1} - (\mathtt{a}_i):$$
$$\mathtt{LAK}^{a_i}_{a_{L-1}\cdots a_{i+1}} = H(K|a_{L-1}\cdots a_{i+1}a_i|K)$$
$$\mathtt{a}_{L-1}\cdots\mathtt{a}_{i+1} - (\mathtt{a}_i, \mathtt{a}'_i):$$
$$\mathtt{LAK}^{a_i,a'_i}_{a_{L-1}\cdots a_{i+1}} = H(K|a_{L-1}\cdots a_{i+1}a_i|K|a_{L-1}\cdots a_{i+1}a'_i|K) \quad \forall a'_i \in \{1,2,3,4\}\setminus a_i$$
$$\mathtt{a}_{L-1}\cdots\mathtt{a}_{i+1} - (\mathtt{a}_i, \mathtt{a}'_i, \mathtt{a}''_i):$$
$$\mathtt{LAK}^{a_i,a'_i,a''_i}_{a_{L-1}\cdots a_{i+1}} = H(K|a_{L-1}\cdots a_{i+1}a_i|K|a_{L-1}\cdots a_{i+1}a'_i|K|a_{L-1}\cdots a_{i+1}a''_i|K) \quad \forall a'_i, a''_i \in \{1,2,3,4\}\setminus a_i.$$

Here, the sequence of the concatenation depends on the actual values of a_i, a'_i, and a''_i, and $a_i \neq a'_i \neq a''_i$. An example is shown in Fig. 2(b), where seven location-based elementary groups at level 1 to which S_u belongs are shown, assuming that S_u's *home cell* is $a_3a_2a_1 = 222$. The corresponding group IDs and LAKs are

$$22-(2): H(K|222|K)$$
$$22-(1,2): H(K|221|K|222|K)$$
$$22-(2,3): H(K|222|K|223|K)$$
$$22-(2,4): H(K|222|K|224|K)$$
$$22-(1,2,3): H(K|221|K|222|K|223|K)$$
$$22-(1,2,4): H(K|221|K|222|K|224|K)$$
$$22-(2,3,4): H(K|222|K|223|K|224|K).$$

The number of LAKs is $7*(L-1)$ for every sensor.
6) A set of *location-class keys* (LCKs): For each location-based group to which S_u belongs, S_u also belongs to the corresponding location-class-based group defined for class C_j sensors, and an LCK is derived from the corresponding LCK as follows: $\mathtt{C_j} - \mathtt{LCK} = H(K|\mathtt{LAK}|C_j|K)$. For example, $\mathtt{C_j} - \mathtt{LCK}^{1,2}_{22} = H(K|\mathtt{LAK}^{1,2}_{22}|C_j|K)$. Clearly, the number of LCKs for S_u is also $7*(L-1)$.

In addition to GKEKs, each sensor is also loaded with $\langle (x_0, y_0), L, len \rangle$ and the locations of the sensors in its level-1 *home cell* and all eight neighboring level-1 cells. In summary, the storage overhead per node is $\mathbb{O}(L + d)$, where d is the number of neighbors per node.

Note that the authentication between the sensors and the leading robot can easily be achieved using the technique introduced in [30]. We omit it here because of space limitations. By the end of the bootstrapping phase, the mobile robots leave the sensor field, and the leading robot should securely erase all the keys from its memory but should report the locations of the sensors to the sink. The assumption underlying this approach is that adversaries do not launch active and explicit pinpoint attacks on mobile robots at this stage, which usually does not last too long. That is, the robots are not likely subjected to compromise. We further note that the aforementioned bootstrapping operation can also be realized through the key predistribution approach [7], [8], instead of using mobile robots. In this case, the sensor nodes utilize secure localization protocols [12], [25], [32] to obtain their locations. The choice of the approaches depends on their availabilities in practice.

V. GPLD: Operation

In this section, we illustrate how fresh group keys and key update keys can efficiently be distributed using the *global-partition, local-diffusion* technique.

A. Notation

$\mathbb{W}$	All network sensors, except for the revoked sensors.
$\mathbb{N}$	All the *recipient sensors* of a multicast/rekeying session.
$\mathbb{R}$	All the revoked sensors in a rekeying session.
$\mathbb{S}_u$	Set of all immediate (nonrevoked) neighbor sensors of a sensor S_u.
$\mathbb{E}$	Elementary group.
$\mathtt{K_g}$	Fresh group key of a multicast session.
$\mathtt{K_{upd}}$	Fresh key refresh key of a rekeying session.
$\overline{\mathbb{S}}_u$	(Sub)set of $\mathbb{S}_u$ that contains only those *recipient sensors* yet to obtain $\mathtt{K_g}$ or $\mathtt{K_{upd}}$ in a multicast/rekeying session.
`Msg`	To-be-sent message.
`Hdr`	Header attached to a to-be-sent message.
'`Revocation`'	Revocation notice in plaintext.

B. Multicast Operation

To ensure security strength, GPLD requires the sink to generate a fresh group key to encrypt the to-be-sent message in each multicast session. For this purpose, the sink attaches a header to the message, which includes the specifications of the multicast group and the keying materials that enable the *recipient sensors* to recover the group key.

1) Group Description: As GPLD allows dynamic formation of multicast groups to support the various multicast group semantics discussed in Section II, it is impossible for sensors to know in advance their memberships of a given multicast session. Hence, there has to be a group description mechanism. One way to do so is to list all the IDs of the *recipient sensors* in the message header. Another way, as in broadcast encryption schemes [14], is to list all the indices of keys that are used to encrypt the group key in the message header; if a sensor possesses one of the corresponding keys, it is a *recipient sensor* for the session. However, both methods are very costly in WSNs, because the resulted message header could be very long in both cases. Moreover, both methods implicitly entail the use of network-wide flooding to deliver the multicast messages to the *recipient sensors*, which is neither necessary nor efficient. Derived from the multicast group semantics discussed in Section II, GPLD, however, efficiently describes multicast groups using the location and/or class information of the *recipient sensors*. Since the sensors are always deployed in a discrete manner at a certain density, we can easily express location constraints in terms of basic geometric shapes, which can efficiently be expressed using simple mathematical representations. More importantly, this location-aware group description approach is naturally supported by efficient message delivery approaches such as geocast [10], [24] so that network-wide broadcast can be avoided.

2) Message Format: In GPLD, a multicast message contains two parts, i.e., the header and message body

$$\{\mathtt{Hdr}, E(\mathtt{K}_g, \mathtt{Msg})\}$$

where $E(K, \bullet)$ is a symmetric encryption algorithm, such as AES [16], that encrypts $\bullet$ with key K. `Hdr` further contains two fields: $\{\mathtt{Hdr} = \mathtt{Grp\_Spec}, \mathtt{GK\_Info}\}$. `Grp_Spec` contains the multicast group information so that each sensor can judge whether it is a *recipient sensor* of the session. $\mathtt{Grp\_Spec} = (\mathtt{Loc\_Info}, \mathtt{Cla\_Info})$, where `Loc_Info` is the description of the location constraints of $\mathbb{N}$, and `Cla_Info` is the class information of $\mathbb{N}$. Recall that $\mathbb{N}$ denotes the *recipient sensors* of a multicast session. $\mathtt{GK\_Info}$ contains the encrypted $\mathtt{K_g}$ and the ID of the elementary group corresponding to the GKEK used for encryption.

3) Header Generation: In a multicast session, `Hdr` is generated as follows once $\mathbb{N}$ is determined:

1) Generate $\mathtt{Grp\_Spec} = (\mathtt{Loc\_Info}, \mathtt{Cla\_Info})$ according to the location and class constraints of $\mathbb{N}$.
2) Find the largest elementary group $\mathbb{E}$ with $\mathbb{E} \subseteq \mathbb{N}$; if there is a tie, select the elementary group that is closest to the sink.
3) Generate a fresh $\mathtt{K_g}$, and encrypt it with the GKEK corresponding to $\mathbb{E}$. $\mathtt{GK\_Info}$ thus contains the encrypted $\mathtt{K_g}$ and the group ID of $\mathbb{E}$.

4) Message Delivery: GPLD employs geocast to deliver multicast messages. By making use of the location-aware nature of WSNs, geocast utilizes a greedy forwarding for the packet delivery toward the target region. In greedy forwarding, a packet is forwarded to only one of the neighbor nodes whose geographical location is closest to the destination. As soon as the message reaches the target region, a restricted flooding (RF) or intelligent flooding technique [24] can be used to disseminate

the packet inside the target region. Specifically, the multicast message is delivered via a localized and hop-by-hop manner.

1) The sink uses greedy forwarding to deliver Hdr to the region taken up by $\mathbb{E}$.
2) As soon as Hdr reaches the target region, sensors in $\mathbb{E}$ that receive Hdr directly recover $\mathtt{K_g}$ from the attached GK˙Info.
3) Once having obtained $\mathtt{K_g}$, each *recipient sensor* S_u executes four steps.
   a) Determine whether it should diffuse the key to its neighbor *recipient sensors* based on the underlying multicast technique, the preloaded location information of the sensors in its neighboring level-1 cells, and Grp˙Spec. If not, proceed to step 4. If yes, proceed to step b.
   b) Find $\overline{\mathbb{S}}_u$ (refer to Section V-A) out of $\mathbb{S}_u$. For every member of $\overline{\mathbb{S}}_u$, find the largest elementary group $\mathbb{E}_i$ it belongs to (based on S_u's own location knowledge), where $\mathbb{E}_i \subseteq \mathbb{N}$. If there is a tie, select the elementary group that S_u belongs to (if applicable); otherwise, randomly select one.
   c) For every found $\mathbb{E}_i$, if $S_u \in \mathbb{E}_i$, encrypt $\mathtt{K_g}$ with the GKEK corresponding to $\mathbb{E}_i$; if $S_u \notin \mathbb{E}_i$, pick up one member from $\overline{\mathbb{S}}_u \cap \mathbb{E}_i$, and encrypt $\mathtt{K_g}$ with the PWK shared between S_u and the selected member.
   d) Replace GK˙Info with the encryptions of $\mathtt{K_g}$ obtained in step c and the group IDs corresponding to the GKEKs used for encryption. Locally broadcast the updated Hdr.
4) The sink further uses greedy forwarding to deliver $E(\mathtt{K_g}, \mathtt{Msg})$ to the region taken up by $\mathbb{N}$. As soon as $E(\mathtt{K_g}, \mathtt{Msg})$ reaches the target region, a sensor in $\mathbb{N}$ that receives it determines whether to diffuse it in the neighborhood based on the underlying routing strategy such as RF or intelligent flooding [24]. If yes, S_u locally rebroadcast $E(\mathtt{K_g}, \mathtt{Msg})$.
5) Finally, every *recipient sensor* recovers Msg using the obtained $\mathtt{K_g}$ and deletes $\mathtt{K_g}$ in the end.

C. Examples

The two examples shown in Fig. 3 illustrate a location-class-based multicast session at time T1 and a location-based multicast session at time T2, respectively. In the former session, the multicast group happens to be an elementary group. That is, $\mathbb{N}$ is the set of class "Δ" sensors located inside Rec consisting of cells 11 and 12, i.e., the group $(1-(1,2),\text{'}\Delta\text{'})$, and Rec is the rectangle function. Hence, $\mathtt{Grp˙Spec} = (\mathtt{Loc˙Info} : \mathtt{Rec}, \mathtt{Cla˙Info} : \text{'}\Delta\text{'})$. According to the header generation algorithm, $\mathtt{GK˙Info} = (E(\mathtt{K_g}, \text{'}\Delta\text{'} - \mathtt{LCK}_1^{1,2}), (1-(1,2),\text{'}\Delta\text{'}))$. The sink then uses greedy forwarding to send Hdr to the closest *recipient sensor* in Rec. Next, $\mathtt{K_g}$ is securely diffused among $\mathbb{N}$ according to message delivery step 3. In this example, if a *recipient sensor* determines that it should diffuse $\mathtt{K_g}$, it simply locally rebroadcasts Hdr.

In the latter session, $\mathbb{N}$ is the set of sensors located inside Elp, and Elp is the corresponding elliptic curve. Here, $\mathtt{Grp˙Spec} = (\mathtt{Loc˙Info} : \mathtt{Elp}, \mathtt{Cla˙Info} : \text{'all'})$. $\mathtt{GK˙Info} = (E(\mathtt{K_g}, \mathtt{LAK}_4^{1,3}), (4-(1,3)))$. Again, the sink uses greedy forwarding to send Hdr to the closest *recipient sensors* in cells 41 and 43. Then, $\mathtt{K_g}$ is securely diffused among $\mathbb{N}$. According to the scheme, $\mathtt{K_g}$ is securely diffused inside each shadow area (i.e., each corresponding location-based group) by using the corresponding LAK, respectively. For instance, inside $(3-(2))$, $\mathtt{K_g}$ is encrypted using $\mathtt{LAK}_3^2$. Furthermore, $\mathtt{K_g}$ is securely diffused from one elementary group to another using a GKEK shared between the sender sensor in the former group and the receiver sensor in the latter. For instance, between $(4-(1,3))$ and $(14-(3,4))$, $\mathtt{K_g}$ is securely diffused from sensor S_u to S_v after being encrypted with $\mathtt{PWK}_{u,v}$; between $(4-(1,3))$ and individual sensor S_z, $\mathtt{K_g}$ is securely diffused from S_u to S_z after being encrypted with $\mathtt{PWK}_{u,z}$.

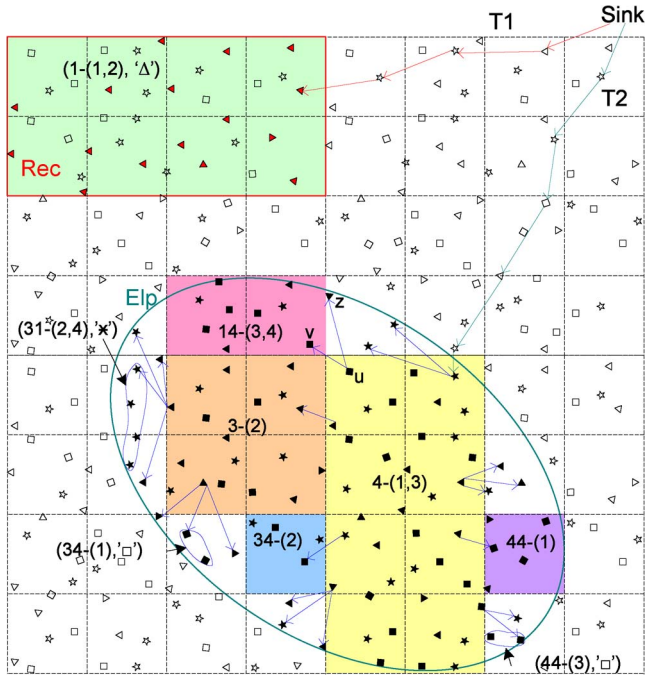

Fig. 3. Two exemplary multicast sessions, where each solid symbol denotes a *recipient sensor*, each shadowed area denotes one location-based elementary group, and each of the three irregular circled area denotes a location-class-based elementary group.

D. Rekeying Operation

Once compromised sensors are detected, all the GKEKs they possess should be either obsoleted or securely refreshed in such a way that no compromised sensor could do so, even by colluding. Thus, all subsequent multicast communications can be kept secret from the revoked sensors. GPLD supports both on-demand and batched (periodical) rekeying strategies. Suppose that r compromised sensors, i.e., $\#\{\mathbb{R}\} = r$, are to be excluded in a rekeying session, where r is usually a small number. The rekeying operation works as follows:

1) Sink:

1) Find the largest location-based elementary group $\mathbb{E}$, where $\mathbb{E} \subseteq \mathbb{N} = \mathbb{W}$; if there are multiple sets of the same cardinality, select the elementary group that is closest to the sink.

2) Generate a fresh $\mathtt{K_{upd}}$, and encrypt it with the LAK corresponding to $\mathbb{E}$.
3) Generate the rekeying message containing the following information: 1) the IDs of revoked sensors; 2) the encrypted $\mathtt{K_{upd}}$; 3) the group ID of $\mathbb{E}$; and 4) $E(\mathtt{K_{upd}}, \text{`}\mathtt{Revocation}\text{'})$, i.e., the encrypted revocation notice.
4) Geocast the rekeying message to $\mathbb{E}$.

2) Sensors (Except for the Revoked Sensors):

1) Diffuse $\mathtt{K_{upd}}$ according to the same approach described for multicast operation (refer to Section V-B).
2) Perform a key-refreshing operation. For every GKEK held by each sensor (except for the IDK and PWKs), $\mathtt{GKEK} = H(\mathtt{K_{upd}}|\mathtt{GKEK}|\mathtt{K_{upd}})$.
3) Delete $\mathtt{K_{upd}}$; delete the *revoked sensors* from $\mathbb{S}_u$ and the PWKs shared with them, if any.

Hence, after the rekeying operation, all the GKEKs held by the *revoked sensors* are now obsoleted and are therefore permanently excluded from the WSN.

Due to the instability of wireless communication, it is possible that a sensor misses some rekey messages and thus fails to receive later multicast messages. One solution is that the sensor asks for help from its neighbors. Upon receiving the request, the neighbor first verifies that the sensor is not on the revocation list and then sends a message containing all the GKEKs that they should share according to the grouping policy presented in Section IV-C. This message can be encrypted with the PWK between these two nodes, which may have been established in other fundamental security protocols. In the case in which such PWKs are not available, we rely on the use of mobile robots to help such sensors obtain the latest GKEKs.

VI. Security Analysis

1) Correctness: The correctness of GPLD derives from the following facts: First, no *revoked sensors* excluded from the WSN can refresh the GKEKs they hold after revocation. This is true, because the *revoked sensors* can never obtain a $\mathtt{K_{upd}}$ using their GKEKs. Since the status of the system is reinstated to its original setting after every rekeying, we only need to consider the possible security issues that arise during a single rekeying operation. There are only two ways for a sensor to obtain a $\mathtt{K_{upd}}$ in a rekeying session. That is, a sensor recovers a $\mathtt{K_{upd}}$ by either directly decrypting the rekeying message sent by the sink or indirectly receiving it from a neighbor *recipient sensor*, which encrypts $\mathtt{K_{upd}}$ with a GKEK shared between the two and known only to the *recipient sensors*. However, neither way can be exploited by the *revoked sensors*. A *revoked sensor* cannot recover $\mathtt{K_{upd}}$, because it has no corresponding GKEKs; at the same time, its neighbor sensors will not send it the key, as its ID is explicitly listed in the rekeying message. Without $\mathtt{K_{upd}}$, it is computationally infeasible for a *revoked sensor* to refresh its GKEKs due to the underlying cryptographically secure hash function used. Consequently, the *revoked sensors* can never recover the group keys of the multicast sessions after their revocation, due to the obsoleteness of their GKEKs.

Second, the *recipient sensors* can always verify the correctness of the update keys and group keys they obtain for the following reasons: 1) The authenticity of the rekeying and multicast messages and, hence, that of $E(\mathtt{K_{upd}}, \text{`}\mathtt{Revocation}\text{'})$ and $E(\mathtt{K_g}, \mathtt{Msg})$ can always be guaranteed through authentication schemes such as $\mu$TESLA [19]. 2) Both '`Revocation`' and `Msg` follow a certain predefined format and are meaningful. Therefore, by decrypting $E(\mathtt{K_{upd}}, \text{`}\mathtt{Revocation}\text{'})$ and $E(\mathtt{K_g}, \mathtt{Msg})$ and verifying the validity of the recovered '`Revocation`' and `Msg`, the correctness of the received $\mathtt{K_{upd}}$ and $\mathtt{K_g}$ can further be verified.

Finally, GPLD allows all *recipient sensors* in a rekeying/multicast session to securely obtain the corresponding $\mathtt{K_{upd}}$ or $\mathtt{K_g}$. That is, no sensor can be excluded from the session in GPLD, as long as it is physically reachable. In the worst case, a sensor can always be updated through the IDK it shares with the sink.

2) Compromise Resilience: Since sensor compromise is unavoidable when the WSNs are deployed in hostile environments, it is crucial to minimize the resulted security risk. Ideally, after a sensor is compromised and before its revocation, the keying information that it possesses should only allow the adversary to compromise those multicast messages, of which it is a legitimate *recipient sensor*; all other messages should still be kept secure against the adversary. That is, the security of a multicast message is broken only if at least one of the corresponding *recipient sensors* is compromised and yet revoked. GPLD achieves this full security strength for all four multicast group semantics discussed in Section II, because of the following reasons: 1) A fresh key is always generated in each different rekeying/multicast session. 2) The fresh key is securely diffused among the *recipient sensors*, which are always encrypted with the GKEKs that are known only to the *recipient sensors*.

3) Other Attacks: We assumed that the adversary may eavesdrop on all traffic, inject packets, or replay old packets. Because the sink authenticates all the rekeying/multicast messages by μTESLA [19], no sensors can inject any fake messages into the WSN or modify any messages they forward while impersonating the sink. The adversary cannot also replay old rekeying packets because of time-stamp information used in μTESLA. The adversary may also want to launch refusal-of-service attacks, such as dropping the packets and jamming the network.[3] However, *revoked sensors* normally do not help the adversary drop the packets, because all the *revoked sensors* have already been excluded from the WSN, i.e., no traffic is going through them. The worst situation caused by such attacks is hence equivalent to that due to packet losses. One salient property of GPLD is that it allows a sensor to miss certain multicast sessions without affecting its ability to participate any future multicast session, as long as it does not miss any rekeying operation. Therefore, GPLD is also resilient to such attacks.

VII. Performance Analysis and Simulation

In this section, the performance of GPLD is analyzed. We mainly focus on the communication cost of GPLD, as

[3]Such attacks are always possible and are not specific to multicast encryption schemes. Mechanisms dealing with such attacks can be found in [27].

it is the most significant factor of energy consumption in WSNs. The computation and storage cost of GPLD are also discussed.

A. Communication Overhead

1) Models for Lower Bound and No-Design Cases: The lower bound of the communication overhead happens in the ideal situation, where a different elementary group is established for each possible combination of network sensors; each sensor stores all the GKEKs for the groups in which it is involved, which are up to $2^{\#\{\mathbb{W}\}-1}$ keys. In this ideal situation, every multicast group is an elementary group. Hence, to securely diffuse a message among the *recipient sensors*, a single GKEK corresponding to the elementary group is sufficient. On the contrary, when there are no predistributed keys for the elementary groups, except for the PWKs existing between neighbor pairs, the multicast/rekeying message has to be encrypted using various PWKs at each step of the diffusion. This is the typical setting provided by most key management schemes designed for WSNs [6], [8], without involving any designs for the purpose of multicast encryption.

In the simulation, we adopt the two aforementioned models as the bases for analyzing and comparing the communication overhead of GPLD. We denote the two models, in which the diffusion of messages is achieved through a single GKEK (i.e., the lower bound case) and through only PWKs, as the "LB model" and the "PWKD model," respectively. Hence, including GPLD, three models are simulated here.

2) Simulation Settings and Evaluation Metrics: The communication overhead of a multicast/rekeying session in GPLD consists of two parts: 1) the cost to unicast the multicast/rekeying message to the largest elementary group of the *recipient sensors* and 2) the cost to locally diffuse the message among the *recipient sensors*. Since the former is relatively small and is the same for all the models, only the latter is considered in the simulation. A multicast/rekeying message in GPLD contains the header and message body. We do not analyze the cost spent on the message body since this cost is independent from the multicast encryption scheme. Instead, we focus on the header part, which contains two parts: 1) the description of the multicast group or revoked sensors and 2) the keying materials. While the size of part 1 is usually small and is identical in all the models, part 2 dominates the communication overhead of a multicast/rekeying session and may greatly vary in length.

Consequently, in the simulation, we use the total number of keys sent or forwarded by all the unrevoked sensors as the metric of evaluating the communication cost. Note that, in the case that a sensor sends/forwards the fresh group/update key to its neighbors using PWKs, we count the number of keys sent as the number of its neighbors. This sensor may put all the encrypted key materials in one message, but this will increase the length of the key materials transmitted anyway.

In the simulation, there are 10 000 sensors randomly distributed in the network, the size of which is 3000×3000 m. The transmission range denoted as tr is 100 or 135 m, which corresponds to 36 or 64 neighbors per sensor, respectively. For each setting, we run the simulation for 100 times and calculate the average values. Two routing strategies are simulated. One is RF, where each sensor broadcasts any message received once using the key according to the largest elementary group to which it and all or part of its neighbors belong, and for those neighbors that only share PWKs with this sensor, it will send the message to them individually. In the other strategy called *once-per-cell* (OPC), the same message is broadcast exactly once within any level-1 cell within the target region using a key corresponding to an elementary group that covers this level-1 cell, if any. If such a key does not exist, the message is diffused using PWKs. Since, in GPLD, we assume that the sensors in the same level-1 cell are always within the direct communication range of each other, the optimization can still ensure the successful transmission of fresh group/update keys.

TABLE I
COMPARISON OF MULTICAST COST UNDER DIFFERENT SETTINGS

	RF $tr = 100$ $L = 6$	RF $tr = 100$ $L = 7$	RF $tr = 100$ $L = 8$	OPC $tr = 135$ $L = 5$	OPC $tr = 135$ $L = 6$
LB	858.86	940.50	878.92	107.64	398.38
GPLD	1844.81	1349.26	1233.88	285.63	478.28
PWKD	27489.39	30188.34	28107.67	830.43	869.33

3) Multicast: Table I compares the communication cost of a multicast session under all the models. In the simulation, the multicast group consists of all the sensors within a randomly generated rectangle for simplicity. The lengths of the sides of the rectangle are uniformly chosen between 300 and 1500 m. As shown in Table I, not only is GPLD more efficient than the PWKD model under both RF and OPC but, by appropriately choosing L, its communication overhead is also only 20.06% and 40.39% more than the LB model under RF and OPC, respectively, with significantly less predistributed keys.

We also notice that, in RF, the multicast cost of GPLD can be decreased by increasing the number of levels of the quadtree, i.e., L. For example, the cost decreases by 26.86% when increasing L from 6 to 7. However, the advantage of further increasing L has recessive effects. When increasing L from 7 to 8, the cost decreases by only 8.55%. Therefore, we need to balance between the storage overhead and the communication overhead while selecting the optimal value of L. Table I also shows that by employing the optimal routing strategy (i.e., OPC[4]), the communication cost of GPLD can be decreased to only 286 when $L = 5$. Since OPC helps only when the number of sensors per level-1 cell is more than one, we only simulate the scenarios of $L = 5$ and $L = 6$.

To evaluate the effectiveness of GPLD under different sizes of multicast groups, we uniformly choose the length of the sides of the rectangle between l_R and $l_R + 200$ and increase l_R from 300 to 1300 m. Fig. 4 shows that GPLD is more effective when the size of a multicast group is large, because the larger the area

[4]Since, in GPLD, we assume that the sensors in the same level-1 cell are always within the direct communication range of each other, we cannot set $L = 5$ when $tr = 100$ m. Thus, to show the effectiveness of OPC under different L's, we simulate OPC under $tr = 135$ m.

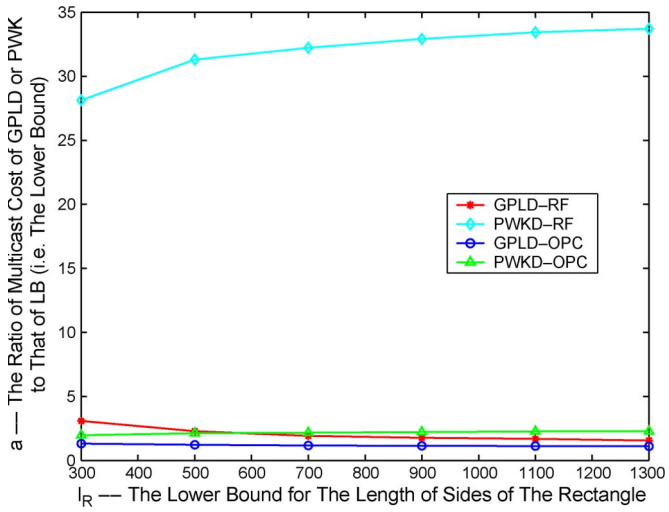

Fig. 4. Multicast cost (in terms of the ratio to the LB model) under different multicast group sizes and various routing strategies ($tr = 100$ m, and $L = 6$).

TABLE II
COMPARISON OF REKEYING COST UNDER DIFFERENT NUMBERS OF REVOKED SENSORS AND VARIOUS ROUTING STRATEGIES

	GPLD-RF	PWKD-RF	GPLD-OPC	PWKD-OPC
$r = 10$	1.007	61.127	1.015	2.442
$r = 20$	1.012	61.103	1.027	2.443
$r = 30$	1.017	61.017	1.038	2.444
$r = 40$	1.022	60.949	1.049	2.444
$r = 50$	1.027	60.884	1.060	2.445

that a multicast group covers, the higher the percentage that a fresh group/update key is encrypted by LAKs/LCKs, instead of PWKs during the diffusion. By employing the method for optimally choosing the LAKs/LCKs (refer to Section V-B), the diffusion using LAKs/LCKs is more efficient than that using PWKs. As a result, GPLD presents higher efficiency for larger multicast groups.

4) Rekeying: In the simulation, for each rekeying session, we randomly choose r revoked sensors from the network. Table II shows the rekeying cost of GPLD and the PWKD model (in terms of the ratio to that of the LB model) under different values of r and various routing strategies, when $tr = 135$ m and $L = 6$. Similar to multicast, GPLD is more efficient than the PWKD model. Moreover, the ratio of the rekeying cost of GPLD to that of the LB model is much smaller than the multicast case. The extra overhead of GPLD over the lower bound ranges from only 2.7% to 6%, because, given that the number of revoked sensors is small, in a rekeying session, the majority of diffusion messages are encrypted using LAKs/LCKs. More importantly, the simulation results also show that the performance of rekeying in GPLD is not sensitive to the increase in the number of revoked sensors. For example, when r increases from 10 to 50, the additional keying materials required are only around 160 under both RF and OPC. It is a significant advantage over other works. Other schemes (such as LKH and SD [14], [26]) either can only revoke one member per session or have the revocation cost (i.e., the number of keys broadcast to the whole network) at least linear to the number of revoked members.

B. Storage and Computation Overhead

1) Storage Overhead: In GPLD, a sensor stores the GKEKs corresponding to all the elementary groups it belongs to. Specifically, a sensor of class C_j belongs to the network-wide group, the individual group of itself, and the class-based group consisting of all class C_j sensors. Moreover, there are n' neighbor-pair groups defined for each sensor, where n' is the number of immediate neighbors that a sensor has. Additionally, each sensor also belongs to $7 * (L-1)$ location-based groups and $7 * (L-1)$ location-class-based groups (refer to Section IV-D). Therefore, there are a total of $1+1+1+n'+7(L-1)+7(L-1) = 14\ L+n'-11$ GKEKs that should be stored by each sensor. In a WSN, n' could usually range from 20 to 60, depending on different applications [6], [8], [23], while L is a system parameter of the grid. Recall that the sensors in a level-1 cell are within each others' direct communication range, as required in GPLD. Then, the number of sensors in a level-1 cell is in the range of around 4–10, given n' ranging from 20 to 60. Hence, for a WSN whose size is no more than 100 000, $L = 9$ will be more than enough to support GPLD, as there will be up to $4^{L-1} = 65\,536$ level-1 cells. Thus, each sensor stores at most 161 GKEKs. Suppose that each GKEK is 8 B, then 161 GKEKs require a storage space of 1.26 kB only. In addition, note that although the sink is required to know all the GKEKs, it does not have to directly store all of them. Instead, the sink could store only the master key and the locations of each sensor and compute the GKEK on the fly.

2) Computation Overhead: The computation overhead introduced by GPLD is lightweight, as each sensor is only required to perform several times of encryption and decryption operations over a very short message (i.e., one key). GPLD does not require sensors to perform any kind of expensive public-key or polynomial-based operations.

VIII. CONCLUSION AND FUTURE WORKS

In this paper, we have analyzed and classified the multicast group semantics for WSNs that are inherently demanded by most applications. We then proposed GPLD to address the multicast encryption problem in WSNs, which, to the best of our knowledge, is the first scheme of its kind that supports various multicast group semantics and is tailored for WSNs. Our proposed scheme advances the current state of the art by enabling not only the dynamic changing but the dynamic formation of multicast groups as well. We developed a novel multicast encryption technique called *global-partition, local-diffusion* to achieve scheme efficiency and meet the resource-constrained nature of WSNs. The security and performance of the proposed scheme are justified through both analysis and simulations.

In this paper, to help bootstrap the network, we assume the existence of mobile robots. In reality, such help may be unavailable or ineffective (e.g., a hostile terrain). Therefore, it is desirable to develop a secure multicast protocol without the robot-based assistance. Another possible future work is to extend GPLD to support the scenarios where bidirectional communication cannot be guaranteed.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast security: A taxonomy and some efficient constructions," in *Proc. IEEE INFOCOM*, 1999, pp. 708–716.

[3] S. Capkun and J. P. Hubaux, "Secure positioning in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 221–232, Feb. 2006.

[4] A. Chadha, Y. Liu, and S. Das, "Group key distribution via local collaboration in wireless sensor networks," in *Proc. IEEE SECON*, 2005, pp. 46–54.

[5] H. Chan and A. Perrig, "Security and privacy in sensor networks," *Comput.*, vol. 36, no. 10, pp. 103–105, Oct. 2003.

[6] H. Chan and A. Perrig, "PIKE: Peer intermediaries for key establishment in sensor networks," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 524–535.

[7] W. Du, J. Deng, Y. Han, and P. Varshney, "A pairwise key predistribution scheme for wireless sensor networks," in *Proc. ACM CCS*, 2003, pp. 42–51.

[8] L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. ACM CCS*, Washington, DC, Nov. 2002, pp. 41–47.

[9] C. Intanagonwiwat, R. Govindan, D. Estrin, and J. Heidemann, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, Feb. 2003.

[10] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. ACM MOBICOM*, Boston, MA, Aug. 2000, pp. 243–254.

[11] L. Lazos and R. Poovendran, "Energy-aware secure multicast communication in ad-hoc networks using geographic location information," in *Proc. IEEE ICASSP*, 2003, pp. 201–204.

[12] L. Lazos and R. Poovendran, "SeRLoc: Secure range-independent localization for wireless sensor networks," in *Proc. ACM WiSe*, Oct. 2004, pp. 21–30.

[13] K. Lorincz, D. Malan, T. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, S. Moulton, and M. Welsh, "Sensor networks for emergency response: Challenges and opportunities," *Pervasive Comput.*, vol. 3, no. 4, pp. 16–23, Oct.–Dec. 2004.

[14] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proc. CRYPTO*, 2001, pp. 41–62.

[15] "Digital hash standard," *Federal Information Processing Stand. Pub. 180-1*, 1995.

[16] NIST, *Fips-197: Advanced Encryption Stand.*, 2001.

[17] S.-J. Park, R. Vedantham, R. Sivakumar, and I. F. Akyildiz, "A scalable approach for reliable downstream data delivery in wireless sensor networks," in *Proc. ACM MOBIHOC*, Tokyo, Japan, May 2004, pp. 78–89.

[18] A. Perrig, D. Song, and D. Tygar, "ELK, a new protocol for efficient large-group key distribution," in *Proc. IEEE SP*, May 2001, pp. 247–262.

[19] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. Culler, "SPINS: Security protocols for sensor networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 521–534, Sep. 2002.

[20] R. Di Pietro, L. Mancini, Y. Law, S. Etalle, and P. Havinga, "LKHW: A directed diffusion-based secure multicast scheme for wireless sensor networks," in *Proc. ICPPW*, 2003, pp. 397–406.

[21] K. Ren, W. Lou, and Y. Zhang, "LEDS: Providing location-aware end-to-end data security in wireless sensor networks," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–12.

[22] K. Ren, W. Lou, and Y. Zhang, "Multi-user broadcast authentication in wireless sensor networks," in *Proc. 4th Annu. IEEE Commun. Soc. Conf. SECON*, San Diego, CA, Jun. 2007, pp. 223–232.

[23] K. Ren, K. Zeng, and W. Lou, "A new approach for random key predistribution in large-scale wireless sensor networks," *Wirel. Commun. Mob. Comput.—Special Issue on Wireless Networks Security*, vol. 6, no. 3, pp. 307–318, May 2006.

[24] I. Stojmenovic, "Geocasting with guaranteed delivery in sensor networks," *IEEE Wireless Commun.*, vol. 11, no. 6, pp. 29–37, Dec. 2004.

[25] S. Čapkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proc. IEEE INFOCOM*, 2005, pp. 1917–1928.

[26] C. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," in *Proc. ACM SIGCOMM*, 1998, pp. 68–79.

[27] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. ACM MobiHoc*, 2005, pp. 46–57.

[28] S. Yu, K. Ren, and W. Lou, "Attribute-based on-demand multicast group setup with membership anonymity," in *Proc. Securecomm*, Istanbul, Turkey, 2008.

[29] W. Zhang and G. Cao, "Group rekeying for filtering false data in sensor networks: A predistribution and local collaboration-based approach," in *Proc. IEEE INFOCOM*, 2005, pp. 503–514.

[30] W. Zhang, H. Song, S. Zhu, and G. Cao, "Least privilege and privilege deprivation: Towards tolerating mobile sink compromises in wireless sensor networks," in *Proc. ACM MOBIHOC*, May 2005, pp. 378–389.

[31] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proc. ACM MOBICOM*, 2000, pp. 275–283.

[32] Y. Zhang, W. Liu, Y. Fang, and D. Wu, "Secure localization and authentication in ultra-wideband sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 4, pp. 829–835, Apr. 2006.

[33] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 247–260, Feb. 2006.

[34] S. Zhu, S. Setia, S. Xu, and S. Jajodia, "GKMPAN: An efficient group rekeying scheme for secure multicast in ad-hoc networks," in *Proc. ACM Mobiquitous*, Boston, MA, Aug. 2004, pp. 42–51.

Kui Ren (M'08) received the B.Eng. and M.Eng. degrees from Zhejiang University, Hangzhou, China, in 1998 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from Worcester Polytechnic Institute, Worcester, MA, in 2007.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago. He was a Research Assistant with Shanghai Institute of Microsystems and Information Technology, Chinese Academy of Sciences, Shanghai, China, from March 2001 to January 2003; the Institute for Infocomm Research, Singapore, from January to August 2003; and the Information and Communications University, Daejeon, Korea, from September 2003 to June 2004. His research interests include ad hoc/sensor network security, wireless mesh network security, Internet security, and security and privacy in networks and systems.

Wenjing Lou (S'01–M'03–SM'08) received the B.E. and M.E. degrees in computer science and engineering from Xi'an Jiaotong University, Xi'an, China, in 1993 and 1996, respectively, the M.A.Sc. degree from Nanyang Technological University, Singapore, in 1998, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, in 2003.

From December 1997 to July 1999, she was a Research Engineer with the Network Technology Research Center, Nanyang Technological University. In 2003, she joined the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute (WPI), Worcester, MA, where she is currently an Assistant Professor. Her current research interests include ad hoc, sensor, and mesh networks, with emphasis on network security and routing issues.

Dr. Lou has been an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS since 2007. She was the recipient of the U.S. National Science Foundation Faculty Early Career Development (CAREER) Award in 2008. He was named Joseph Samuel Satin Distinguished Fellow in 2006 by WPI.

Bo Zhu (M'06) received the B.Eng. and M.Eng. degrees from Wuhan University, Wuhan, China, in 1996 and 1999, respectively, and the M.Sc. and Ph.D. degrees from the National University of Singapore, Singapore, in 2002 and 2006, respectively.

He was a Postdoctoral Researcher with the Center for Secure Information Systems, George Mason University, Fairfax, VA, for two years. In 2007, he joined Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada, as an Assistant Professor. His research interests include security and privacy issues in various types of networks (including ad-hoc/sensor/peer-to-peer/wireless networks, Internet, LAN, and grids), information and system security (such as database security and multimedia security), and digital signatures with specific properties.

Sushil Jajodia (M'88–SM'88) received the Ph.D. degree from the University of Oregon, Eugene.

He was the Head of the Database and Distributed Systems Section, Naval Research Laboratory, Washington, DC. He then became the Director of the Database and Expert Systems Program of the National Science Foundation. He is currently a University Professor, the BDM International Professor of Information Technology, and the Director of the Center for Secure Information Systems, George Mason University (GMU), Fairfax, VA. He has also been a Visiting Professor with the University of Milan, Milan, Italy; the University of Rome "La Sapienza," Rome, Italy; and the Isaac Newton Institute for Mathematical Sciences, Cambridge University, Cambridge, U.K. He has authored six books and more than 300 technical papers in refereed journals and conference proceedings and edited 28 books and conference proceedings. He has served in different capacities for various journals and conferences. He is the founding Editor-in-Chief of the *Journal of Computer Security* and serves on the Editorial Boards of the *IEE Proceedings on Information Security*, the *International Journal of Cooperative Information Systems*, the *International Journal of Information and Computer Security*, and the *International Journal of Information Security and Privacy*. His research interests include information secrecy, privacy, integrity, and availability problems in the military, civil, and commercial sectors.

Dr. Jajodia was the recipient of the 1996 Kristian Beckman Award from IFIP TC 11 for his contributions to the discipline of information security, the 2000 Outstanding Research Faculty Award from GMU's School of Information Technology and Engineering, and the 2008 ACM SIGSAC Outstanding Contributions Award for his research and teaching contributions to the information security field and his service to ACM SIGSAC and the computing community.