Location-Aware Secure Wireless Multicast in Ad-Hoc Networks under Heterogeneous Path-loss

Loukas Lazos and Radha Poovendran
{l_lazos, radha}@ee.washington.edu
Network Security Lab, Dept. of EE,
University of Washington, Seattle, WA

Gregory H. Cirincione
crincione@arl.army.mil
U.S. Army Research Laboratory
Adelphi, MD

Abstract

In wireless ad hoc networks, energy expenditure is a major performance bottleneck as almost all nodes rely on batteries. When identical data has to be sent to multiple receivers, a group communication model reduces the network communication cost, and hence, saves energy resource. In this paper we address the problem of minimizing the communication overhead associated with realizing secure group communications in energy-constrained wireless ad hoc networks. We observe that the communication overhead, defined as number of messages in wired networks, is not a suitable measure of the energy expenditure in wireless networks and introduce a novel metric to incorporate the energy parameter. We argue that great energy savings can occur when node location information is exploited in the construction of the key management scheme. We develop a location-aware key management scheme that takes into account the heterogeneity of the medium and vastly improves the energy efficiency. We present simulation studies that demonstrate the improvements achieved using our algorithms.

1 Introduction

The idea of many personal devices being able to communicate over a short range without the need for a predeployed network infrastructure has recently become feasible. Wireless ad hoc networks have many civilian applications such as dynamic communication for emergency-rescue operations and disaster-relief efforts, patient monitoring and drug inventory management in hospitals, as well as military applications such as deployment of surveillance networks and real time information distribution for mobile army operations. Almost all applications rely on the capability of peer-to-peer collaboration in computation and/or communications.

Most of the devices rely onbatteries and therefore are constrained in computational power and communication capability. Recent advances in microprocessor technology have significantly reduced the energy consumption for performing computations and information processing. However, when small devices are wirelessly networked, energy consumption due to communications energy, becomes the dominant factor in battery depletion [1], [2]. Hence, the communication model and the employed algorithms should minimize the energy consumption due to communication.

Securing the communication channel is a critical requirement in most multicast applications. Secure multicast communications can be realized through encrypting the information that is transmitted over the open wireless channel. Additionally, cryptography can support group access control for dynamic multicast groups through secure management of the cryptographic keys [3]. The *key management* problem is to ensure that only legitimate members hold valid keys at any time. The key management problem can be reduced to the *key distribution* problem, which involves the secure and efficient distribution of the cryptographic keys to valid members.

Reducing the network management overhead involved in realizing security in ad hoc wireless networks is essential for reducing energy consumption. In [4], we developed a key distribution scheme for wireless ad hoc networks deployed in a homogeneous medium, where the path loss attenuation factor is be constant. In this paper, we examine the more realistic case where the wireless medium is heterogeneous (varying path loss factor), for which solutions in [4] are inadequate.

We observe that the communication overhead associated with the realization of security has to be measured in energy units rather than number of control messages, used in wired networks. We introduce a new metric called *average update energy* for measuring the communication overhead involved in securing group communications. We show that when using our new metric the communication overhead of a key distribution scheme is dependent on network topology. To incorporate the network topology and heterogeneity of the medium in the construction of energy-efficient key tree hierarchies, we will use clustering techniques that can handle multiple dissimilarity measures simultaneously and achieve great energy savings.

The remainder of the paper is organized as follows. In Section 2 we present relevant background. In Section 3 we describe the network model assumed and state the problem to be addressed. In Section 4 we introduce a new

performance evaluation metric for the key management communication overhead. In Section 5 we present prior work for the homogeneous medium. In Section 6 we develop a location-aware key distribution scheme for networks deployed in a heterogeneous medium. In Section 7 we provide simulation results and show the improvements achieved by our algorithms. In Section 8 we present related work. In Section 9 we describe open problems and future work and in Section 10 we conclude.

2 Background

2.1 Notation

The following notation will be used through the rest of the paper.

N	Multicast group size.
$d_{i,j}$	Distance between nodes i,j.
$P(d_{i,j})$	Transmit power needed for communication between node i and j at distance $d_{i,j}$.
$PL(d_{i,j})$	Path loss for communicating between i and j at distance $d_{i,j}$.
$diss(i,j)$	Dissimilarity between i and j.
γ	Path loss attenuation factor.
M_i	The i_{th} member of the multicast group.
T	Key distribution tree.
l	Level of a key distribution tree T.
δ	Degree of a key distribution tree T.
$K_{l.j}$	Key assigned to the j_{th} node of the l_{th} level of the key distribution tree T. The root of the tree has level $l = 0$.
$\{m\}_{K_{l.j}}$	Message m encrypted with key $K_{l.j}$.
GC	Group controller of the multicast group.
R	Underlying routing tree.
S	Subset of members of the multicast group.
$A \rightarrow S : m$	A sends a message m to all members of subset S.

2.2 A quick review on tree-based group key management

Encryption methods that use public-key algorithms such as Diffie-Hellman key agreement [5] or RSA signatures [6] have high energy consumption due to the increased computational load that is sometimes even beyond the capability of the devices. Cryptographic methods for energy constrained wireless sensor networks were evaluated in [2], [7]. On the other hand, symmetric cryptography reduces the required computations and energy for performing encryption. By using a single key called Session Encryption Key (SEK) only known to legitimate group members, the sender performs one encryption (in broadcast mode) and every user performs one decryption per message, thus reducing the computational and communication overhead. However, the use of a single key known to all members requires its update through additional keys called Key Encryption Keys (KEK), each time a membership change occurs, in order to provide backward and forward traffic protection [8].

When the secure communications involve large dynamic groups where frequent membership changes occur, the key management/distribution scheme needs to be scalable with the group size. The number of keys updated after a member leave are significantly higher than the updated keys after a member join [8], [9]. Hence, key management schemes mainly address the overhead of a member (or multiple members) leave rather than join.

The most efficient group key management techniques that have been proposed for group communication in wired networks are relying upon logical key trees [8], [9], [10], [3], [11]. Logical key tree based schemes reduce the complexity of rekeying operation after a member leave from $O(N)$ *to* $O(logN)$ [8]. Additionally, new approaches attempt to bound the communication cost and processing overhead, and hence improve scalability and performance, by performing periodic or batched rekeying instead of rekeying on every membership change [12], [13], [14].

In Figure 1(a), we present a key distribution tree for a multicast group of $N = 8$ members plus the GC. Each member is assigned keys that are along the path traced from the leaf node to the root [8]. For example M_1 is assigned keys $\{K_0, K_{1.1}, K_{2.1}, K_{3.1}\}$. If M_1 leaves the multicast group keys $\{K_0, K_{1.1}, K_{2.1}\}$ need to be updated and the following messages need to be sent to appropriate group members.

$$\begin{aligned} GC \rightarrow M_2 : &\quad \{K'_{2.1}\}_{K_{3.2}}, \{K'_{1.1}\}_{K_{3.2}} \\ GC \rightarrow \{M_3, M_4\} : &\quad \{K'_{1.1}\}_{K_{2.2}} \\ GC \rightarrow \{M_2 - M_4\} : &\quad \{K'_0\}_{K'_{1.1}} \\ GC \rightarrow \{M_5 - M_8\} : &\quad \{K'_0\}_{K_{1.2}} \end{aligned}$$

In Figure 1(b) we need to send an individual message to every member of the multicast group since the members do not share any common keys other than K_0.

2.3 Definitions

In wired networks the communication overhead of key management is measured in number of messages. However, the number of messages does not reflect the energy expenditure parameter, which is a vital constraint of wireless ad hoc networks. We first introduce some useful definitions connecting the network energy expenditure with key man-

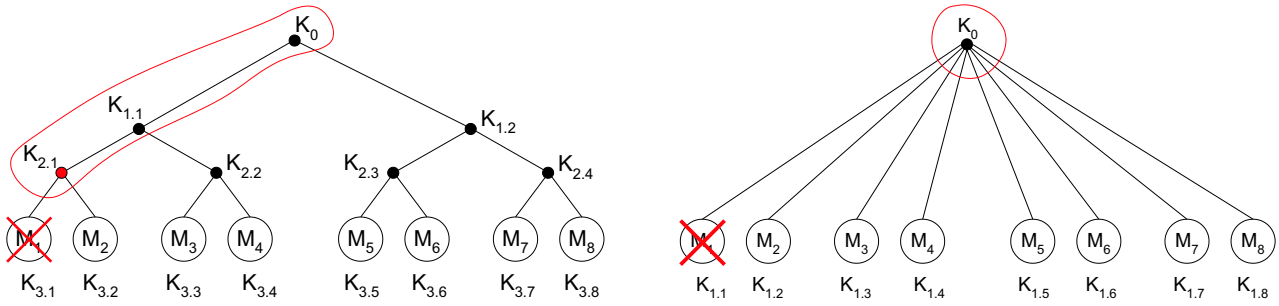


Figure 1: M_1 leaves the multicast group in (a) binary key tree hierarchy, (b) N-degree key tree (key star) hierarchy.

agement operations.

Definition 1 $\mathbf{E_S(R)}$: Key update energy for subset S. The energy required to broadcast a key to members of the subset S according to the routing tree R. The keys are securely distributed through the routing paths that are established from the routing procedure. The key update energy for a subset S is the amount of energy that the network (GC and relay nodes) has to spend, so all members of the subset S successfully receive the key.

Definition 2 $\tilde{\mathbf{E}}_{\mathbf{M_i}}\mathbf{(R,T)}$: Key update energy for member leave. The energy required for rekeying when the i^{th} member leaves the multicast group. After a member leaves the multicast group, several keys need to be updated. The key update energy for a member leave is the amount of energy that the network has to spend, so that all keys are successfully delivered to appropriate members.

Definition 3 $\mathbf{E_{Total}(R,T)}$: Total update energy. The total energy required for rekeying when every user leaves once the multicast group, according to the key distribution tree T and the routing tree R.

$$E_{Total}(R,T) = \sum_{i=1}^{N} \tilde{E}_{M_i}(R,T) \qquad (1)$$

In a balanced tree structure any member leave requires the same number of messages for rekeying due to symmetry. However, in an ad hoc setup, member leaves lead to different energy expenditure for rekeying, although the number of messages remains the same. $E_{Total}(R,T)$ expresses the sum of all rekeying energies if each user leaves the multicast group assuming that all others are present.

Instead of the total update energy we can use the average update energy $E_{Ave}(R,T)$, assuming that each member has equal probability of leaving the multicast group.

$$E_{Ave}(R,T) = \frac{1}{N} \sum_{i=1}^{N} \tilde{E}_{M_i}(R,T) \qquad (2)$$

3 Network model and Problem Statement

3.1 Network model assumptions

We assume that the network consists of N members of a multicast group plus the GC, randomly distributed in a specific area. We consider a single-sender multiple-receiver communication model. All users can act as relay nodes and therefore relay information to any user within their communication range defined by their transmission power. The nodes of the network are assumed to be static (no mobility is incorporated).

The network nodes have limited computational capabilities and constrained energy resources. However, we assume that they are capable of generating and managing cryptographic keys.

We consider a heterogeneous medium, where a model with varying path loss characterizes the signal attenuation in different parts of the network region. We encounter heterogeneous environments in many ad hoc networking situations, especially during mobile operations. Even when node locations are relatively static as assumed in our model, path loss attenuation can vary significantly when the network is deployed in mountains, dense foliage, urban, or inside different floors of a building. Though our algorithms can be combined with any propagation model, we consider two models of varying path loss for calculating the power attenuation at a distance d from the transmitter [15], [16]: (a) Suburban area - A slowly varying environment where the attenuation loss factor changes slowly across space, (b) Office building - A rapidly varying environment where the attenuation loss factor changes fast over space.

We assume that signal transmission is the major component of energy expenditure and therefore ignore any energy cost due to computation and information processing [1], [2]. We further assume that omnidirectional antennas are used for transmission and reception of the signal.

The omnidirectionality of the antennas used for signal transmission and reception results in a property unique in

the wireless environment known as the *broadcast advantage* [17]. When the sender is transmitting a message to a member, all other members that lie within the communication range will receive the broadcasted message for free. Hence, when an identical message needs to be sent to multiple receivers, the sender can significantly reduce the energy expenditure by making one transmission to the farthest member. However, omnidirectional antennas require more power to transmit a signal at distance d than directional ones [1].

We assume that the network has been successfully initialized, and initial cryptographic quantities (pair-wise trust establishment) have been distributed. Several novel approaches that address the critical problem of secure initialization in ad hoc networks with energy limitations, have been recently presented in [2], [18], [19], [20].

In [23], we showed that for a homogeneous medium, the joint consideration of physical and network layers leads to energy-efficient solutions for key distribution. In this paper, we do not attempt a cross-layer design. We assume that the underlying routing is optimized in order to minimize the energy required for broadcast. Although it is known that finding the optimal solution for power-optimal broadcast is NP-complete [24], [25], [26], several heuristics have satisfactory performance in constructing a sub-optimal broadcast routing tree [17], [25], [27].

We also assume that nodes can acquire their location information through the Global Positioning System (GPS) [28], [29]. This assumption has its own security challenges, but this is not our focus. For indoor ad hoc networks where GPS performs fairly poorly, location information may be acquired through a location support system [30]. After a node correctly acquires its location, it can report it to other nodes through a location service algorithm [31], [32]. It is essential to the robustness of our algorithms that node location is reported in a secure and verifiable way [33].

3.2 Problem statement

Our goal is to minimize the energy expenditure due to the communication overhead for performing key management for group communications (multicast or broadcast mode) in a wireless ad hoc network deployed in a heterogeneous medium.

We develop key distribution schemes that satisfy the following requirements:

- Minimize the average update energy $E_{Ave}(R,T)$ when the routing tree R is fixed.
- Adapt to the heterogeneity of the medium where the network is deployed.
- Are scalable with group size in communication overhead and required memory allocation.
- Limit the number of messages required to be exchanged for ensuring protection against unauthorized access when changes in group membership occur.

[1] We have considered directional antennas in the context of routing, but do not include it in this paper.

4 Performance Evaluation Metric

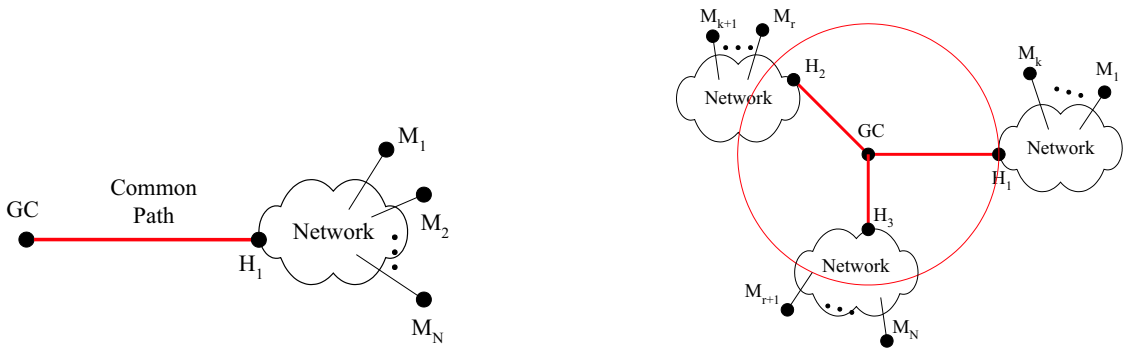
In this section we introduce a new performance evaluation metric that reflects the energy expenditure associated with the key management overhead. We also show that if key graphs are evaluated with the new metric, their performance is dependent not only on key graph structure, but also on node location.

4.1 Energy expenditure of the communication overhead

In wired networks, the communication overhead associated with the key management is measured as the number of exchanged messages between the multicast group members and the GC, when a membership change occurs. The storage cost is measured as the number of keys each member needs to store. Depending on the key distribution scheme employed, there is a tradeoff between the storage cost and the communication cost. In Figure 1(a) each member stores 4 keys, but only 5 messages are required for updating the cryptographic keys when member M_1 leaves the multicast group. In Figure 1(b) each member stores only 2 keys, but 7 messages are required for updating the cryptographic keys when member M_1 leaves the multicast group.

If key trees are employed as a key distribution method, a higher degree tree will result in lower storage, but higher communication overhead. For logical key hierarchies as proposed in [9] the communication cost is equal to $\delta \log_\delta N$ where δ is the degree of the tree. The optimal degree for lowest communication cost is $\delta = 3$ [9], [3]. For key trees that use one way functions, the communication cost is equal to $(\delta - 1)\log_\delta N$ and the optimal tree degree is $\delta = 2$ [10], [3].

In wireless ad hoc networks, every message requires different energy expenditure depending on the location of the receiver and the surrounding environment. Hence, a lower number of transmitted messages does not imply lo- wer energy expenditure. As it will be shown in subsection 4.3, it is possible for a higher degree tree to have a lower energy cost, even if more messages need to be transmitted. Therefore, for energy-constrained ad hoc networks the optimal



(a) (b)

Figure 2: (a) Members $M_1 \ldots M_N$ share a first-hop common path. (b) Members $M_1 \ldots M_N$ exploit broadcast advantage on the first-hop.

tree degree is network topology dependent.

Although we are interested in keeping the number of message exchange low, our major design goal is minimizing the energy expenditure due to communication overhead (we assume no bandwidth constraint). We introduce a new performance evaluation metric called *average update energy* defined in (2), to measure the energy consumption due to communication.

$$E_{Ave}(R,T) = \frac{1}{N}\sum_{i=1}^{N}\tilde{E}_{M_i}(R,T)$$

As mentioned earlier, we consider the case of a member leave since significantly higher communication cost occurs during a member leave than a member join [8].

4.2 Comparison of multi-hop broadcast and unicast

In this section we provide a useful Lemma that will allow us to compare the energy expenditure of different key distribution schemes.

Lemma 1 *If network nodes transmit with omnidirectional antennas and the same routing tree is used for both unicast and broadcast, then wirelessly broadcasting a message to a group of nodes always requires strictly less energy than unicasting the message to each node of the group.*

$$E_{\{M_1..M_N\}}(R) < \sum_{i=1}^{N} E_{\{M_i\}}(R),\ N \geq 2 \qquad (3)$$

Proof: Assume that the GC sends an identical message to N members of the multicast group where $N \geq 2$. The routing is considered to be fixed and will be omitted from the equations. When broadcasting a message, all receivers that lie within the communication range can receive the message with only one transmission. Hence, broadcasting a message to those receivers can never require more energy than unicasting the message multiple times to each one of them.

In order to show that a strict inequality holds in a wireless environment, it is enough to show that energy savings occur in the first hop from the sender. There are only two cases concerning the first hop from the source of the broadcast (GC) to the N members, $N \geq 2$.

Case 1: The first hop from the GC towards the N members is a common path.

When broadcasting a message to all N members, the message will be sent only once through the common path and then be relayed to the N members through possibly different paths. In unicast mode the message has to be sent N times through the common path. Hence, in broadcast mode we save at least $(N-1)$ times the energy expenditure for the common path. This case is demonstrated in Figure 2(a).

Case 2: The first hop from the GC towards the N is not a common path.

If the first hop is not a common path, there will be at least two different nodes used to relay the broadcast message to the N members. In such a case these two different nodes will exploit the broadcast advantage [17]. In unicast mode the message has to be sent N times through the different paths leading to the N members. Hence, in broadcast mode we save at least the energy needed to relay the message to the nearest first-hop node. This case is demonstrated in Figure 2(b).

■

4.3 Comparison of the energy expenditure due to key updates in a binary tree with a tree of degree N

In this section we will show that, depending upon node location, a key tree of degree N (key star) that requires $O(N)$ rekey messages after a member leave, can have less

energy expenditure due to communication overhead than a binary key tree that requires $O(\log N)$ rekey messages after a member leave[2]. Hence, fewer number of rekey messages do not imply lower energy consumption.

Revisiting Figure 1, we calculate the energy expenditure occurring when member M_1 leaves the multicast group and all the cryptographic keys known to M_1 need to be updated. Recall that in wired networks where the communication overhead was measured in number of messages, the binary key tree always had a lower overhead than the key star [8]. For wireless ad hoc networks where the communication overhead is measured in energy units, the performance is dependent on the location of the nodes. For clarity of the equations we will use the metric in definition 2 for the comparison of the two key tree structures.

For the scheme in Figure 1(a), we need to send 5 update messages. The energy required for updating the key tree after M_1 leaves the multicast group is:

$$\begin{aligned}\tilde{E}_{M_1}(Tree) &= 2E_{\{M_2\}} + E_{\{M_3,M_4\}} \\ &\quad + E_{\{M_2..M_4\}} + E_{\{M_5..M_8\}} \end{aligned} \quad (4)$$

For the scheme in Figure 1(b), we need to send an individual message to every member of the multicast group, since the members do not share any common keys other than K_0. The energy required for updating the key tree after M_1 leaves the multicast group is:

$$\tilde{E}_{M_1}(Star) = \sum_{i=2}^{8} E_{\{M_i\}} \quad (5)$$

For comparing the two energies we can write:

$$\begin{aligned}\tilde{E}_{M_1}(Tree) &= 2E_{\{M_2\}} + E_{\{M_3,M_4\}} \\ &\quad + E_{\{M_2..M_4\}} + E_{\{M_5..M_8\}} \\ &= \sum_{i=2}^{8} E_{\{M_i\}} + E_{\{M_2\}} + E_{\{M_3,M_4\}} \quad (6) \\ &\quad + E_{\{M_2..M_4\}} + E_{\{M_5..M_8\}} - \sum_{i=3}^{8} E_{\{M_i\}} \\ &= \tilde{E}_{M_1}(Star) + E_{\{M_2\}} + E_{\{M_3,M_4\}} \\ &\quad + E_{\{M_2..M_4\}} + E_{\{M_5..M_8\}} - \sum_{i=3}^{8} E_{\{M_i\}} \end{aligned}$$

The structure that is more energy efficient can be determined by the quantity in equation (7).

$$\begin{aligned} &E_{\{M_2\}} + E_{\{M_3,M_4\}} + E_{\{M_2..M_4\}} \\ &+ E_{\{M_5..M_8\}} - \sum_{i=3}^{8} E_{\{M_i\}} \underset{Tree}{\overset{Star}{\gtrless}} 0 \end{aligned} \quad (7)$$

[2]Note that an N-degree tree corresponds to $(N-1)$ unicasts and worst case in wired networks. Also note that the use of binary tree can be generalized to arbitrary d-ary tree.

$$\begin{aligned} &E_{\{M_2\}} + E_{\{M_2..M_4\}} \\ &+ (E_{\{M_3,M_4\}} - E_{\{M_3\}} - E_{\{M_4\}}) \\ &\underset{Tree}{\overset{Star}{\gtrless}} \left(\sum_{i=5}^{i=8} E_{\{M_i\}} - E_{\{M_5..M_8\}} \right) \end{aligned} \quad (8)$$

From Lemma 1, the right side of (8) is always positive and the expression inside the brackets on the left side of (8) is always negative. Hence, the left side of (8) can be either negative or positive, while the right side of (8) is always non negative. Also, the left side of (8) is independent of the right side of (8). Depending upon the node location, the N-ary key tree can be more energy efficient than the binary key tree and vice versa. Similar results can be obtained for other key tree structures of arbitrary degree δ.

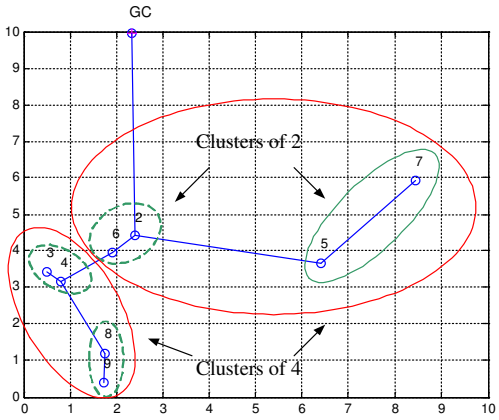
Hence, a solution based on a binary tree may be energy inefficient compared to an N-ary tree that requires $(N-1)$ unicast messages for update. Therefore, the number of messages does not adequately capture the energy impact of the communication overhead. Also key graphs that performed optimally in wired networks do not hold their optimality under the new criteria of energy efficiency.

5 Prior Work for the Homogeneous Medium

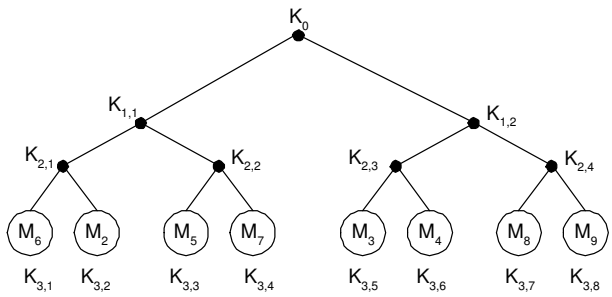
In [4], we developed a location-aware key distribution scheme for the homogeneous medium. We demonstrated the need for consideration of the node location information in the construction of the key distribution tree. We observed that members located close to each other can potentially be reached with broadcast, or use the same routing paths to receive data. Then, we showed that placing near-by members adjacently in the leaves of the key distribution tree leads to great energy savings.

We represented members as points in the 2-dimensional plane, and developed a variant of *K-means* algorithm [34] to cluster them into groups and construct a hierarchical key tree structure. In the homogeneous case, the path loss factor is constant across the network deployment region. The transmit power is proportional to some power of the Euclidean distance and hence, K-means is suitable for the clustering due to monotonicity.

An example for the homogeneous medium: In Figure 3(b) we show the results of the application of the location-aware algorithm as described in [4] for the nine-node network in Figure 3(a). The network is deployed in a homogeneous medium and the path loss factor is assumed to be $\gamma = 2$. We observe that members that are clustered together, share multi-hop common paths. Therefore, when



(a)



(b)

Figure 3: (a) Application of the k-means clustering algorithm in an ad hoc network. (b) Resulting location-aware key distribution tree.

sending an updated key to members of a cluster we save a great amount of energy by transmitting only once through the common path.

6 Location-Aware Key Distribution Scheme for the Heterogeneous Medium

When the wireless medium is heterogeneous, models with variable path loss factor have to be employed to describe the signal attenuation in different parts of the network deployment region. A typical environment where path loss varies is an urban area, or an office building. The signal attenuation for nodes located in different floors is significantly higher than for nodes located in the same floor [16].

If the medium is not homogeneous, high spatial correlation of the nodes does not necessarily translate to low required transmission power for communication. A receiver physically located closer to a transmitter, but being in a location hindered by an obstacle may require more power for communication than a receiver located further away from the transmitter but in line of sight. K-means algorithm that was used in homogeneous path loss model, cannot be combined with an arbitrary dissimilarity measure, since it uses the notion of mean vectors and hence, the same dissimilarity measure has to hold for all points.

To overcome these limitations, we need a grouping method that can incorporate the following:

- *Allow the use of an arbitrary dissimilarity measure (Not just proportional to some power of the Euclidean distance).*
- *Allow the simultaneous use of different dissimilarity measures for different network regions.*
- *Minimize the dissimilarity inside the clusters.*
- *Scale well with group size.*

6.1 Suitable clustering techniques for the heterogeneous medium

In this section we will describe clustering techniques that satisfy the design criteria we set for the heterogeneous medium. By modifying and applying these clustering techniques to the nodes of the network we build a hierarchy that will be mapped to a virtual hierarchy of the key distribution scheme. We identify the clustering methods PAM [35], CLARA [35] and DIANA [35] for these purposes.

PAM: Partitioning Around Medoids [35]. PAM partitions a global cluster into k subclusters, so the total cluster dissimilarity is minimal. It can use an arbitrary dissimilarity measure and can handle the use of various dissimilarity measures simultaneously. In PAM, the creation of k distinct clusters is achieved by determining a representative object (node) for each cluster. The representative object, called k-medoid, is the most centrally located object within the cluster, according to the dissimilarity measure used. The remaining $(N - k)$ objects are assigned to a cluster if they are most similar to its k-medoid. Through a search method, the medoids that result in the smallest total cluster dissimilarity are located. We describe PAM algorithm in Appendix A.1.

PAM is very robust against outliers that significantly degrade the clustering quality. Also it does not depend upon the order by which the nodes are examined. However, it does not scale well with group size. It has $O((1 + \beta)k^2(N - k)^2)$ complexity where k is the number of clusters created and β is the number of successful swaps for finding the optimal k-medoids [35], [36]. For small k as in

our case ($k = 2$), the complexity becomes $O((1+\beta)N^2)$.

CLARA: Clustering LArge Applications [35]. For larger data sets, Kaufman and Rousseeauw developed a scalable method named CLARA (Clustering LARge Applications). CLARA samples the data set and finds the representative objects (k-medoids) by applying PAM on the samples instead of the entire data set. However, the clustering quality is measured by computing the dissimilarity in the whole data set and not just the sample. To improve performance, the above described procedure is repeated for multiple sample sets. The complexity of CLARA is $O(\alpha(k^2s^2+k(N-k))+\beta k^2s^2)$ where α is the number of samples, s is the size of the sample set. It was also shown in [35] CLARA significantly reduces the complexity of PAM. We describe CLARA in Appendix A.1.

DIANA: DIvisible ANAlysis [35]. DIANA is a divisible hierarchial clustering technique that is an intuitive choice for creating a hierarchical tree structure. It also uses arbitrary dissimilarity measures and can handle multiple measures at different network regions. We modified DIANA algorithm to produce a balanced hierarchical tree structure. Initially all objects are contained in a global cluster. At each step, the cluster C with the biggest diameter, where $diam(C) := \max_{i,j \in C} diss(i,j)$ is split in two clusters A, B. Cluster A contains all the objects, while B is empty. The object with the biggest average dissimilarity is moved from A to B. In subsequent steps, objects of A more similar to B are moved to cluster B. The complexity of DIANA is $O(N^3)$.The algorithm terminates after $(N-1)$ splits, when N clusters containing a single object are obtained. We describe the modified DIANA algorithm in Appendix A.2.

6.2 LocKeD - A solution for the heterogeneous medium

In this section we present our location-aware key distribution algorithm for the heterogeneous medium. Initially all members (represented by points on a 3-dimen- sional plane) of the multicast group belong to a global cluster. Depending on the network size, we can employ either PAM or CLARA or DIANA as a method for splitting the global cluster into subclusters that will be mapped to the tree hierarchy.

The dissimilarity measure $diss(i,j)$ used as a clustering criterion in all methods is the required power $P(d_{i,j})$ for communication between nodes i and j computed according to the path loss model that is assumed.

LOCation-aware KEy Distribution scheme (LocKeD) for Heterogeneous Medium

Step 1: *Assign all points to an initial global cluster.*

Step 2: *Divide each cluster into two clusters with PAM, or CLARA or modified DIANA algorithm.*

Step 3: *If PAM or CLARA is used, apply the Refinement Algorithm (RA).*

Step 4: *Iterate step 2 and 3 until clusters of two points have been created.*

Step 5: *Map the cluster hierarchy into tree hierarchy.*

Our refinement algorithm ensures that the created clusters have been assigned equal number of points so that the final key tree is balanced. Assume that a cluster C is split into two clusters C_1 and C_2. If cluster C_1 has $|C_1|$ points and medoid m_{C_1}, and cluster C_2 has $|C_2|$ points and medoid m_{C_2}, with $|C_1| > |C_2|$, the refinement procedure moves to C_2, $\lfloor (|C_1| - |C_2|)/2 \rfloor$ points with the highest dissimilarity with medoid m_{C_1}.

Refinement Algorithm - RA

for $(k := 1; k \leq \lfloor (|C_1| - |C_2|)/2 \rfloor ; k++)$
  find i* such that

$$diss(x_{i^*}, m_{C_2}) = \min_{i \in C_1} diss(x_i, m_{C_2}) \qquad (9)$$

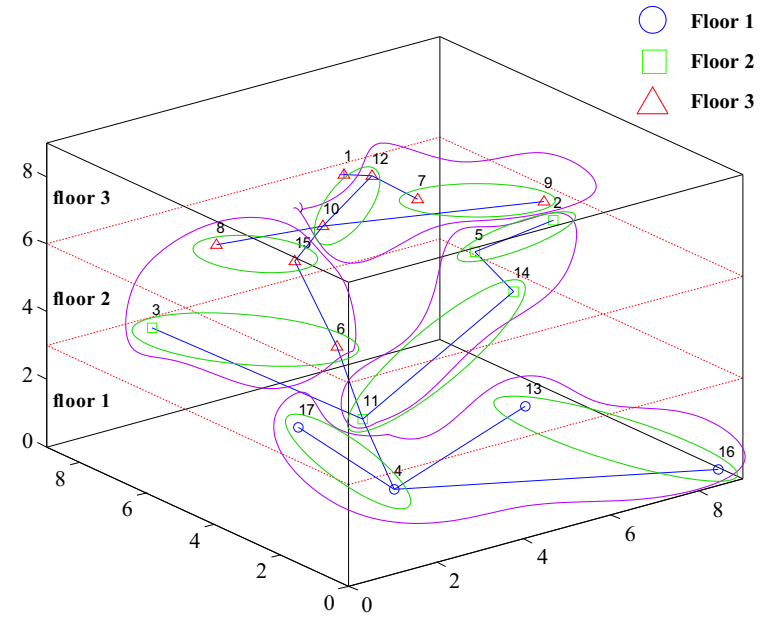
  and move it to cluster C_2.
endfor

When LocKeD uses PAM we will call it LocKeD I, when it uses CLARA we will call it LocKeD II and when it uses DIANA we call it LocKeD III.

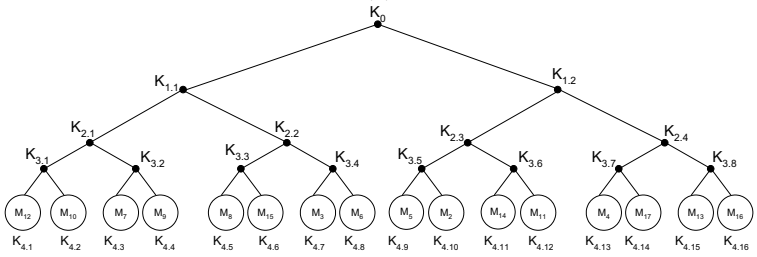
An example for the heterogeneous medium: We applied LocKeD I in the sixteen-node plus the GC network in the 3-story building of Figure 4(a). In Figure 4(b), we show the key distribution tree constructed by the application of LocKeD I. The path loss model for the required power for communication between nodes i, j is described in equation (10) [15], [16].

$$PL(d_{i,j}) = PL(d_0) + 10\gamma \log\left(\frac{d_{i,j}}{d_0}\right) \qquad (10)$$

where PL is the signal attenuation in dB and d_0 is the close-in reference distance set to 1m. Table 1 shows typical values of the attenuation factor and its standard deviation σ measured in different locations inside the building [15].



(a)



(b)

Figure 4: (a) Application of the location-aware key distribution scheme using LocKeD I in a 17-node network deployed in a 3-story building. (b) Resulting location-aware key distribution tree.

Location	γ	σ **(dB)**
Same floor	2.76	12.9
Through one floor	4.19	5.1
Through two floor	5.04	6.5
Through three floor	5.22	6.7

Table 1: The path loss factor and its standard deviation for an office building.

From Figure 4 we can observe the preference of the clustering to group together nodes that are located in the same floor, as the required power for communication is significantly smaller. Even if nodes are closely located, but belong to different floors (nodes 2, 9 for example) they are not grouped together since the have high dissimilarity.

7 Performance Evaluation

7.1 Description of the simulation setup

Simulation was performed in randomly generated network topologies confined in a specific region. After the network generation, the routing paths were established according to the routing algorithm. Though any suitable algorithm can be applied to provide the routing paths, we used BIP [17]. The resulting routing tree was used to calculate the consumed energy for transmitting an updated key to each of the group members. Our network is assumed to be static.

Since there is no algorithm to provide the optimal solution for the key distribution tree construction, we performed exhaustive search for small group sizes $N = 8, N = 16$. For larger group sizes, $N = 32, 64, 128, 256$, we generated for each network instance, 10,000 different key tree structures and compared the performance of our algorithms with the key tree that requires the minimum, maximum and median energy for key update, out of the 10,000 tree structures. Further, we repeated the same comparison for 100 different network topologies and averaged the result. Although the best solution might not be included in the 10,000 generated trees, this still remains a valid comparison since it emulates the logical assignment in the wired networks.

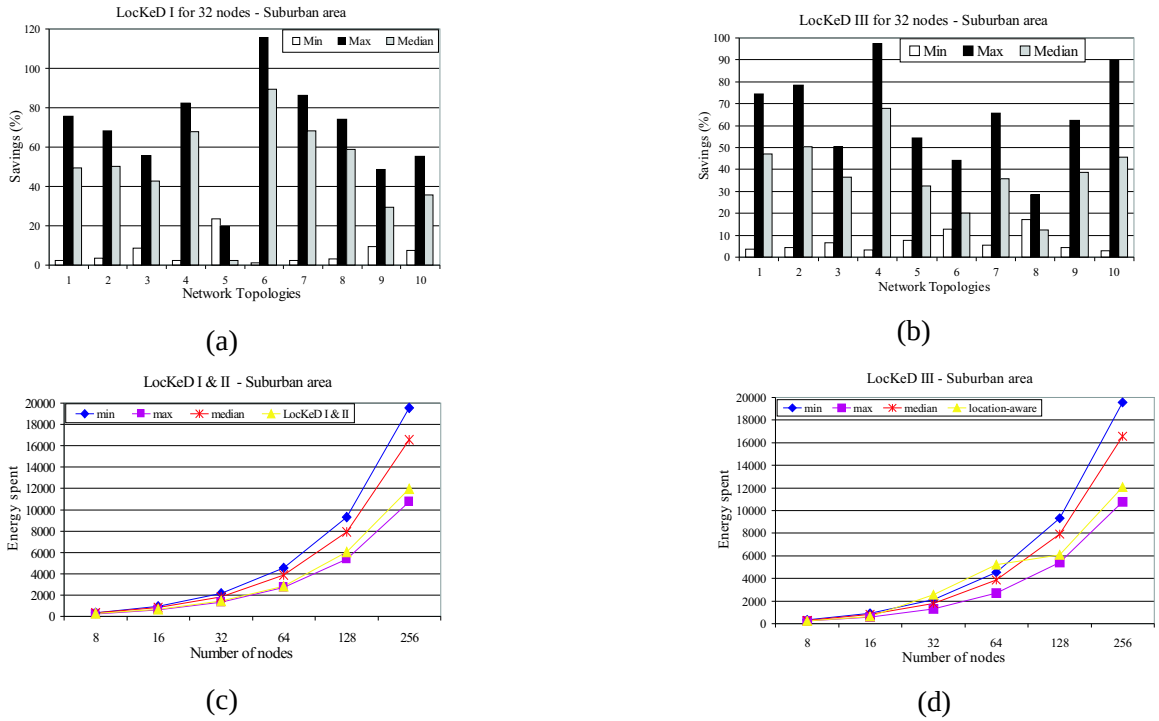


Figure 5: Savings for a network of 32 nodes plus the GC for 10 different network topologies deployed in a suburban area: (a) LocKeD I. (b) LocKeD III.
Savings for different number of nodes deployed in a suburban area, averaged over 100 different network topologies: (c) LocKeD I and II (d) LocKeD III.

7.2 Experiment 1: Suburban area

In our first experiment we evaluated the performance of the location-aware key distribution scheme for a slowly variating heterogeneous medium. We considered a suburban area where the attenuation factor γ is not constant throughout the network deployment region. However, we assumed that it changes slowly across space. We confined our network in a 12x12 square region and assumed the path loss model in equation (11) for computing the required power for communication between nodes i, j at distance $d_{i,j}$.

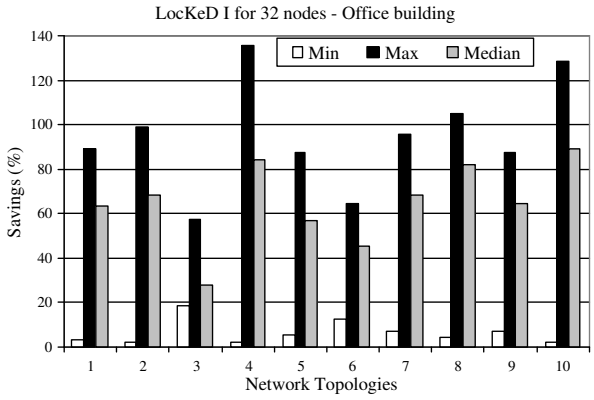
$$P(d_{i,j}) = \begin{cases} d_{i,j}^2, & d_{i,j} \leq 4 \\ d_{i,j}^3, & 4 \leq d_{i,j} \leq 8 \\ d_{i,j}^4, & 8 \leq d_{i,j} \leq 12 \end{cases} \quad (11)$$

Nodes that are located near-by are assumed to be in line-of-sight (LOS, $\gamma = 2$), while nodes located farther away experience higher attenuation ($\gamma = 3$ or $\gamma = 4$).

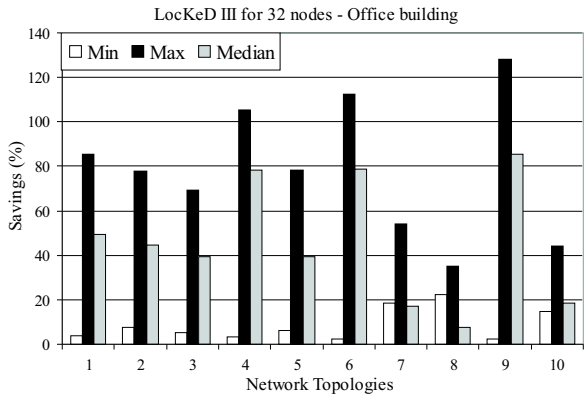
We applied LocKeD I, II and III to construct the key distribution tree and compared the performance with the best, worst and median solution out of 10,000 randomly generated trees. The dissimilarity measure used in all clustering algorithms was the required communication power as described in equation (11). In Figure 5(a) the energy savings of the LocKeD I, for a network of 32 nodes plus the GC in a suburban area for 10 different network topologies are shown. In Figure 5(b) the average savings over 100 different network instances are shown.

We can observe that LocKeD I and II, provides 20% - 118% savings from the worst possible assignment, 5% - 85% savings from the median case and does 3% - 22% worse than the best possible case out of 10,000 trees. The gains in a heterogeneous environment are significantly higher than a homogeneous one, due to the different path loss model assumed. If inefficient node grouping is chosen (two far away nodes are grouped together) the energy penalty will be far higher as the attenuation factor will be equal to 4. Hence, a bad key distribution tree results in a very poor performance.

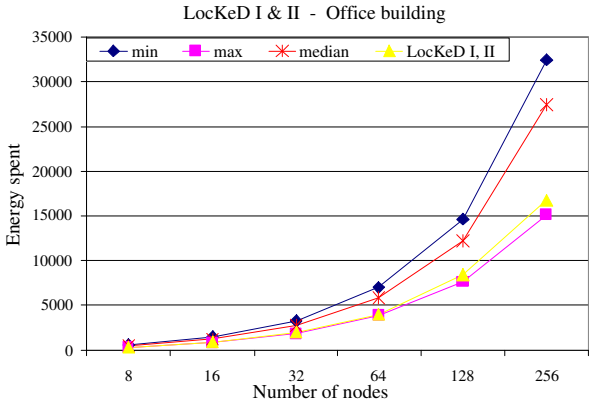
In some cases our location-aware key distribution scheme performs average rather than close to the optimal tree structure. The performance of the clustering technique for constructing the key distribution tree is dependent upon the correct prediction of the routing paths. Nodes that are located near-by, will receive information through similar paths. Hence, their grouping will result in energy savings in the key update. However, clustering fails to capture the circularity of the broadcast advantage. Nodes might be located in opposite directions, but receive broadcast (one



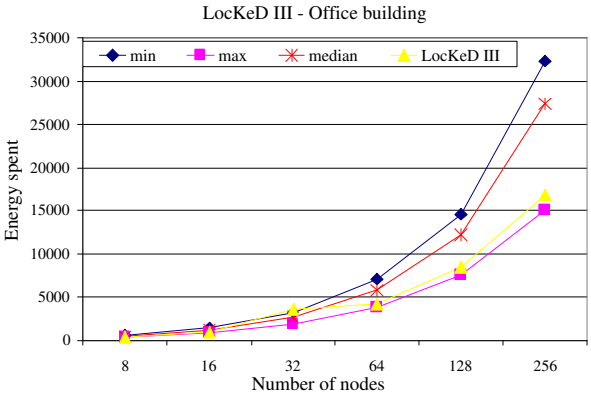
(a)



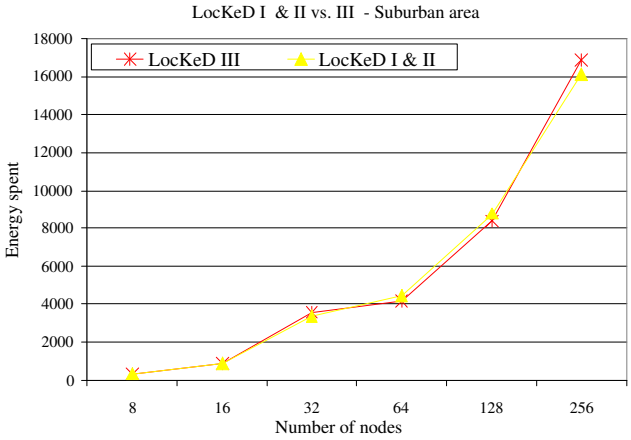
(b)



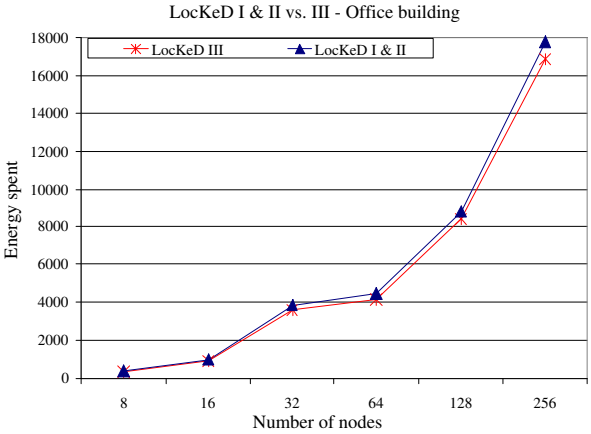
(c)



(d)



(e)



(f)

Figure 6: Savings for a network of 32 nodes plus the *GC* for 10 different network topologies deployed in a 4-story office building: (a) LocKeD I. (b) LocKeD III.
Savings for different number of nodes deployed in a 4-story building averaged over 100 different network topologies: (c) LocKeD I and II. (d) LocKeD III.
Comparison between LocKeD I, II with LocKeD III averaged over 100 different network topologies, for different number of nodes deployed in: (e) suburban area (f) 4-story office building.

receives the information for free). Although, grouping of such nodes is beneficial for reducing the energy expenditure, clustering will not group them together.

In Figure 5(c) we present the energy savings of LocKeD III, for a network of 32 nodes plus the GC in a suburban area for 10 different network topologies. In Figure 5(d) we show the average savings over 100 different network instances. Again, we can observe a case where clustering performs average due to the circularity of the broadcast advantage.

We can observe that LocKeD III, provides 25% - 97% savings from the worst possible assignment, 12% - 68% savings from the median case and does 4% -16% worse than best possible case out of 10,000 trees. LocKeD I performs slightly better than LocKeD III as it is an exhaustive search algorithm, and LocKeD III has been modified to meet the balanced structure requirement. However, as the number of nodes increases and LocKeD II is used instead of LocKeD I the performance becomes identical to LocKeD III.

7.3 Experiment 2: 4-story office building

In our second simulation experiment we evaluated the performance of the location-aware key distribution algorithm for a heterogeneous medium with a rapid changing environment. We assumed that the network is deployed in 4-story office building emulated by a 12x12x12 cubic space and the path loss is described by (10).

We used LocKeD I, II and III to construct the key distribution tree and compared the performance with the best, worst and median solution out of 10,000 generated key trees. The dissimilarity measure used for clustering was the required communication power as described in equation (10) and Table 1. In Figure 6(a) we present the energy savings of LocKeD I, for a network of 32 nodes plus the GC in a heterogeneous medium (suburban area) for 10 different network topologies. In Figure 6(b) we present the average savings over 100 different network instances.

We can observe that LocKeD I and II for the office building environment, provides even higher savings up to 136% from the worst possible assignment, 90% savings from the median case and does 2%-20% worse than the best possible case out of 10,000 trees. The severe attenuation environment increases the performance gap between efficient and inefficient key distribution trees.

In Figure 6(c) we present the energy savings of LocKeD III, for a network of 32 nodes plus the GC in an office building for 10 different network topologies. In Figure 6(d) we present the average savings over 100 different network instances.

We can observe that LocKeD III, provides 37% - 128% savings from the worst possible assignment, 10% - 82% savings from the median case and does 3%-21% worse from the best possible case out of 10,000 trees. Again, LocKeD I performs better than LocKeD III.

In Figure 6(e) we compare LocKeD I and II with LocKeD III for the suburban heterogeneous environment. The two methods have almost identical performance as the attenuation factor does not change rapidly over space. In Figure 6(f) we compare LocKeD I and II with LocKeD III for the office building environment. As we explained earlier, LocKeD I and II perform better than LocKeD III.

7.4 Stopping criteria and algorithm complexity

The complexity of LocKeD I is $O((1+\beta)N^3)$ since PAM has to be applied multiple times to produce the final hierarchy. LocKeD II reduces the complexity of LocKeD I to $O(N^2)$. The complexity of LocKeD III is $O(N^3)$. An intuitive question is whether we can have satisfactory performance by only doing partial calculation. We could reduce the complexity of our location-aware key distribution algorithm by terminating the clustering at a specific level of the tree construction. Instead of splitting up from the initial global cluster down to clusters of 2 members, we can stop the clustering procedure at a stage where more than 2 members belong to one cluster and randomly place these members randmoly at the leaves of the subtree representing that cluster.

We performed the following experiment to evaluate the effect of terminating the algorithms earlier than the last splitting step that refines into clusters of 2 members. We created 100 random network topologies of 32 nodes plus the GC and we applied LocKeD I to generate the appropriate clusters. After every split (construction of each level of the tree) we generated key trees corresponding to the current clusters. For example when clusters of 8 members were formed we restricted members to belong to a specific subtree of 8 members in the key distribution tree, but placed them randomly in the leaves of the subtree. We then observed the gains achieved by further applying clustering and splitting the tree to acquire more detailed clustering. We averaged the result over the 100 network topologies and obtained Figure 7.

From the graph we can make the following important observation. The first split is very beneficial since on average it gives gains about 30% compared to the random key tree. The first step clustering can roughly separate the deployment region into two sectors and reduce the chances of bad clustering of nodes. Subsequent splits do not give similar gains to justify the additional complexity of the al-

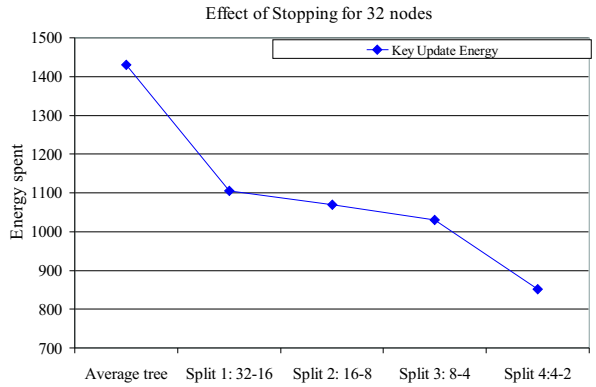


Figure 7: The effect of terminating the algorithm before clusters of 2 pairs have been created for reduced complexity.

gorithms. However, the last split that results in clusters of 2 members gives an additional 20% gain in energy savings.

The explanation for the leap in energy savings observed in the last split, lies in the numbers of messages that the pairs of members receive during a key update due to a member leave. Consider Figure 4(b) and member M_{12} leaving the multicast group. Members M_7 and M_9 (cluster of 2) have to receive 3 different keys, while members M_5, M_2, M_{14}, M_{11}, M_4, M_{17}, M_{13}, M_{16} (cluster of 8) have to receive only 1 key. Hence, a good clustering in members of 2 will lead to great energy savings (more keys need to be updated for those clusters).

Hence, both the first and the last split are critical in the energy reduction of the energy required for updating the key tree after a member leave.

8 Related Work

Providing secure communication in an ad hoc network environment is a fairly unexplored research area. Very few key management schemes have recently been proposed for such networks.

In [19], a key management scheme for Distributed Sensor Networks (DNS) of ad hoc nature is proposed. The scheme is based on key pre-distribution and probabilistic key sharing among the nodes of the network and provides services of rekeying, key revocation, as well as node addition and deletion. By using arguments from random graph theory, the authors propose a key pre-distribution of a small subset of keys, called key ring, randomly selected from a large pool of keys, to each node of the network. The key rings are chosen in such a way that two nodes share a key with probability p. A link exists between two nodes only if they share a key. Additionally, two nodes that are not within each other's communication range can establish a pair-wise key if a path key exists between them. However, all communication between the two nodes is exposed to the intermediate nodes that determine the path key.

In [20], the authors strengthen the resilience against node capture by requiring q common keys to exist between two nodes for secure link establishment, at the expense of increased vulnerability against large scale attacks. Additionally, they propose the key update through multiple independent paths so that a malicious adversary will not be able to acquire the updated keys in case of a node capture. The also employ random pair-wise keys for supporting security functions as node identity authentication and node revocation.

In [37], [38], [39], public key cryptography is proposed as a key management scheme for wireless ad hoc networks. In [37] threshold cryptography is proposed for the distribution of trust in the key management. N servers are able to sign certificates and $t + 1$ of them are required to perform a cryptographic operation. In [38] the servers that realize the threshold scheme are called Mobile Certificate Authorities (MOCA). In [40] public key cryptography is proposed for multicast application in ad hoc networks. However, the proposed scheme does not attempt to minimize the energy consumption due to the communication overhead. In [2] identity-based public-key cryptography is proposed for energy-efficient group keying is sensor networks. The proposed scheme focuses on securing link-layer broadcasts of local groups and does not address large multicast groups.

The use of digital signatures require increased computational power and communication resources. In an ad hoc environment mobile hosts often have low computational power and operate in a limited battery. We do not focus on authntication which is also a very critical parameter studied in [18], [20], [21], [22]. Hence, our work is based on symmetric, rather than asymmetric cryptography. We are motivated by the observation that the incorporation of physical layer parameters in the key distribution leads to energy efficiency. In our prior work for homogeneous medium, we also presented cross-layer design schemes [23].

Our work is based upon performing key management by using key graphs structures and especially trees, although it can be directly extendable to any form of graph. Key trees were initially proposed for group communication for wired networks in [8], [9], mainly addressing the scalability issues associated with multicast. In [12], [13], [14] periodic or batched rekeying instead of rekeying on every membership change is proposed in an attempt to bound the communication cost. All proposed key management schemes are logical while our schemes consider the un-

derlying network structure in the design.

9 Future Work and Open Problems

Optimization over the graph structure: In our present work we showed that node location has to be taken into account when designing a key management scheme for group communications in a wireless ad hoc environment. We adopted the tree structure as a scalable solution, and developed algorithms to perform key management in an energy efficient way. We presented the application of our algorithms to binary trees, although they are directly applicable to a tree of any degree. What structure is most energy efficient on the average sense, remains an open problem. For example, if ternary trees are used instead of binary trees, fewer keys have to be stored, but more messages have to be transmitted for a key update. Since each message has different energy cost, a ternary tree might prove on average more energy efficient than a binary one, depending on the network topology.

Incorporation of mobility for the nodes of the network: Mobility can be seen as a series of successive snapshots of the network. If nodes are mobile, the network topology is dynamic and the underlying routing tree is adapting to it in a dynamic way. Since our algorithms are tightly dependent upon the location of the nodes, if that location changes, the key tree structure will be outdated and will no longer reflect the network topology. Hence, we have to develop a method to adjust the key tree structure to the dynamically changing network structure. However, we note that the specifics of the type of mobility model used is critical in this context

The issues involved with the algorithm to be developed concern how frequently the adjustment has to take place in order to be energy efficient, how sensitive is our key structure to routing changes, how can we battle effects of oscillation and deal with problems of stabilization.

Impact of directional antennas: In our analysis we assumed that omnidirectional antennas were used by all the nodes of the network. The use of omnidirectional antennas allowed energy savings due to the broadcast advantage [17]. Certain nodes were receiving information for free when they were within the range of communication of a specific link. However, the circularity of the communication is degrading the performance of our location-aware key distribution scheme in certain network topologies where clustering cannot group together nodes located in different directions.

If directional antennas are used for transmission, greater savings can occur by using our location-aware key distribution scheme. The sectoring provided by the directional antennas can limit the angular range of the broadcast advantage and hence make clustering more effective. Further investigation is required to analyse the impact of directional antennas in the behavior of our location-aware algorithms.

10 Conclusions

We studied the problem of multicast key management in energy-constrained wireless ad hoc networks, under heterogeneous path loss. We observed that in energy-constrained wireless ad hoc networks, the communication overhead of key management has to be measured in energy units and introduced a new performance metric called average update energy. We developed 3 key distribution algorithms that make use of the node location and of the variable path loss parameter. We showed that unlike wired case, the energy-constrained ad hoc multicast key trees do not have a fixed optimal degree tree. We illustrated the application of our algorithms for suburban and indoor environment and presented the gains achieved. We also listed open problems.

References

[1] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-Aware Wireless Microsensor Networks," IEEE Signal Processing Magazine, Vol. 19, Issue 2, pp:40–50, March 2002.

[2] D.W. Carman G.H. Cirincione and B.J. Matt, "Energy-Efficient and Low-Latency Key Management for Sensor Networks," in Proc. of the 23^{rd} Army Science Conference, Orlando FL, December 2002.

[3] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor and B. Pinkas, "Multicast Security: A Taxonomy and Some Efficient Constructions," In Proc. of IEEE INFOCOM 99.

[4] L. Lazos and R. Poovendran, "Energy-Aware Secure Multicast Communication in Ad-hoc Networks Using Geographic Location Information," in Proc. of IEEE ICASSP 2003, Hong Kong, China, 2003.

[5] W. Diffie, and M Hellman, "New Directions in Cryptography," IEEE Transactions in Information Theory, IT-22, pp. 644-654, November 1976.

[6] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, 21(2):120 pp. 120-126, 1978.

[7] D.W. Carman, P. Kruus and B.J. Matt, "Constraints and Approaches for Distributed Sensor Network Security," NAI Labs Technical Report #00-010 Sep. 2000.

[8] D.M. Wallner, E.C. Harder and R.C. Agee, "Key Management for Multicast: Issues and Architectures," INTERNET DRAFT, Sep. 1998.

[9] C.K. Wong, M. Gouda and S. Lam, "Secure Group Communications Using Key Graphs," IEEE/ACM Trans. On Networking Vol.8, No.1, pp. 16-31, Feb. 2000.

[10] D. Balenson, D. McGrew and A. Sherman, "Key management for large dynamic groups: One-way Function trees and amortized Initialization," INTERNET DRAFT, Feb. 1999.

[11] A. Perrig, D. Song and D. Tygar, "ELK, a new protocol for efficient large-group key distribution," In Proc. of the IEEE Security and Privacy Symposium 2001, May 2001.

[12] Y. Yang, X. Li and S. Lam, "Reliable Group Rekeying: Design and Performance Analysis," in Proc. of IEEE ACM SIGCOMM 2001, San Diego, CA, USA, Aug. 2001.

[13] X. Li, Y. Yang, M. Gouda and S. Lam, "Batch Re-keying for Secure Group Communications," in Proc. of World Wide Web 10 (WWW10), Hong Kong, China, May 2001.

[14] S. Setia, S. Koussih and S. Jahodia, "Kronos: A Scalable Group Re-Keying Approach for Secure Multicast," In Proc. of the IEEE Security and Privacy Symposium 2000, Oakland, CA, USA, May 2000.

[15] S. Seidel and S. Rappaport, "914 MHz Path Loss Prediction Models for Indoor Wireless Communications in Multifloored Buildings," in IEEE Transactions on Antennas and Propagation, Vol. 40, No.2, pp. 207-217, February 1992.

[16] T. Rappaport, *Wireless Communications: Principles & Practice*, Prentice Hall, New Jersey 1996.

[17] J.E. Wieselthier, G.D. Nguyen and A. Ephremides, "On the Construction of Energy Efficient Broadcast and Multicast Trees in Wireless Networks," in Proc. of IEEE INFOCOM 2000, Tel-Aviv, Israel, pp. 586-594.

[18] F. Stajano and R. Anderson, "The resurrecting duckling: Security issues for ad-hoc wireless networks," In Security Protocols, 7^{th} International Workshop, 1999.

[19] L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks," In Proc. of the 9^{th} ACM Conference on Computer and Communications Security Washington D.C., USA, November 2002.

[20] H. Chan, A. Perrig and D. Song, "Random Key Predistribution Schemes for Sensor Networks," in IEEE Symposium on Security and Privacy, California, USA, May 2003.

[21] R. Canetti, A. Perrig D. Song and D. Tygar, "The TESLA Broadcast Authentication Protocol," in RSA Cryptobytes, Summer 2002.

[22] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. Tygar. "SPINS: Security Protocols for Sensor Networks," In Seventh Annual ACM International Conference on Mobile Computing and Networks(Mobicom 2001), Rome, Italy, July 2001.

[23] L. Lazos and R. Poovendran, "Secure Broadcast in Energy-Aware Wireless Sensor Networks," IEEE International Symposium on Advances in Wireless Communications (ISWC'02), Victoria BC, Canada, September 2002.

[24] F. Li and I. Nikolaidis, "On Minimum-Energy Broadcasting in All-Wireless Networks," in Proc. of the 26^{th} Annual IEEE Conference On Local Computer Networks (LCN 2001), Tampa, Florida, November 2001.

[25] M. Cagalj, J.P. Hubaux and C. Enz, "Minimum-Energy Broadcast In All Wireless Networks: NP-Completeness and Distribution Issues," in Proc. of the 8^{th} ACM Annual International Conference on Mobile Computing and Networking, (MobiCom 2002), Atlanta, Georgia, September 2002.

[26] W. Liang, "Constructing Minimum-Energy Broadcast Trees in Wireless Ad Hoc Networks," in Proc. of the 3^{th} ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2002), Lausanne, Switzerland, June 2002.

[27] S. Lee, W. Su, J. Hsu, M. Gerla and R. Bagrodia, "A Performance Comparison Study of Ad hoc Wireless Multicast Protocols," in Proc. of the IEEE Conference on Computer Communications (INFOCOM 2000), pp. 565-574, Tel Aviv, Israel, March 2000.

[28] Educational Observatory Institute GPS page, available via WWW at URL: http://www.edu-observatory.org/gps/gps.html.

[29] G. Dommety and R. Jain, "Potential networking applications of global positioning systems (GPS)," Technical Report TR-24, CS Dept., The Ohio State University, April 1996.

[30] N.B. Priyantha, A. Chakraborty and H. Balakrishnan, "The Cricket Location-Support System," in Proc. of the Sixth Annual ACM International Conference on Mobile Computing and Networking (MOBICOM), Boston MA, August 2000.

[31] J. Li, J. Jannotti, D. De Couto, D. Karger and R. Morris, "A Scalable Location Service for Geographic Ad-Hoc Routing," in Proc. of the Sixth Annual ACM International Conference on Mobile Computing and Networking (MOBICOM), Boston MA, August 2000.

[32] Y. Xue, B. Li, and K. Nahrstedt, "A Scalable Location Management Scheme in Mobile Ad-Hoc Networks," In Proc. of the IEEE Conference on Local Computer Networks (LCN 2001), Tampa FL, November 2001.

[33] N. Sastry, U. Shankar and D. Wagner, " Secure Location Verification," in Proc. of the ACM workshop on Wireless security (Wise 2003), San Diego CA, September 2003.

[34] T. Hastie, R. Tisbshirani and J. Friedman, *The Elements of Statistical Learning, Data Mining, Inference and Prediction*, Springer Series in Statistics, NY, 2001.

[35] L. Kaufman and P.J. Rousseeauw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, 1990.

[36] S. Chu, J. Roddick, T. Chen and J. Pan, "Efficient Search Approaches for K-medoids-based Algorithms," In Proc. of the International Conference on Computers, Communications, Control and Power Engineering (IEEE TENCON'02), Beijing, China, 2002.

[37] Z. Haas and L. Zhou, "Securing ad hoc networks," in IEEE Network Magazine, vol. 13, no. 6, pp. 24-30, Nov./Dec. 1999.

[38] S. Yi and R. Kravets, "Key management for heterogeneous ad hoc wireless networks," University of Illinois at Urbana-Champaign, Department of Computer Science Technical Report #UIUCDCS-R-2001-2241, UILU-ENG-2001-1748, July 2002.

[39] S. Capkun, L. Buttyan, and J. Hubaux "Self-Organized Public-Key Management for Mobile Ad-Hoc Networks," in IEEE Transactions on Mobile Computing (TMC) 2002.

[40] S. Mki, T. Aura, and M. Hietalahti, "Robust membership management for ad-hoc groups," In Proc. 5^{th} Nordic Workshop on Secure IT Systems (NORDSEC 2000), Reykjavik, Iceland, October 2000.

A Appendix

A.1 K-medoids clustering algorithms

The idea of k-medoids is to select k representative objects, called medoids $m_1, m_2, \ldots m_k$, one for each cluster. The choice of the representative object should minimize the sum of the dissimilarities of all the objects to their nearest medoid.

$$Total\_diss = \sum_{i=1}^{N} \min_{j=1,..,k} diss\,(i,j) \qquad (12)$$

After the total dissimilarity in (12) has been minimized each object is assigned to the cluster C_i that corresponds to the nearest medoid.

$$diss(i, m_{C_i}) \leq diss(i, m_{C_r}), \forall r = 1, \ldots, k \qquad (13)$$

If the representative objects belong to set $O_{rep} = \{m_1, m_2, \ldots m_k\}$ and all the rest of the objects belong to set $O_{rem} = O - O_{rep}$ where O is the set of all objects, the PAM (Partitioning Around Medoids) algorithm can be stated as follows:

Partitioning Around Medoids (PAM) [35]

Step 1: *Select k representative objects arbitrarily and compute the total dissimilarity as in equation 12.*

Step 2: *Compute the total dissimilarity for all pairs of objects* (i, j) *with* $i \in O_{rep}$ *and* $j \in O_{rem}$.

Step 3: *If a smaller dissimilarity is found, swap* $i \leftrightarrow j$ *which decreases the total dissimilarity the most and go to step 2.*

Step 4: *Otherwise, assign each object* $j \in O_{rem}$ *to the nearest medoid* $i \in O_{rep}$ *and halt.*

PAM does not scale for large data sets. Instead Kaufman and Rousseeuw have proposed CLARA (CLustering LARge Applications) that relies on sampling [35]. The representative objects are drawn from a sample set rather than the whole data set. Then PAM is applied on the sample and the best k-medoids are found from the sample. For better performance multiple random samples are drawn and the medoids that give the best clustering are kept. Simulations in [35] have shown that 5 different samples of $40 + 2k$ objects are sufficient to give satisfactory results.

Clustering Large Applications (CLARA) [35]

Step 1: *Repeat for 5 different samples.*

Step 2: *Draw a sample of 40+2k objects randomly form O.*

Step 3: *Apply PAM to the sample and compute the total dissimilarity as in equation 12.*

Step 4: *Retain the medoids the yield to the smaller total dissimilarity.*

Step 5: *Assign all objects in O to the nearest medoid computed in step 4.*

A heuristic that improves the performance of CLARA involves the inclusion of the best set of medoids O_{rep} found so far to the next sample. Except from the first sample, all subsequent ones draw $40 + k$ samples and add the best medoids so far to the sample set.

A.2 Divisible Analysis (DIANA) [35]

Diana is a divisible hierarchical method. Initially, all objects belong to a global cluster. At each step, the largest available cluster is split in two subclusters, until all clusters contain 1 single objects. Hence, $(N - 1)$ successive splits have to occur in order for the algorithm to terminate. DIANA uses the average dissimilarity measure defined as in (14).

Definition 4 $\mathbf{a(i)}$ **Average Dissimilarity:** For each object i of a cluster A, the average dissimilarity is the sum of the dissimilarities between i and the rest of the objects of A, divided by the number of objects of A minus one.

$$a\left(i\right) = \frac{1}{|A| - 1} \sum_{j \in A, j \neq i} diss\left(i, j\right) \tag{14}$$

Definition 5 $\mathbf{diam(J)}$ **Diameter of a cluster** J**:** The diameter of a cluster J is the largest dissimilarity between two of its objects.

$$diam(C) := \max_{i,j \in C} diss(i, j) \tag{15}$$

Definition 6 $\mathbf{w(i, B)}$ **Average dissimilarity with objects of another cluster** B**:** The average dissimilarity of object $i, i \in A$ with the objects of cluster B.

$$w\left(i, B\right) = \frac{1}{|B|} \sum_{j \in A, j \neq i} diss\left(i, j\right) \tag{16}$$

Assume that C is the initial global cluster and A, B are the clusters after the split. Assume also that we want to assign equal number of points to each cluster.

Modified DIANA

Step 1: *Repeat until all objects belong to a single cluster.*

Step 2:

$$A := \max_{J \in clusters} diam(J) \ \ and \ \ B = \varnothing \tag{17}$$

Step 3: *Calculate the average dissimilarity* $a(i), \forall i \in A$. *Move the object with the highest* $a(m)$ *to cluster* B. $A := A - \{m\}, B = \{m\}$.

Step 4: *If* $|A| = 1$ *stop.*

Step 5: *Otherwise,* $\forall i \in A$ *compute the average dissimilarity to all objects of* B $w(i, B)$.

Step 6: *Repeat until* $|A| = \frac{N}{2}$*:*
Select the object h for which

$$a(h) - w(h, B) = max_{i \in A}(a(i) - w(i, B)) \tag{18}$$

and move it to cluster B.
end repeat
go to Step 1.