Supporting Secure Communication and Data Collection in Mobile Sensor Networks

Li Zhou and Jinfeng Ni and Chinya V. Ravishankar
Department of Computer Science & Engineering
University of California, Riverside
Riverside, CA 92521, USA
{lzhou,jni,ravi}@cs.ucr.edu

***Abstract*— Sensor deployments may be static, but researchers have recently been making a case for mobile collector nodes to enhance data acquisition. Since mobile nodes are often more privileged, their compromise can give the adversary a significant advantage. Hence, security mechanisms for such networks must tolerate mobile node compromises. Unlike static sensors, which communicate mostly with their neighbors, mobile nodes may communicate with nodes all over the network. Hence, key establishment is a much harder challenge with mobile nodes.**

We first analyze the impact of mobile collector compromises on the reliability of data received by the base station, and the circumstances under which reliability can be guaranteed. Second, we present mGKE, a key predistribution scheme for very general group-based sensor deployments. mGKE allows any pair of neighboring sensors to establish a unique pairwise key, regardless of sensor density or distribution. It is also usable by mobile collectors. Our analysis and evaluation show the superiority of mGKE over current methods in terms of resilience, connectivity, communication overhead, and memory requirements.

I. Introduction

Sensor networks are already used in applications like monitoring of traffic, the environment or wildlife, and in battlefields. Sensors are now cheap enough to be deployed in numbers, and on-demand [25]. To enhance effectiveness and resilience, they are commonly deployed in groups [18], [6], [12].

Data within a group is aggregated by sensors called *cluster heads*, and sent to one or more *base stations* for analysis. Recently, researchers [29], [31], [14], [27], [32] have suggested using mobile collectors within a static sensor network to facilitate data collection. Sensing regions may be large, or far from base stations in applications such as battlefields or hazard monitoring, so sending data directly to base stations will waste energy at intermediate sensors, increase delay, and render transmitted data liable to manipulation en-route.

A. Issues and Challenges

We will use the term *node* to refer to a sensor or a mobile collector. We will apply the term *mobile sensor network* whenever some nodes are mobile. Security is an important concern in such networks. Sensors are resource-constrained devices with little physical protection, making them prone to compromise or capture. Attackers could mount false report attacks to waste resources, or even trick the base station into making wrong decisions with serious high-level consequences [30], [37], [36], [28]. Mobile collectors could also be compromised, causing even more serious damage [32].

A single static base station may be adequately secured. However, when there are many mobile collectors, one or more of them may become compromised. An important issue is to analyze the impact of such compromises on security. We must also secure communications between neighboring static sensors, between static sensors and mobile collectors, and between mobile collectors. This task is challenging. First, sensor nodes are resource-limited, ruling out expensive public key cryptosystems such as RSA [23]. Second, the ad-hoc, on-demand nature of sensor deployments, as well as mobility cause a senor’s neighbors to be unknown before deployment, so shared keys can not be preloaded in any simple way.

Figure 1 shows a battlefield with static sensors and a set of mobile Robomotes [24]. When an enemy is sensed, the sensors collaborate to aggregate data, which can later be collected by a Robomote, and sent to the base station. Soldiers may also carry mobile collectors in backpacks. In such cases, mobile collectors may have more memory, computing, battery power, and transmission range than static sensors. Now consider an ocean or river water monitoring scenario, where some sensors are anchored, while others are floating, and are carried around by water movements. It is likely that static and mobile sensors in this scenario are resource-constrained in similar ways.

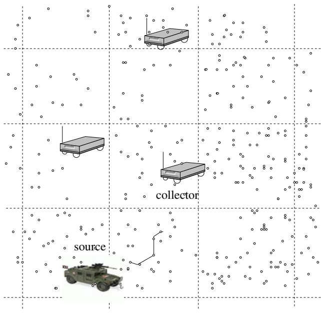


Fig. 1. Robomotes collecting data.

B. Our Work: Mobility and Security

We address two issues in our work. First, we analyze the impact of compromised mobile collectors on security, and the circumstances under which the system can remain secure. Second, we present a key predistribution method that

1-4244-0222-0/06/$20.00 (c)2006 IEEE

allows pairwise key establishment for very general sensor deployments, and in the presence of mobility.

Mobility makes it harder to secure communication in sensor networks. Surprisingly, our analysis shows that mobility can in fact *improve* data consistency when mobile collectors may be compromised. Our analysis can serve as a foundation for introducing mobile collectors into static sensor networks.

We also present mGKE, a **G**roup-based **K**ey **E**stablishment scheme for mobile sensor networks, an extension of the GKE scheme [35], [34] to support mobility. The mGKE scheme is efficient and effective for a very general group-based sensor deployment model, in the presence of mobility. First, unlike previous random key predistribution schemes [9], [5], [6], [7], [17], [18] which require high sensor density with uniform distribution, mGKE can establish unique pairwise key for any pair of neighboring nodes regardless of sensor density or distribution, as long as the network is physically connected. Second, mGKE security degrades gracefully with the number of compromised sensors, significantly improving the resilience against node compromise. Finally, mGKE has far lower communication overhead than schemes like PIKE [4], which require network-wide communication for key establishment. The communication required in mGKE for pairwise key establishment is localized to two adjacent groups.

The rest of this paper is organized as follows. We describe related work in Section II, and list our assumptions in Section III. We analyze the impact of compromised mobile collectors and the circumstances under which reliability can be guaranteed in Section IV. In Section V, we present the mGKE scheme. In Section VI, we describe the metrics that we use to evaluate the security and performance of mGKE. We analyze the security of mGKE scheme in Section VII, and evaluate its performance in Section VIII. Section IX concludes the paper.

II. Related Work

A. Resilience Against Node Compromise

Compromised nodes can generate false reports or drop valid reports. We focus on detecting false reports. This is an area of significant current interest, and a large body of work including [30], [37], [36], [28] are designed to filter the false reports as early as possible.

Mobile collectors introduce new security challenges. As [32] argues, mobile collectors are typically privileged, and are hence attractive targets. Compromised mobile collectors may abuse their privileges, mounting various attacks that might compromise the entire network. Security mechanisms must hence be resilient to mobile collector compromises. Zhang et al. [32] proposed several schemes to limit mobile collector privileges, based on establishing privilege-dependent pairwise keys between mobile collectors and sensors. Successful key establishment serves as proof of privilege. However, the total number of node compromises these schemes can tolerate is limited by a sensor's memory, and can be as low as 200. This threshold is too low, since sensor deployments can consist of thousands of nodes.

B. Mobility and Security

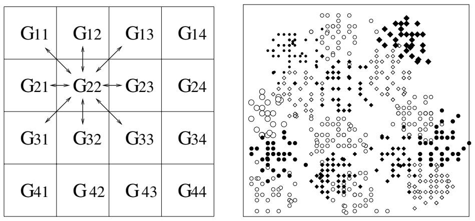
Several researchers have argued that mobility can facilitate secure communication and authentication in mobile ad hoc networks [16], [1], [3]. However, their work targets the domain of ad-hoc wireless networks, while we address security in sensor networks, which are far more resource constrained. The techniques in [16], [3] all use public-key cryptosystems, making them unsuitable for resource-constrained sensor networks. Also, the work in [16], [1], [3] is concerned with establishing secure associations or providing certificate service among mobile nodes. In contrast, our work deals with the impact of compromised mobile collectors on the data collection in sensor networks.

C. Key Predistribution

Recently, *random key predistribution (RKP)* schemes [9], [5], [6] have been proposed for large scale sensor networks. The basic idea is to randomly preload each sensor with a subset of keys from a global key pool before deployment. Since these subsets are chosen randomly, any pair of sensors will share a key with a certain probability. Two neighboring sensors can choose any element in the intersection of these subsets to be their pairwise key. When these subsets are disjoint, two neighboring sensors may establish a *path key* using intermediary sensors. These schemes are based on results from random graph theory [8], which guarantee that a random graph is connected with high probability if the number of edges in it exceeds a threshold. To improve the resilience of RKP against node capture, [7], [17], [18] proposed *structured random key predistribution (SRKP)* schemes, which have a nice threshold property: When the number of compromised sensors is less than a threshold, other keys shared between non-compromised sensors are affected with probability close to zero.

However, these random key predistribution schemes suffer from two major problems, which make them inappropriate for many applications. First, these schemes require that the deployment density be high enough to ensure connectivity. This requirement seriously hinders the use of RKP and SRKP when sensor networks are sparse, as is likely when sensors fail over time and new sensors are added, or when the deployments are themselves sparse. Second, the approach to key (or key space) sharing in RKP and SRKP also degrades resilience against node capture. Compromising a sensor also compromises the set of keys (or key spaces, respectively) in it, so that the security of all other sensors using keys from this set (or space) will be weakened.

The PIKE proposal [4] addresses the high density requirements in RKP and SRKP. In PIKE, each sensor is assigned an ID of the form (i,j), which corresponds to a location on a $\sqrt{n} \times \sqrt{n}$ grid, where n is the network size. Each sensor is also preloaded with pairwise keys, each of which is shared with a sensor that corresponds to a location on the same row or the same column of the grid. Now, any pair of sensors that do not share a preloaded pairwise key can use one or more peer sensors as trusted intermediaries to establish a path key.



(a) Groups in [12] (b) Groups in mGKE

Fig. 2. Group-based deployment methods

PIKE shows significantly improved security over SRKP since pairwise keys are unique. However, PIKE requires network-wide communication to establish path keys, each of which requires $O(\sqrt{n})$ communication overhead [4]. In addition (see Section VIII-B.2), a large fraction ($> 99\%$) of neighboring sensor pairs in PIKE do not share preloaded keys, and must therefore establish path keys. Consequently, the PIKE scheme can involve significant communication overhead, making it unsuitable for large sensor networks.

III. Issues and Assumptions

We assume that sensors have resource limitations typical of the current generation of sensors, such as MICA2 motes [25], and that they are deployed in a group-based fashion as in [18], [6], [12]. If the deployment region is viewed as a collection of subregions , previous group-based schemes [6], [12], have assumed that the assignment of groups to subregions is fixed, so that group adjacencies are known before sensor deployment. Figure 2(a) illustrates the group-based deployment approach in [12]. In contrast (Figure 2(b)), we assume that any randomly chosen group of sensors can be deployed into a subregion, making our sensor deployment more flexible.

We assume that mobile collectors may either be resource-rich class devices, or be resource-limited sensors as in [24], [21]. We present separate schemes for each case. We also assume that sensors and mobile collectors are prone to capture. Once a node is compromised, all keys stored at the node are known to attackers. We assume that attackers may eavesdrop, intercept or manipulate transmitted packets. We design efficient key generation schemes to secure the communication in such networks.

For purposes of analysis, we assume that mobile collectors follow the Random Waypoint model [2]. This mobility model is very common in wireless mobile networks, and our analysis may be extended to other mobility models. As shown in Section IV, this mobility model may actually improve data consistency when false report attacks [30] are mounted by compromised mobile collectors.

IV. Dealing with Mobile Collector Compromises

Consider n_m mobile collectors moving in a detecting region. We assume the Random Waypoint mobility model, so waypoints are distributed uniformly in the region. At step i, a mobile node moves at constant velocity v from its current waypoint P_{i-1} to a new random waypoint P_i, where it pauses for a constant time w to communicate with neighboring sensors. It does not communicate with sensors while in transit.

This region is divided into g subregions of equal area. A collector collects data from the sensors in each subregion it visits, and relays this data to the base station. During an interval T of interest, several mobile collectors may visit a subregion r_i, so the base station has several reports of data for r_i. The base station can compare these reports to filter out wrong reports.

If there are x compromised mobile collectors, an attacker can send at most x false reports for subregion r_i to the base station. Let $Y(T)$ be the number of uncompromised mobile collectors visiting a given subregion. The base station uses majority voting when reports are in conflict.

Definition 1 (Data Consistency): The data for subregion r_i is said to be *consistent* if $Y(T) > x$ for r_i.

Here we present an analysis of the data consistency . Intuitively, since any mobile collector visits any subregion at each step with the same probability, the expected number of uncompromised mobile collectors visiting a subregion during interval T increases with number of steps taken, which increases with T. We will show that with high probability, more uncompromised mobile collectors will visit a subregion than compromised ones, for reasonable configurations. The base station is hence likely to receive consistent data. To the best of our knowledge, this is the first work which gives a theoretical analysis on the relationship between data consistency under compromised mobile collectors and the mobility model.

A. Data Consistency under Random Waypoint Mobility Model

Let r_i be ith subregion in the region. Let $\mathbf{C_m(x)}$ be the event that x mobile collectors have been compromised. Let $\mathbf{K(i)}$ be the event that the data received for r_i are consistent. Let $Y(T)$ be the number of uncompromised mobile collectors which appear in r_i during the time interval T. Now, the probability of *consistency* for r_i with x compromises is

$$\Pr[\mathbf{K(i)}|\,\mathbf{C_m(x)}] = \Pr[Y(T) > x].$$

Let the collector m_j take τ_j steps during the interval T. Let its ith step be of length l_i. The time taken for step i is $t_i = (l_i/v + w)$. Using linearity of expectation, we get

$$E[t_i] = E[(l_i/v + w)] = E[l_i]/v + w,$$

and

$$E[t_i^2] = E[(l_i/v + w)^2] = E[l_i^2]/v^2 + 2(w/v)E[l_i] + w^2.$$

since the variance $\sigma^2(X) = E[X^2] - (E[X])^2$,

$$\sigma^2(t_i) = \frac{E[l_i^2]}{v^2} + 2\frac{w}{v}E[l_i] + w^2 - (\frac{E[l_i]}{v} + w)^2.$$

When the detecting region is an $s \times s$ rectangle, we know from [2] that

$$E[l_i] = 0.5214s, \text{ and } E[l_i^2] = \frac{s^2}{3}.$$

The number of steps m_j takes in time T is

$$\tau_j(T) = \max\{k : S_k \leq T\}, \text{ where } S_k = \sum_{i=1}^{k} t_i.$$

Technically, $\tau_j(T)$ is a *renewal process*, since t_i are independent identically distributed non-negative random variables [11]. If F_k is the distribution function of $S_k = \sum_{i=1}^{k} t_i$, we know from the theory of renewal process [11], that

$$\Pr[\tau_j(T) = k] = F_k(T) - F_{k+1}(T). \qquad (1)$$

Since l_i is bounded by the length $\sqrt{2}s$ of the square region's diagonal, we get $\tau_j \geq \tau_{\min} = T/(\sqrt{2}s/v + w)$ as a lower bound. Using a step size of zero gives us the upper bound $\tau_j \leq \tau_{\max} = T/w$.

At each step, m_j visits any subregion with probability $p = \frac{1}{g}$. Let $\mathbf{V_j(i)}$ be the event that m_j visits subregion r_i at least once, and let $\boldsymbol{\tau}_j$ be the event that m_j takes τ_j steps during interval T. Now,

$$\Pr[\mathbf{V_j(i)} | \, \boldsymbol{\tau}_j] = 1 - q^{\tau_j}, \text{ where } q = 1 - p.$$

Now,

$$\begin{aligned}
\Pr[\mathbf{V_j(i)}] &= \sum_{\tau_j=\tau_{\min}}^{\tau_{\max}} \Pr[\mathbf{V_j(i)} | \boldsymbol{\tau}_j] \, \times \Pr[\boldsymbol{\tau}_j] \\
&= \sum_{\tau_j=\tau_{\min}}^{\tau_{\max}} (1 - q^{\tau_j}) \Pr[\boldsymbol{\tau}_j] \\
&= 1 - E[q^{\tau_j}].
\end{aligned}$$

Since $t_1, t_2, \cdots, t_k$ are all independent and identically distributed, $\tau_1, \tau_2, \cdots, \tau_{n_m}$ will also be i.i.d. Hence, $\Pr[\mathbf{V_j(i)}]$ will be the same for all mobile collectors, regardless of j. To estimate this probability, we apply the Central Limit Theorem and approximate $S_k = \sum_{i=1}^{k} t_i$ with a Gaussian distribution. Hence,

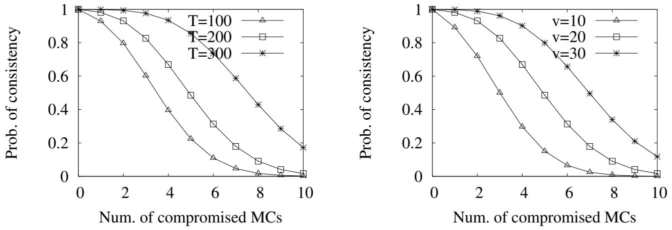
$$\mu(S_k) = kE[t_i], \text{ and } \sigma^2(S_k) = k\sigma^2(t_i).$$

From Equation 1, $\Pr[\boldsymbol{\tau}_j] = (F_{\tau_j}(T) - F_{\tau_j+1}(T))$, where $F_{\tau_j}(T)$ is the Gaussian distribution function for S_{τ_j}. Hence,

$$\begin{aligned}
\Pr[\mathbf{V_j(i)}] = 1 - E[q^{\tau_j}] &= 1 - \sum_{\tau_j=\tau_{\min}}^{\tau_{\max}} q^{\tau_j} \times \Pr[\boldsymbol{\tau}_j] \\
&= 1 - \sum_{\tau_j=\tau_{\min}}^{\tau_{\max}} q^{\tau_j} \times (F_{\tau_j}(T) - F_{\tau_j+1}(T)) \\
&= 1 - \sum_{\tau_j=\tau_{\min}}^{\tau_{\max}} q^{\tau_j} \times (\Pr[S_{\tau_j} \leq T] - \Pr[S_{\tau_j+1} \leq T])
\end{aligned}$$

We can compute $\Pr[S_{\tau_j} \leq T]$ and $\Pr[S_{\tau_j+1} \leq T]$ from Gaussian approximations for S_{τ_j} and S_{τ_j+1}.

Now, there are n_m collectors in all, of which $n_m - x$ are uncompromised. Each mobile collector will visit subregion r_i at least once with probability $\beta = \Pr[\mathbf{V_j(i)}]$, so that $Y(T)$ is Binomially distributed with success probability β. That is,



(a) Consistency vs. T (v=20). (b) Consistency vs. speed (T=200).

Fig. 3. The security under compromised mobile collectors (MCs).

$$\Pr[Y(T) = y] = \binom{n_m - x}{y} \beta^y (1 - \beta)^{n_m - x - y}.$$

Therefore, we have

$$\begin{aligned}
\Pr[Y(T) > x] &= 1 - \sum_{y=0}^{x} \Pr[Y(T) = y] \\
&= 1 - \sum_{y=0}^{x} \binom{n_m - x}{y} \beta^y (1 - \beta)^{n_m - x - y}
\end{aligned}$$

B. An Example

As an example, suppose we have a detecting area $1,000m \times 1,000m$ divided into 100 subregions. Suppose the mobile collectors move at speeds of $v = 10m/s$, $20m/s$ or $30m/s$ between consecutive waypoints, and pause for $w = 5s$ to collect data. Let there be 100 mobile collectors. Suppose base stations collect data every $T = 100s$, $200s$ and $300s$. Using our analysis, Figure 3(a) plots the probability of data consistency for any subregion with respect to T, when $v = 20m/s$, while Figure 3(b) plots the consistency with respect to speed v when T is fixed to be 200, in the case that x mobile collectors are compromised.

Clearly, as T or v increases, the probability of consistency increases. This is expected, since higher T and v will allow each uncompromised mobile collector to visit more subregions. That is, each subregion will be visited by more uncompromised mobile collectors. Based our analysis, we conclude that when the application does not requires time-sensitive data, we can improve the data consistency by increasing T. For real-time applications, we can trade the consistency with the power consumed for mobility by increasing the speed of the mobile collectors.

V. The mGKE Scheme

We now present mGKE, a pairwise key establishment scheme for securing communications between neighboring static sensors, between static sensors and mobile collectors, and between mobile collectors. We use the notation in Table I.

Let there be n_s sensors and n_m mobile collectors. We will denote the ith static sensor by s_i and the jth mobile collector by m_j. We arrange the static sensors into g groups $G_i, 1 \leq i \leq g$, each of which has $\gamma = n_s/g$ sensors. Group G_u will comprise sensors s_i such that $(u-1)\gamma < i \leq u\gamma$. Let $\langle G_u, s_i \rangle$

Notation	Description
s_i	the i-th static sensor
m_j	the j-th mobile collector
G_u	the u-th static sensor group
n_s	the number of static sensors
n_m	the number of mobile collectors
γ	the group size
g	the number of groups
δ	the average number of sensors in a sensor's transmission range
μ	the number of preloaded keys that a sensor shared with sensors that are in the different groups
t	the number of agents (see Definition 1 in Section V-B) for a group or a mobile collector in every other group

TABLE I
OUR NOTATION

denote sensor s_i from group G_u. We will replace $\langle G_u, s_i\rangle$ by s_i, when no confusion can arise.

In the following, we refer to the pairwise key between a pair of sensors as an S-S key, and the pairwise key between a mobile collector and a sensor as an M-S key.

A. Outline of mGKE

mGKE preloads each sensor or mobile collector with a carefully chosen set of keys, each shared pairwise with one other node. We say that two nodes are *associated* if they share a *preloaded* pairwise key.

To establish pairwise key between any neighboring sensor pair, we preconfigure each sensor so that it is associated with every other sensor in its own group. We also ensure that each sensor is associated with sensors from one or more other groups, in a pattern designed to ensure several sensor associations across each pair of groups. A sensor s_i can now establish a unique pairwise key with any of its neighbors s_j. If s_i and s_j are from the same group, they start off associated. If they are from different groups, there will exist *multiple* associations between their groups, so they can establish a pairwise key using any pair of these associated sensors as intermediaries. This process involves only localized communication, which differentiates our scheme from PIKE [4].

To establish pairwise keys between a mobile collector and a nearby sensor, we present two different approaches. The first method is usable only when the mobile collector has $O(n_s)$ memory, but the second method is usable when mobile nodes have resources as limited as regular sensors. In our second approach, the base station selects a subset of groups for each mobile collector m_i, and preloads m_i with keys ensuring several associations with each of the selected groups. m_i can now establish a unique pairwise key with any of its neighboring sensors $\langle G_u, s_j\rangle$ using its associations in G_u (or in any nearby group, since all groups are associated). A mobile collector pair can use this method to establish a path key.

We describe S-S key establishment in Section V-B, and M-S key establishment in Section V-C.

B. Key Establishment for Neighboring Sensor Pairs

1) S-S Key Predistribution: We preload each pair of sensors from the same group with a unique pairwise key. This strategy is feasible even with limited memory if we choose the group size γ appropriately. For example, let the group size γ be 100 as in [6], [12], so that each sensor must store 99 keys. If the key size is 64 bits, each sensor requires 792 bytes. This is doable for a Mica2 Mote sensor that has $4KB$ SRAM [25]. We can further halve this memory requirement by using the techniques adopted in [4], so that allocating 396 bytes for keys suffices to ensure that any pair of neighboring sensors from the same group share a unique preloaded key.

We now address key establishment between neighboring sensors from different groups. We begin with a few definitions.

Definition 2 (Agent): $\langle G_u, s_i\rangle$ is called an agent for G_v in G_u, if $\langle G_u, s_i\rangle$ is associated with some $\langle G_v, s_j\rangle$ in G_v.

Definition 3 (t-Associated Group): Groups G_u and G_v are said to be t-associated if they have t agents for each other.

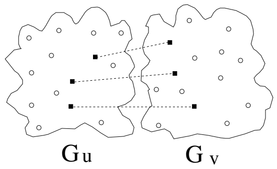


Fig. 4. G_u and G_v are 3-associated.

Since each pair of agents share a pairwise key, neighboring sensors from groups G_u and G_v can establish path keys using any pair of agents as intermediaries. As group adjacencies are unknown prior to sensor deployment under our deployment model, the problem of key establishment between sensors in different groups reduces to that of creating group associations.

We will require each group to be associated with every other group. If there are g groups, and each sensor has enough memory to hold μ inter-group pairwise keys, each group can have up to $t = \lceil \frac{\mu\gamma}{g-1} \rceil$ agents in each of the other groups. Algorithm 1 shows how to define group associations. We use functions $\mathcal{F}_i$ $(1 \le i \le t)$ which uniformly map group pairs from $[1, g] \times [1, g]$ to $[1, n_s]$. $\mathcal{F}_i(G_u, G_v)$ selects the ith agent for G_v in G_u, and is defined as follows

$$\mathcal{F}_i(G_u, G_v) = \big(t(v-1) + i\big) \pmod{\gamma} + (u-1)\gamma.$$

G_u comprises the sensors s_i with $(u-1)\gamma < i \le u\gamma$. Hence $\mathcal{F}_1(G_u, G_v), \cdots, \mathcal{F}_t(G_u, G_v)$ select t sensors, with indices between $\big(t(v-1)+1\big) \mod \gamma + (u-1)\gamma$ and $tv \mod \gamma + (u-1)\gamma$ as agents for G_v.

Algorithm 1 Inter-group S-S key predistribution

```
$t = \lceil \frac{\mu\gamma}{g-1} \rceil$
for each pair of groups $G_u$, $G_v$ do
  for $i = 1$ to $t$ do
    $s_x = \mathcal{F}_i(G_u, G_v)$
    $s_y = \mathcal{F}_i(G_v, G_u)$
    assign a unique pairwise key to $s_x$ and $s_y$
  end for
end for
```

Algorithm 1 has the following attractive features:

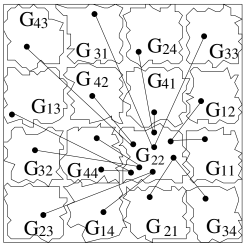


Fig. 5. Inter-Group Key Predistribution for G_{22} ($t = 1$)

- *Uniformity*: Each sensor is agent for the same number of groups. This balances loads and creates no high-value targets, since no sensor holds more keys than any other.
- *Resilience*: Multiple agents improve resilience for establishing path keys.
- *Easy agent discovery*: Agents can be discovered using the functions $\mathcal{F}_1, \cdots, \mathcal{F}_t$, rather than by lookups.

Figure 5 shows the inter-group S-S key predistribution for sensors in group G_{22}. For simplicity, we only show the scenario when each group pair has one agent pair. Accordingly, each sensor is required to be preloaded with $\mu = 2$ keys shared with sensors in distinct groups.

2) S-S Key Establishment: A unique pairwise key is preloaded for every intra-group sensor pair. For a pair of neighboring sensors from different groups, we adopt the Highest Random Weight technique [26] to choose agents for path key generation, using a hash function $\mathcal{H}$ to realize distributed agreement. The work in [26] discusses how to select $\mathcal{H}$. Sensors $\langle G_u, s_i \rangle$ and $\langle G_v, s_j \rangle$ generate a path key as follows (Figure 6).

1) One principal, say $\langle G_u, s_i \rangle$, first computes $\mathcal{H}(s_i, s_j, p)$ for $1 \leq p \leq t$, and selects the p that yields the biggest $\mathcal{H}$ value. It now uses the function $\mathcal{F}_p$ to pick an associated sensor pair $\langle G_u, s_x \rangle$ and $\langle G_v, s_y \rangle$ for path key generation. Now, s_i then randomly generates a key K_{ij} and sends it to agent s_x, encrypted with the association key K_{ix} it shares with s_x.

$$s_i \rightarrow s_x : (K_{ij}, G_v)_{K_{ix}}$$

2) Upon receipt, s_x decrypts this message and re-encrypts it with the association key K_{xy} it shares with s_y, and sends it to s_y.

$$s_x \rightarrow s_y : (K_{ij})_{K_{xy}}$$

3) s_y decrypts this packet, re-encrypts it with the key K_{jy} it shares with s_j, and sends it to s_j.

$$s_y \rightarrow s_j : (K_{ij})_{K_{jy}}$$

4) s_j first applies $\mathcal{H}$ to select the same associated pair $\langle G_u, s_x \rangle$ and $\langle G_v, s_y \rangle$ that s_i selected for path key establishment. It then recovers K_{ij} using K_{jy}, its preloaded association key with s_y.

C. Key Establishment Between Mobile Collectors and Sensors

1) M-S Key Predistribution: We say $\langle G_u, s_i \rangle$ is an agent for mobile collector m_i in G_u, if m_i and $\langle G_u, s_i \rangle$ are associated.

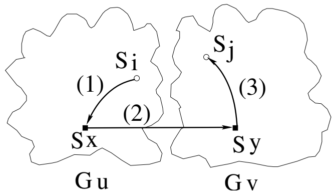


Fig. 6. Inter-Group S-S Key Establishment

Mobile Collectors with $O(n_s)$ Memory: In this case, static sensors are memory-limited but mobile nodes are not. Each sensor s_j is preloaded with a secret key K_{s_j} shared pairwise with the base station. Sensor s_j communicates securely with mobile collector m_i using key $K_{ij} = \mathcal{R}(K_{s_j}, m_i)$, where $\mathcal{R}$ is a pseudo-random function (PRF) [10]. Each mobile collector m_i is preloaded with the set of keys $\{K_{ij}\}$ for all sensors s_j. Sensor s_j can compute a unique pairwise key shared with every mobile collector m_i on-demand. However, mobile collectors have enough memory to store the keys they need. While $\mathcal{R}$ may be easy to compute, the overhead can be high if the number of mobile collectors is high.

Mobile Collectors with Limited Memory: We create associations between each mobile collector m_i and sensors from some selected g' groups, in a pattern that ensures that m_i is t-associated with each of the g' groups. The g' groups can be selected using g' functions analogous to the $\mathcal{F}_i$ functions defined in Section V-B.1, to ensure that each group is likely to be chosen by the same number of mobile collectors. This balances loads and creates no high-value targets, since no group of sensors hold more keys than any other.

Also, agents for mobile collectors can be chosen using functions $\mathcal{F}'_i$ analogous to the functions $\mathcal{F}_i$ in Section V-B.1, in whose definition we can treat m_i as a group. The function $\mathcal{F}'_i(G_u, m_i)$ is used to select the ith agent for m_i in G_u.

2) M-S Key Establishment: A mobile collector m_i and one of its non-associated neighbor $\langle G_u, s_i \rangle$ generate a path key as follows. If m_i has agents in G_u, we use Highest Random Weight technique as in Section V-B.2 to choose an agent for path key generation. Otherwise, m_i finds an agent in an adjacent group (say G_v), and uses that agent and the agent pair between G_u and G_v as intermediaries to establish path keys. To further reduce the communication overheads at sensors, we may allow m_i to move to the agent.

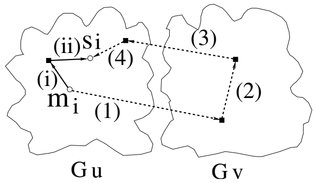


Fig. 7. M-S Key Establishment

D. Features of mGKE

- *Resilience to impersonation:* All messages above are secured with the preloaded pairwise keys shared between the sender and the receiver. It is hence impossible for an attacker to impersonate the intermediaries, since it does not have the preloaded keys.

- *Failure resilience:* We can use the techniques in [26] to guarantee resilience. Each pair of groups has $t = \lceil \frac{\mu\gamma}{g-1} \rceil$ agent pairs, and each pair of mobile collector and static sensor group also is t-associated. If there are $n = 10,000$ sensors arranged into $g = 100$ groups, and each sensor is preloaded with $\mu = 20$ pairwise keys shared with sensors in other groups, we will get $t = 20$ agent pairs for each pair of groups. Let a pair of intermediaries be selected for a path key request using function $\mathcal{H}$. If this agent pair fails, we simply select the pair corresponding to the index q that yields the second biggest $\mathcal{H}$ value, and use $\mathcal{F}_q$ to determine the new agent pair for path key generation. We can continue until we find an agent pair that is alive.
- *Routing protocol:* Routing is an issue orthogonal to our work. PIKE uses the geographic routing protocol GPSR [15] with a globally addressable infrastructure GHT [22] to find routes to the intermediate nodes. mGKE can also use GPSR and GHT to find the route either from a static/mobile node to the agent, or between the agents. However, there is a major difference between PIKE and mGKE in this respect. Finding a route to the trusted intermediary nodes in PIKE involves network-wide route discovery, since these intermediaries may not always be in the vicinity. In contrast, the static/mobile node and the agent are either within the same group or within nearby groups in mGKE, so discovering a route to the agent only involves route discovery within the group or nearby groups. Route discovery between agents is also local since they are in adjacent groups. mGKE can accomplish key establishment even without a globally addressable infrastructure. The overhead of routing in mGKE is much smaller than that in PIKE.

VI. Evaluation Metrics

We evaluate mGKE in terms of security and performance. We measure security in terms of resilience against node capture and connectivity, and measure performance in terms of communication and memory overhead.

A. Security Metrics

1) Resilience: This metric measures how the capture of some sensors affects the security of the *rest* of the network. Let U be the set of uncompromised sensors. Let $L(U)$ and $\hat{L}(U)$, respectively, be the sets of total and compromised links between sensors in the set U. Resilience is defined as the ratio $\hat{L}(U)/L(U)$.

This definition of resilience is similar to those used in previous random key predistribution schemes [9], [5], [7], [17], [6], [12]. However, our meanings of *link* is different. In our definition, a link is secured either by a preloaded pairwise key or by a path key. In contrast, the previous schemes consider only the links secured by preloaded keys [9], [5], [6] or keys derived from preloaded key space [7], [17], [12].

As [7] points out, a path key is compromised if an attacker can decipher the messages during key establishment or compromise any of the intermediaries. It is hence important to consider the security of path keys to properly evaluate the effects of sensor compromise. In Section VII-A, we present some analytical and simulation results of resilience, considering the security of both preloaded keys and path keys.

2) Connectivity: Connectivity is defined as the probability that a sensor network is securely connected. In Section VII-B, we show that the mGKE scheme can enable a sensor network securely connected with 100% probability, as long as the network is physically connected.

B. Performance Metrics

1) Communication Overhead: We measure communication overhead as the average number of hops that messages must be transmitted to establish a S-S key or a M-S key. We compare the communication overhead for establishing a S-S key in mGKE with that of PIKE, since only mGKE and PIKE show graceful security degradation as the number of compromised nodes increases (see SectionVII-A). For both mGKE and PIKE, we only measure the communication overhead of transmitting the encrypted path keys among the nodes, neglecting the routing communication overhead. First, as indicated in [4], the routing communication highly depends on the underlying routing protocol, which is out of the scope of our paper. Second, as analyzed in Section V-D, with the same routing protocol, mGKE will introduce smaller routing communication overhead than PIKE. Therefore, neglecting the communication overhead of routing for both mGKE and PIKE does not favour our scheme mGKE in any aspect, when compared to PIKE. Rather, such processing will help us focus on the efficiency of key establishment technique.

2) Memory Overhead: As in earlier schemes, we quantify memory overhead in terms of the number of keys preloaded into each sensor. We do not count the temporary storage during the execution of our scheme, or the memory to store the newly established pairwise keys.

C. System Setting

We used the following configuration in our analysis and simulations. The number of static sensors n_s varied between 10,000 and 50,000, with 10,000 being the default value. The number of mobile collectors n_m was 100. The wireless communication range for each sensor was $40m$. The deployment density δ, the average number of static sensors in a sensor's transmission range, varied from 20 to 100, to represent low- to high-density deployments. The deployment area A is determined by the number of static sensors n_s, sensor density δ, and the communication range. $A = \frac{n_s \pi r^2}{\delta}$. The group size γ was set to be 100 as previous group-based schemes [6], [12], and the number of groups varied from 100 to 500 accordingly. For simplicity, we assume that sensors in each group were uniformly distributed within a region of area A/g.

VII. Security Analysis

We now compare mGKE with SRKP [7], with the group-based key predistribution scheme in [6], and with PIKE [4] in terms of resilience against node capture and connectivity.

A. Resilience Properties

In mGKE, only static sensors may act as agents. Hence, compromising a mobile collector will reveal its own keys, but no other keys. That is, the compromise of a mobile collector will not affect the security of any node it is not associated with. Therefore, it suffices for us to focus on the resilience of our scheme in response to the compromise of static sensors.

1) S-S Keys Shared Between Static Sensors: Let s_i and s_j be two uncompromised static neighbors. Let L_{ij} be the communication link between them, and let K_{ij} be the key used to secure this link. Let $\mathbf{\Lambda(K_{ij})}$ be the event that K_{ij} is a preloaded key, and let $\mathbf{\Pi(K_{ij})}$ be the event that K_{ij} is a path key. Let $\mathbf{\bar{L}_{ij}}$ be the event that link L_{ij} is compromised, and $\mathbf{C(x)}$ be the event that x static sensors have been compromised. The probability that $\mathbf{\bar{L}_{ij}}$ has occurred given that x static sensors have been compromised is

$$\begin{aligned}\Pr[\mathbf{\bar{L}_{ij}}|\,\mathbf{C(x)}] =& \Pr[\mathbf{\bar{L}_{ij}}|\,\mathbf{C}(\mathbf{x}) \wedge \mathbf{\Lambda(K_{ij})}] \times \Pr[\mathbf{\Lambda(K_{ij})}] \\ &+ \Pr[\mathbf{\bar{L}_{ij}}|\,\mathbf{C}(\mathbf{x}) \wedge \mathbf{\Pi(K_{ij})}] \times \Pr[\mathbf{\Pi(K_{ij})}].\end{aligned}$$

Schemes such as [9], [5], [7], [17], [6], [12] consider only the links secured by preloaded keys in evaluating resilience. Since pairwise keys are established by randomly selecting them from a global pool, the compromise of one sensor may compromise a number of pairwise keys for other sensors.

This is impossible in mGKE since preloaded pairwise keys are unique. A link secured by a preloaded key can not be compromised unless one of its endpoints is compromised. Therefore, mGKE achieves *perfect* resilience against node capture by their definition.

By our definition of resilience, for mGKE,

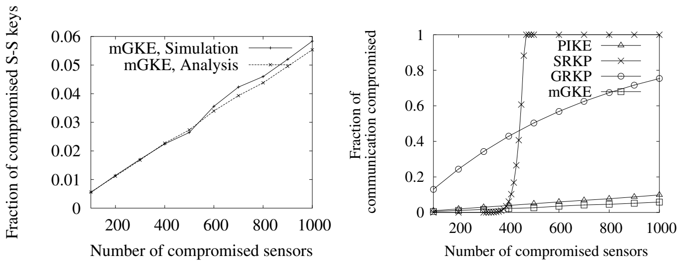
$$\Pr[\mathbf{\bar{L}_{ij}}|\,\mathbf{C(x)}] = \Pr[\mathbf{\bar{L}_{ij}}|\,\mathbf{C}(\mathbf{x}) \wedge \mathbf{\Pi(K_{ij})}] \times \Pr[\mathbf{\Pi(K_{ij})}].$$

Now, $\Pr[\mathbf{\Pi(K_{ij})}]$ is simply the ratio of the number of path keys to the total number of keys among all pairs of neighboring sensors. Let $\mathbf{\Pi_2(K_{ij})}$ be the event that the path key K_{ij} is generated using two agents, and $\mathbf{\Pi_1(K_{ij})}$ be the event that the path key K_{ij} is generated using a single agent, as in the case when s_i or s_j is itself the agent for the other's group. Now,

$$\begin{aligned}&\Pr[\mathbf{\bar{L}_{ij}}|\,\mathbf{C(x)}] = \\ &\big(\Pr[\mathbf{\bar{L}_{ij}}|\,\mathbf{C}(\mathbf{x}) \wedge \mathbf{\Pi_1(K_{ij})}] \times \Pr[\mathbf{\Pi_1(K_{ij})}] \\ &\;+ \Pr[\mathbf{\bar{L}_{ij}}|\,\mathbf{C}(\mathbf{x}) \wedge \mathbf{\Pi_2(K_{ij})}] \times \Pr[\mathbf{\Pi_2(K_{ij})}]\big) \times \Pr[\mathbf{\Pi(K_{ij})}].\end{aligned}$$

Let there be g groups each of size γ, and let each sensor hold μ preloaded keys for sensors in other groups. As shown in Section V-B.1, each group has $t = \frac{\mu\gamma}{g-1}$ agents in every other group. If α is the probability that either s_i or s_j is the agent of its neighbor's group, then

$$\alpha = \frac{\binom{\gamma-1}{t-1}}{\binom{\gamma}{t}} = \frac{t}{\gamma}$$



(a) mGKE: Analysis vs. Simulation

(b) mGKE vs. SRKP & GRKP & PIKE

Fig. 8. Links compromised between uncompromised sensors ($n_s = 10^4$, $\delta = 50$)

$\Pr[\mathbf{\Pi_1(K_{ij})}]$ is equivalent to the probability that one endpoint of the link L_{ij} is the agent for the other group, while the other endpoint is not. Thus we get

$$\Pr[\mathbf{\Pi_1(K_{ij})}] = 2\alpha(1-\alpha).$$

Similarly, $\Pr[\mathbf{\Pi_2(K_{ij})}]$ is equivalent to the probability that neither s_i nor s_j is an agent for the other group, and can be computed as

$$\Pr[\mathbf{\Pi_2(K_{ij})}] = (1-\alpha)^2.$$

Now, $\binom{n-3}{x}/\binom{n-2}{x}$ is the probability that the agent used to transmit the path key K_{ij} is not compromised, when s_i and s_j are uncompromised, but x other sensors are compromised. Thus $\Pr[\mathbf{\bar{L}_{ij}}|\,\mathbf{C}(\mathbf{x}) \wedge \mathbf{\Pi_1(K_{ij})}]$ can be computed as

$$\Pr[\mathbf{\bar{L}_{ij}}|\,\mathbf{C}(\mathbf{x}) \wedge \mathbf{\Pi_1(K_{ij})}] = 1 - \frac{\binom{n-3}{x}}{\binom{n-2}{x}} = \frac{x}{n-2}.$$

Similarly,

$$\Pr[\mathbf{\bar{L}_{ij}}|\,\mathbf{C}(\mathbf{x}) \wedge \mathbf{\Pi_2(K_{ij})}] \quad = \quad 1 - \frac{\binom{n-4}{x}}{\binom{n-2}{x}} = 1 - \frac{(n-x-2)^{\underline{2}}}{(n-2)^{\underline{2}}},$$

where $a^{\underline{k}}$ is the falling factorial function $a(a-1)\cdots(a-k+1)$. Therefore,

$$\begin{aligned}\Pr[\mathbf{\bar{L}_{ij}}|\,\mathbf{C(x)}] = \Bigg\{&(1-\alpha)^2\left(1 - \frac{(n-2-x)^{\underline{2}}}{(n-2)^{\underline{2}}}\right) \\ &+2\alpha(1-\alpha)\left(\frac{x}{n-2}\right)\Bigg\} \times \Pr[\mathbf{\Pi(K_{ij})}].\end{aligned}$$

$\Pr[\mathbf{\Pi(K_{ij})}]$, the ratio of the number of path keys to the total number of keys among all pairs of neighboring sensors, is dependent on sensor deployment. Based on our simulation results, Figure 14(c) plots $\Pr[\mathbf{\Pi(K_{ij})}]$ in mGKE.

Figure 8(a) shows that our analytical and experimental results for the number of compromised links match each other closely. Figure 8(b) compares the resilience of mGKE with that of SRKP [7], the group-based random key predistribution scheme (GRKP) in [6], and PIKE [4]. We compute the resilience of SRKP using the analysis in [7], preloading each sensor with 200 keys drawn from 4 key spaces randomly chosen from 50 key spaces. We compute the resilience of the GRKP scheme in [6] using their analysis, with a key space size of 100,000 and connectivity of 99.99% [6]. (The connectivity of mGKE is 100%. See Section VII-B.) We note that the analysis in [6] only considers links secured by preloaded keys, so that the fraction of compromised links in [6] will be even higher if we consider path keys as well. For mGKE and PIKE, we consider the security of both preloaded keys and path keys. If only the links secured by preloaded keys were considered, the resilience graphs of PIKE and mGKE would both be lines of zero, representing perfect resilience against node capture.

Figure 8(b) shows that when around 350 of 10,000 sensors are compromised, the resilience of SRKP decreases dramatically. In contrast, both PIKE and mGKE shows graceful degradation of resilience with respect to the number of compromised sensors, so that attackers are unable to compromise a large fraction of other communication links by compromising a small number of sensors.

Figure 8(b) also shows that the resilience of mGKE is about twice as high as that of PIKE, since a significantly larger fraction of links are secured by pairwise keys in mGKE (see Section VIII-B). In Section VIII-B, we further show that mGKE achieves this improvement of resilience with significantly lower communication overhead than PIKE.

2) M-S Keys Shared Between Mobile and Static Nodes: Let s_i be an uncompromised sensor in group G_u, and m_j be a neighboring uncompromised mobile collector. As with static sensor pairs, the link L_{ij} between nodes s_i and m_j will be compromised only when K_{ij} is a path key established using a compromised sensor. Let $\mathbf{\Pi_a(K_{ij})}$ be the event that m_j is associated with G_u but s_i is not associated with m_j. In this case, K_{ij} is a path key established through a sensor s_k, $k \neq i$ in G_u. Let $\mathbf{\Pi_{\bar{a}}(K_{ij})}$ be the event that m_j is not associated with G_u, so that K_{ij} must be established using an intermediary $s_k \in G_v$ with which m_j is associated, and G_v is a nearby group. The probability of $\mathbf{\bar{L}_{ij}}$ occurring with x sensors compromised is

$$\begin{aligned}\Pr[\mathbf{\bar{L}_{ij}} |\ \mathbf{C(x)}] = &\Pr[\mathbf{\bar{L}_{ij}} |\ \mathbf{C(x)} \wedge \mathbf{\Pi_a(K_{ij})}] \times \Pr[\mathbf{\Pi_a(K_{ij})}] \\ &+ \Pr[\mathbf{\bar{L}_{ij}} |\ \mathbf{C(x)} \wedge \mathbf{\Pi_{\bar{a}}(K_{ij})}] \times \Pr[\mathbf{\Pi_{\bar{a}}(K_{ij})}].\end{aligned}$$

Since m_j is associated with g' out of g groups, we get

$$\Pr[\mathbf{\Pi_a(K_{ij})}] = \left(\frac{g'}{g}\right) \text{ and } \Pr[\mathbf{\Pi_{\bar{a}}(K_{ij})}] = 1 - \frac{g'}{g}.$$

Proceeding as in the analysis for L_{ij} between static sensors, we get

$$\begin{aligned}\Pr[\mathbf{\bar{L}_{ij}} |\ \mathbf{C(x)} \wedge \mathbf{\Pi_a(K_{ij})}] &= (1-\alpha)\left(1 - \frac{\binom{n-2}{x}}{\binom{n-1}{x}}\right) \\ &= (1-\alpha)\frac{x}{n-1}\end{aligned}$$

and

$$\begin{aligned}\Pr[\mathbf{\bar{L}_{ij}} |\ \mathbf{C(x)} \wedge \mathbf{\Pi_{\bar{a}}(K_{ij})}] &= \alpha^2\left(1 - \frac{\binom{n-2}{x}}{\binom{n-1}{x}}\right) \\ + 2\alpha(1-\alpha)\left(1 - \frac{\binom{n-3}{x}}{\binom{n-1}{x}}\right) &+ (1-\alpha)^2\left(1 - \frac{\binom{n-4}{x}}{\binom{n-1}{x}}\right).\end{aligned}$$

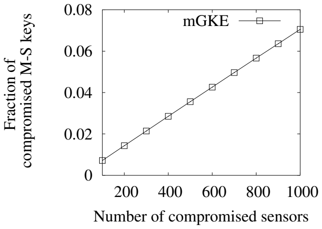


Fig. 9. Links compromised between uncompromised MCs and sensors ($n_s = 10,000$, $\delta = 50$)

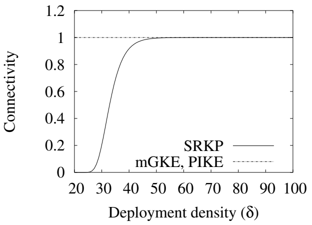


Fig. 10. Connectivity

Combining these expressions, simplifying binomial coefficients, and using the falling factorial notation, we get

$$\begin{aligned}&\Pr[\mathbf{\bar{L}_{ij}} |\ \mathbf{C(x)}] = \frac{g'}{g}(1-\alpha)\frac{x}{n-1} \\ &+ \left(1 - \frac{g'}{g}\right)\left\{\alpha^2\frac{x}{n-1} + 2\alpha(1-\alpha)\left(1 - \frac{(n-k-1)^{\underline{2}}}{(n-1)^{\underline{2}}}\right)\right. \\ &\left. +(1-\alpha)^2\left(1 - \frac{(n-k-1)^{\underline{3}}}{(n-1)^{\underline{3}}}\right)\right\}\end{aligned}$$

Figure 9 shows how the resilience for links between uncompromised mobile and static nodes changes with the number of compromised static sensors, when $n_s = 10,000$, $\delta = 50$, $\frac{g'}{g} = 0.3$, $\alpha = \frac{t}{\gamma} = 0.1$. It is not meaningful to compare our scheme with the SRKP and PIKE schemes. SRKP shows a dramatic degradation in resilience even for the static case, and this will remain true if it were applied to the mobile case. PIKE uses a globally addressable infrastructure to find intermediaries, and can not be directly adopted to support mobile sensor networks. As Figure 9 shows, in mGKE, the resilience of links between static and mobile nodes degrades linearly with the number of compromised static sensors, which is the best theoretically possible.

B. Connectivity

RKP and SRKP require high density deployments to ensure the entire sensor network is securely connected with high probability [13], [4]. In contrast, mGKE ensures that any two neighboring sensors are able to establish a path key, regardless of the sensor density or distribution, as long as the sensor network is physically connected. This guarantee is achieved since (1) any pair of sensors from the same groups have preloaded pairwise keys, (2) sensors from associated groups are able to establish path keys, and (3) the inter-group S-S key predistribution scheme ensures that any two groups are t-associated (see Section V-B.1).

Figure 10 compares the connectivity of SRKP, PIKE and mGKE in a 10,000-sensor network. For SRKP, each sensor has 4 key spaces chosen from a pool of 50 key spaces, and preloaded with 200 keys. This is a typical configuration from [7].

As Figure 10 clearly shows, the connectivity of SRKP decreases dramatically when the sensor density is less than 50, and is almost surely disconnected when the density is around

25. In contrast, PIKE and mGKE retains full connectivity regardless of sensor density. Remarkably, only 55 keys are required for the mGKE scheme to achieve full connectivity among static sensors when any pair of groups are 10-associated (See Section VIII-A).

VIII. Performance Evaluation

As shown in Section VII, PIKE and mGKE have substantially better resilience against node compromises than random key predistribution schemes, and guarantee that any two neighbors can establish a path key if needed. We now compare PIKE and mGKE in terms of memory and communication overhead.

A. Memory Overhead

The mGKE scheme imposes low memory requirements. If a sensor network has n_s static sensors, with group size γ, mGKE requires each sensor to be preloaded with $\gamma - 1$ pairwise keys shared with sensors from the same group and $t(g-1)/\gamma$ pairwise keys shared with sensors in different groups. Further, we use the method in [4] to reduce the memory requirement by a factor of two. Therefore, the memory needed per sensor to establish S-S key is $\lceil \frac{1}{2}(\gamma - 1) \rceil + \lceil \frac{(n-\gamma)t}{2\gamma^2} \rceil$ keys. To establish M-S keys with n_m mobile collectors, each of which is t-associated with g' groups, each sensor must also be preloaded with an additional $\lceil \frac{g' t n_m}{2 n_s} \rceil$ keys shared with mobile collectors.

In contrast, the memory overheads of PIKE-2D and PIKE-3D are $\lceil \sqrt{n_s} \rceil + 1$ and $3 \lceil \sqrt[3]{n_s} \rceil + 1$ respectively [4]. As noted in Section V-D, PIKE can not be directly adopted to support mobility, since it requires a globally addressable infrastructure. Figure 11 shows the memory requirements of PIKE-2D, PIKE-3D and mGKE (t=10). For mGKE, the solid line shows the memory overhead for supporting static sensors only, while the dashed line shows the memory needed to support mobile sensor networks with $g'/g = 0.3$.

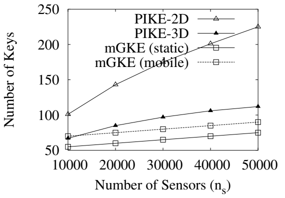


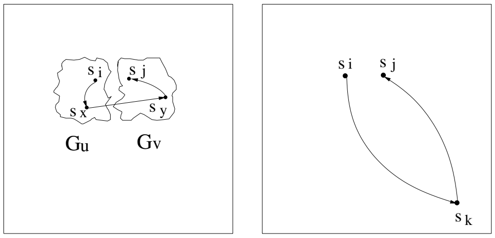
Fig. 11. Memory Requirements.

B. Communication Overhead for S-S keys

Messages are transmitted in mGKE only for path key establishment. Let H denote the average number of hops that a message traverses when any path key K_{ij} is established. Therefore, the average communication overhead is simply $H \times \Pr[\mathbf{\Pi}(\mathbf{K_{ij}})]$.

Two major differences between PIKE and mGKE result in a big difference in their communication overheads. First, sensors use local intermediaries when establishing path keys in mGKE, so only local communication is needed to transmit key establishment messages. In contrast, intermediaries in PIKE could be *anywhere* in the entire target region, so that network-wide communication is required.

Second, the fractions of keys that are path keys is much higher in PIKE than in mGKE. Sensors are deployed in groups in mGKE, so that sensors from the same group are more likely to be neighbors. In mGKE, all pairs of sensors from the same group are preloaded with pairwise keys. In PIKE, only sensors on the same grid column or row share preloaded pairwise keys. No deployment knowledge can be predetermined on constructing the grid makes the fraction of path keys in PIKE significantly higher than that of mGKE.



(a) mGKE (b) PIKE

Fig. 12. Path Key Establishment in mGKE and PIKE

1) Communication Overhead for an S-S Path Key Establishment: Establishing a path key between s_i and s_j in mGKE (see Figure 12(a)) requires messages from s_i to s_x, from s_x to s_y, and from s_y to s_j. If $h(s_p, s_q)$ denotes the hop distance between s_p and s_q, the number of hops required for path key establishment is $H(s_i, s_j) = h(s_i, s_x) + h(s_x, s_y) + h(s_y, s_j)$. If H_{mGKE} is the expected number of hops for path key establishment in mGKE, linearity of expectation leads to

$$H_{mGKE} = 2 * \bar{h}_{mGKE} + \bar{h'}_{mGKE},$$

where $\bar{h}_{mGKE}$ is the expected hop distance between any two nodes in a group, and $\bar{h'}_{mGKE}$ is the expected hop distance between any two sensors from adjacent groups.

Establishing a path key between neighboring sensors s_i and s_j in PIKE (Figure 12(b)) includes the round trip from the neighboring sensors to the intermediary s_k, who may be anywhere in the region. The number of hops required is $h(s_i, s_k) + h(s_k, s_j)$. If $\bar{h}_{PIKE}$ is the average hop distance between any two nodes in the entire region, the expected communication overhead in PIKE is

$$H_{PIKE} = 2 * \bar{h}_{PIKE}$$

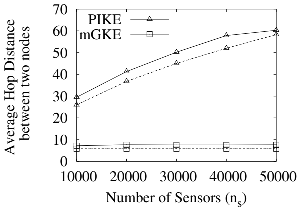
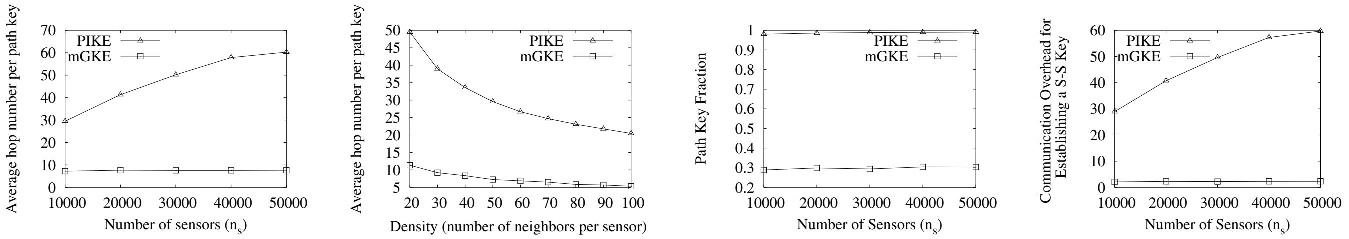


Fig. 13. $\bar{h}_{mGKE}$ and $\bar{h}_{PIKE}$.

Next, we give lower bounds for $\bar{h}_{PIKE}$ and for $\bar{h}_{mGKE}$. If two nodes are separated by physical distance $\bar{\lambda}$, we will need at least $\bar{\lambda}/r$ hops, where r is the transmission radius. Therefore, we can use $\bar{\lambda}/r$ as a lower bound for the average hop distance.

Due to space limitations, we only give the following results, and refer interested readers to [33] for details.

$$\begin{aligned} H_{mGKE} &= 2 * \bar{h}_{mGKE} + \bar{h'}_{mGKE} \\ &\geq \frac{1.04\sqrt{A/g}}{r} + \frac{4.35A/g + 1.46\pi r\sqrt{A/g}}{(4\sqrt{A/g} + \pi r)r}, \end{aligned}$$



(a) Avg hop count vs. network size ($\delta = 50$)

(b) Avg hop count vs. density ($n_s = 10,000$)

(c) Path-key fraction ($\delta = 50$)

(d) Communication overhead ($\delta = 50$)

Fig. 14. Average hop count, path-key fraction (S-S Keys) and communication overhead.

and

$$H_{PIKE} = 2 * \bar{h}_{PIKE} \geq \frac{1.04\sqrt{A}}{r}.$$

In Figure 13, the solid line shows the experimental results and the dashed line shows theoretical lower bound $\bar{h}$ for PIKE and mGKE, using a density $\delta = 50$. For both schemes, the experimental results match the lower bound quite closely. Therefore, we may use this lower bound to approximate $\bar{h}$.

Figure 14(a) plots simulation results for the average number of hops H to establish a path key (S-S key) in PIKE and mGKE, varying the network size from $10,000$ to $50,000$, for a density of 50. For a fixed group size, H_{mGKE} remains constant as the network grows, indicating that network size has no impact on the expected communication overhead. This is because the communication of establishing a path key in mGKE is localized to two adjacent groups. In contrast, establishing a path key in PIKE requires network-wide communication, and thus H_{PIKE} increases as the network size increases.

Figure 14(b) plots the average number of hops for establishing a path key, for network densities from 20 to 100, for a network size of 10,000. The number of neighbors increases with network density, so the average hop counts decrease in both PIKE and mGKE. Again, mGKE requires much lower communication overhead to establish path keys than PIKE.

2) Fraction of S-S Keys Which are Path Keys: Let the path key fraction be the fraction of S-S keys which are path keys. Figure 14(c) shows the path key fraction in PIKE and mGKE, respectively. Almost all (about 99%) of the links in PIKE are secured by path keys. This is expected, since only sensors at the same column or row of the logical grid have preloaded keys. This logical grid used to predistribute pairwise keys, includes no deployment information, so that sensors sharing preloaded keys are rarely neighbors. In contrast, although deployment information is not available in mGKE, sensors in the same group, which are preloaded with pairwise keys shared with one another, are more likely to be neighbors. As a result, mGKE has a much smaller ratio of path keys, around 30%.

3) Communication Overhead: We plot the communication overhead, which is $H \times \Pr[\mathbf{\Pi}(\mathbf{K_{ij}})]$, in Figure 14(d). Clearly, mGKE reduces the communication overhead by a factor of about 6 for a network of size $10,000$, with the improvement proportional to the network size. This demonstrates that mGKE is especially suitable for very large sensor networks.

C. Communication Overhead for M-S keys

Establishing a key between $\langle G_u, s_i \rangle$ and m_j requires two intra-group messages when G_u is associated with m_j, or two inter-group messages and two intra-group messages when G_u is not associated with m_j. Let $\frac{g'}{g} = 0.3$. Then, m_j is associated with G_u with probability $\Pr[\mathbf{\Pi_a}(\mathbf{K_{ij}})] = \frac{g'}{g} = 0.3$. Otherwise m_j is associated with at least one of G_u's eight adjacent groups with probability $P_2 = 1 - \Pr[\mathbf{\Pi_a}(\mathbf{K_{ij}})] - (1 - \Pr[\mathbf{\Pi_a}(\mathbf{K_{ij}})])^9 = 0.66$. Since these two cases happens with probability close to 1, we consider only these two cases to simplify our analysis.

In the first case, the two intra-group messages require on average $\bar{h}_1 = 1.04\sqrt{A/g}/r$ hops. For the second case, the two inter-group messages require $\bar{h}_2 = 2.57\sqrt{A/g}/r$. The derivations of $\bar{h}_1$ and $\bar{h}_2$ are detailed in [33]. Now, the average number of hops to establish an M-S key is $\Pr[\mathbf{\Pi_a}(\mathbf{K_{ij}})] \times \bar{h}_1 + P_2 \times \bar{h}_2$.

To demonstrate the feasibility of our scheme in support of mobile sensor networks, we will evaluate the fraction of total available energy in the sensor networks consumed to establish the keys between mobile collectors and static sensors. Suppose the network has $n_s = 10,000$ static sensors, divided into $g = 100$ groups with size $\gamma = 100$. The number of mobile collectors is $n_m = 100$. The region is a $1,000m \times 1,000m$ square, divided into 100 subregion of size $100m \times 100m$. Mobile collectors move at constant speed $v = 10m/s$, and pause $w = 5s$ at waypoints. Let $\frac{g'}{g} = 0.3$, and the transmission radius is $r = 40m$.

Now, from our analysis, we know the average number of hops per M-S key is approximately 7. When a mobile collector moves to a subregion, it will establish 100 keys with all the static sensors in that subregion. As analyzed in Section IV, the average time for each data collection is $E[t_i] = 57s$. Therefore, on average, every $57s$ all the 100 mobile collectors will establish in total $100 \times 100 = 10^4$ keys, or require 7×10^4 hops transmission.

Suppose the energy consumption to transmit a packet per hop is approximately $0.48mJ$ [19]. The total energy consumption for key establishment will be $7 \times 10^4 \times 0.48 = 3.36 \times 10^4 mJ$. Suppose each sensor has two AA batteries, each with average capacity 2,850 mAh [20]. Now, the total energy capacity of 10,000 sensors would be $10,000 \times 2,850 \times 2 =$

5.7×10^7 mAh. This amount of capacity will allow the sensor network alive for about $57s \times (3 \times 5.7 \times 10^7 \times 3600)/(3.36 \times 10^4) = 12087$ days when the energy is only used for M-S key establishment.

We have not considered the routing overhead or packet losses. However, these numbers clearly suggest that our scheme is feasible to support the key establishment in a typical mobile sensor networks.

IX. Conclusions

In this work, we have addressed secure data collection and secure communication in mobile sensor networks. We first analyzed the impact of mobile collector compromises, and the circumstances under which reliability can be guaranteed. Our analysis shows that mobility can in fact improve data consistency when mobile collectors may be compromised.

We also present mGKE, a new group-based key predistribution scheme for large sensor networks. mGKE has a number of advantages over current methods. First, it accommodates very flexible deployment models as well as mobility. Second, it enables any pair of neighboring sensors to establish a unique pairwise key, regardless of sensor density or distribution, making it suitable for a wide range of applications. Third, mGKE is nearly perfectly resilient against node capture attacks, due to the uniqueness of pairwise keys. Unlike SRKP, which also establishes unique pairwise keys, system security in mGKE does not degrade dramatically when the number of compromised sensors reaches a certain threshold. Instead, mGKE is remarkably resilient, and degrades gracefully. Finally, mGKE involves only local communication to establish pairwise keys, and has very low communication overhead.

Acknowledgment

This work is supported in part by grants from Tata Consultancy Services, Inc., and the Fault-Tolerant Networks program of Defense Advanced Research Projects Agency, under contract F30602-01-2-0536.

References

[1] D. Balfanz, D. Smetters, P. Stewart, and H. Wong. Talking to strangers: Authentication in ad hoc wireless networks. In *NDSS*, 2002.

[2] C. Bettstetter, H. Hartenstein, and X. P. Costa. Stochastic properties of the random waypoint mobility model. *Wireless Networks*, 10(5):555–567, 2004.

[3] S. Capkun, J.-P. Hubaux, and L. Buttyan. Mobility helps security in ad hoc networks. In *MobiHoc*, 2003.

[4] H. Chan and A. Perrig. PIKE: Peer intermediaries for key establishment in sensor networks. In *INFOCOM*, 2005.

[5] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, 2003.

[6] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In *INFOCOM*, 2004.

[7] W. Du, J. Deng, Y. Han, and P. Varshney. A pairwise key predistribution scheme for wireless sensor networks. In *ACM CCS*, 2003.

[8] Erdos and Renyi. On random graphs. In *I. Publ. Math. Debrecen*, 1959.

[9] L. Eschenaer and V.D.Gligor. A key-management scheme for distributed sensor networks. In *ACM CCS*, 2002.

[10] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[11] G. Grimmett and D. Strirzaker. *Probability and Random Processes*. Oxford Press, 2001.

[12] D. Huang, M. Mehta, D. Medhi, and L. Harn. Location-aware key manageent scheme for wireless sensor networks. In *ACM Workshop SASN*, 2004.

[13] J. Hwang and Y. Kim. Revisiting random key pre-distribution schemes for wireless sensor networks. In *ACM Workshop SASN*, 2004.

[14] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin. Intelligent fluid infrastructure for embedded networks. In *MobiSys*, 2004.

[15] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *MobiCom*, 2000.

[16] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiquitous security support for mobile ad hoc networks. In *ICNP*, 2001.

[17] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In *ACM CCS*, 2003.

[18] D. Liu and P. Ning. Location-based pairwise key establishments of static sensor networks. In *ACM Workshop SASN*, 2003.

[19] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In *ACM SIGMOD*, 2003.

[20] D. J. Malan, M. Welsh, and M. D. Smith. A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography. In *SECON*, 2004.

[21] UC Berkey The EECS department. Cotbots: The mobile mote-based robots. `http://www-bsac.eecs.berkeley.edu/projects/cotbots/`.

[22] S. Ratnasamy, B. Karp, L. Yin, D. Estrin, R. Govindan, and S. Shenker. GHT: a geographic hash table for data-centric storage. In *ACM Workshop WSNA*, 2002.

[23] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. In *Comm. of the ACM*, 1978.

[24] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme. Robomote: A Tiny Mobile Robot Platform for Large-Scale Sensor Networks. In *IEEE ICRA*, 2002.

[25] I. C. Technology. MICA2: Wireless Measurement System. `http://www.xbow.com/Product/pdf/files/wireless_pdf/6020-0042-0_MICA2.pd%f`.

[26] D. Thaler and C. V. Ravishankar. Using name-based mappings to increase hit rates. *IEEE/ACM Transactions on Networking.*, 6(1):1–14, 1998.

[27] Y. Tirta, Z. Li, Y. Lu, and S. Bagchi. Efficient collection of sensor data in remote fields using mobile collectors. In *ICCCN*, 2004.

[28] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh. Toward resilient security in wireless sensor networks. In *MobiHoc*, 2005.

[29] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *MobiCom*, 2002.

[30] F. Ye, H. Luo, S. Lu, and L. Zhang. Statistical en-route detection and filtering of injected false data in sensor networks. In *INFOCOM*, 2004.

[31] W. Zhang, G. Cao, and T. L. Porta. Data Dissemination with Ring-Based Index for Wireless Sensor Networks. In *ICNP*, 2003.

[32] W. Zhang, H. Song, S. Zhu, and G. Cao. Least privilege and privilege deprivation: towards tolerating mobile sink compromises in wireless sensor networks. In *MobiHoc*, 2005.

[33] L. Zhou, J. Ni, and C.V.Ravishankar. Supporting secure communication and data collection in mobile sensor networks. Technical report, Dept. of Computer Science and Engineering, UC, Riverside, June 2005.

[34] L. Zhou, J. Ni, and C. Ravishankar. Efficient Key Establishment for Group-Based Wireless Sensor Deployments. In *ACM Workshop WiSe*, 2005.

[35] L. Zhou, J. Ni, and C. Ravishankar. (SHORT PAPER) GKE: Efficient Group-based Key Establishment for Large Sensor Networks. In *SecureComm*, 2005.

[36] L. Zhou and C. Ravishankar. A Fault Localized Scheme for False Report Filtering in Sensor Networks. In *IEEE ICPS*, 2005.

[37] S. Zhu, S. Setia, S. Jajodia, and P. Ning. An interleaved hop-by-hop authentication scheme for filtering false data injection in sensor networks. In *IEEE Symposium on Security and Privacy*, 2004.