

Location-aware Key Establishment in Wireless Sensor Networks

Fang Liu
Department of Computer
Science
George Washington University
Washington, DC 20052
fliu@gwu.edu

Major Jose "Manny"
Rivera
Architecture Operation
Network and Space (AONS)
US Army Chief Information
Office, G6
jose.rivera@us.army.mil

Xiuzhen Cheng
Department of Computer
Science
George Washington University
Washington, DC 20052
cheng@gwu.edu

ABSTRACT

Due to its efficiency, symmetric key cryptography is very attractive in sensor networks. A number of pairwise key pre-distribution protocols have been proposed, but the scalability is often constrained by the conflict between the desired probability of sharing keys between two nodes and the resilience against node capture attack under a given storage capacity for key-related information within each sensor. In this paper, we propose LKE, a self-configuring scheme for location-aware key establishment that targets resource-constrained sensor networks. In LKE, location information is employed for a deterministic key space generation and keying information distribution. LKE exhibits "perfect" security in against node capture attack and achieves high connectivity (close to 1) in the induced key-sharing graph at the expense of a small amount of memory in worker sensors. LKE is a pure in-situ key establishment scheme, which scales well to very large sensor networks.

Categories and Subject Descriptors

C.2.0 [Computer-Communication Networks]: General—*Security and protection*; C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Wireless communication*

General Terms

Security, Design, Algorithm

Keywords

Wireless Sensor Networks, Security, In-Situ Key Establishment

1. INTRODUCTION

Secure communication is a critical requirement for many sensor network applications. Nevertheless, the constrained capabilities of smart sensors (batter supply, CPU, memory, etc.) and the harsh deployment environment of a sensor network (wireless, ad hoc, etc.) make this problem very challenging [4]. A secure sensor network

requires a "sound" key establishment scheme that should be easily realized by individual sensors, should be localized to scale well to large sensor networks, should require small amount of space for keying information storage, and should be resilient against node capture attacks.

Due to its efficiency, symmetric key cryptography is very attractive in sensor networks. Researchers have proposed a number of pairwise key establishment protocols recently [5,7–9,12,13]. However, these methods may not scale well or may require strict deployment knowledge for better scalability. Further, they may require a prohibitive amount of keying information to be pre-loaded into the memory of a sensor, thus wasting storage space since many information may never be used during the lifetime of the sensor.

In this paper, we propose LKE, a self-configuring scheme for location-aware key establishment in large-scale sensor networks. In LKE, a fraction of sensors are self-elected to become *service nodes*, which are in charge of key space generation and keying information distribution. The majority of the sensors, namely *worker nodes*, get the keying information from service sensors in the neighborhood. Two worker sensors can compute a common key as long as they obtain the keying information from the same service node. The keying information computation and distribution are both based on location information through a deterministic procedure. LKE places no special requirement on worker sensors. It is a truly in-situ scheme for bootstrapping keys in sensor networks. Simulation study indicates that LKE achieves a high level of connectivity, with a tradeoff of a small amount of storage overhead per node.

This paper is organized as follows. Related work and network model are sketched in Section 2 and Section 3, respectively. LKE, the location-aware key establishment algorithm, is proposed in Section 4 and evaluated in Section 5. Finally, we conclude our paper in Section 6.

2. RELATED WORK

In this section, we summarize a number of most related works. For a more comprehensive literature survey, we refer the readers to [3].

The basic *random keys scheme* is proposed by Eschenauer and Glgor in [9], in which a large key pool \mathcal{K} is computed offline and each sensor picks k keys randomly from \mathcal{K} without replacement to form a key ring before deployment. Two sensors can establish secure communication as long as they have at least one common key in their key rings. An enhanced scheme is proposed in [5], which requires $q > 1$ number of common keys for two nodes to establish a shared key.

Copyright 2006 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by a contractor or affiliate of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
IWCMC'06, July 3–6, 2006, Vancouver, British Columbia, Canada.
Copyright 2006 ACM 1-59593-306-9/06/0007 ...\$5.00.

To improve security, two *random key spaces schemes* [7, 12] have been proposed. These two schemes are very similar in nature, except that the key spaces are defined differently. In both schemes, a number of key spaces are precomputed and each sensor is associated with one or more key spaces before deployment. Two sensors can compute a pairwise key after deployment if they have keying information from a common key space. In LKE no key space is precomputed. Compared with [7, 12], LKE achieves much better performance in connectivity of the key sharing graph and storage overhead, as indicated in our simulation study.

To achieve a better scalability, the *group-based schemes* [8, 14, 19] have been designed. Du *et al.* [8] employ a group deployment model and associate each group of sensors with a sub key space. Sub key spaces overlap if the corresponding groups are deployed at adjacent deployment points. In [14, 19], sensors are grouped based on IDs to form horizontal and vertical groups. Any pair of sensors belonging to the same group are preloaded with a unique key before deployment. Compared to the deployment knowledge based scheme [8], the schemes in [14, 19] release the strong topology assumption, but require flooding for path key establishment. Compared with [8, 14, 19], our scheme can support more efficient path key establishment. Further, LKE reduces the unpredictability of key predistribution since it is a pure in-situ key establishment scheme.

3. PRELIMINARIES, ASSUMPTIONS, AND MODELS

3.1 Preliminaries

Our scheme works fine with both key space models introduced by [2, 8]. We employ the polynomial key space model [2] as an example. A polynomial key space utilizes a bivariate λ -degree polynomial $f(u, v) = f(v, u) = \sum_{i,j=0}^{\lambda} a_{ij}u^i v^j$ over a finite field F_s , where s is a prime that is large enough to accommodate a cryptographic key. By plugging a value z (e.g. z can be the id, location, etc.) associated with a sensor, we obtain the *polynomial share* allocated to that sensor. In this paper, we choose $z = Hash(x_i, y_i)$, where (x_i, y_i) is the physical position of sensor i . Therefore sensor i receives the polynomial share $f(z, v)$ from the key space $f(u, v)$. Thus two sensors knowing each other's position information can compute a shared key if they have polynomial shares from the same key space.

3.2 Network Model

We consider a large-scale static sensor network deployed in outdoor environments. Sensors are able to position themselves through any of the techniques proposed in literature (e.g. [6, 15]), and they communicate with each other based on a geographic routing protocol (e.g. [10]). We also assume a secure location verification protocol (e.g. [11, 18]) is available which can detect false position claims.

We assume homogeneous sensors are densely deployed in a given region. The operation of the sensor network is unattended after the initial bootstrap procedure for sensor localization and key establishment is done. Sensors are preloaded with several system parameters, and are differentiated as either worker sensors or service nodes after deployment. Worker sensors are in charge of sensing and reporting data, and are expected to operate for years. Service sensors take charge of key space construction and keying information distribution. They may die after their duty is complete.

3.3 Adversary Model

We assume initial trust exists among sensors within a short period of time after deployment. An adversary can only passively

monitor a small proportion of the communications during the node self-configuration time, while no active attacks (flooding, jamming, spoofing, etc.) can be launched and no physical access to the network can be obtained during this initial period. This assumption is realistic, since the short configuration period represents a very small window of opportunity for an attacker when compared to the long network lifetime [1]. But after the initial bootstrap procedure, an adversary is capable of all possible attacks, such as eavesdropping on all traffic, injecting packets, replaying older messages, physically attacking nodes, etc. Once a node is compromised, all the information it holds will be released.

4. THE LOCATION-AWARE KEY ESTABLISHMENT SCHEME

LKE consists of four phases: Each sensor is preloaded with a bootstrap program and several system parameters during the *pre-distribution* phase, and is differentiated as either a service sensor or a worker node in the *node self-configuration* phase. A worker sensor first obtains a polynomial share from a service node through a secure channel in the *polynomial share distribution* phase, then computes pairwise keys shared with others in the *pairwise key establishment* phase.

4.1 Pre-distribution

In LKE, sensors have the same configuration (e.g. communication capability, resources, preloaded information, etc) before deployment. They take different roles and self-configure accordingly after deployment. The selection and configuration of service sensors are controlled automatically by a bootstrapping program that is preloaded into each sensor before deployment. This program also computes a polynomial key space for a service node.

Several pre-configured system parameters, listed in Table 1, are also pre-loaded to each sensor. The security parameter, t , is determined by the memory budget of sensors (to be explained in Section 5.2). L determines the coverage area of a service node, and δ is used for service node election. L and δ are initialized according to the following criteria:

$$\pi L^2 = t \times A/N, \quad (1)$$

$$\delta = R/\sqrt{5}, \quad (2)$$

where A is the size of the deployment region, N is the total number of nodes, R is the nominal transmission range. Note that these parameters can be estimated easily before deployment.

t	The collusion resistance degree of a key space
L	The size of a grid and the range covered by a key space
δ	The range for service node competition
T_0	The time for bootstrapping keys

Table 1: Preloaded System Parameters.

4.2 Node Self-Configuration

Right after deployment, a sensor positions itself and determines its role independently and locally according to the following algorithm.

A virtual grid is first computed based on location information. Assume a sensor S is located at (x, y) . The *home grid*, where S resides in, can be derived as (X, Y) with $X = \lfloor x/L \rfloor$, $Y = \lfloor y/L \rfloor$. The grid center lies at (X_0, Y_0) , where $X_0 = (X + 1/2) \times L$, $Y_0 = (Y + 1/2) \times L$. Then S computes its distance to (X_0, Y_0) .

If the distance is less than δ , S is eligible to compete for being a service node.

An eligible sensor first waits a random length of time. If the node receives no competition message from the others, it announces its decision to be a service node. Otherwise, the sensor just self-configures as a worker node. Note that all the eligible nodes are within δ -distance from the grid center. The setting of δ ensures that all eligible sensors within a grid can communicate with each other directly. This means the pre-configured δ value restricts the competition messages within a local range.

Whenever an eligible node succeeds in the competition, the preloaded bootstrapping program generates a prime number s and computes a symmetric bivariate t -degree polynomial $f_{X,Y}(x, y) = \sum_{i,j=0}^t a_{i,j}x^i y^j$ over a finite field $GF(s)$ serving as a key space for shared key establishment in the neighborhood of the service sensor. This program also generates two large distinct primes p and q satisfying $p \equiv q \equiv 3 \pmod{4}$. p and q constitute Rabin's public cryptosystem [17] with a public key $n = p \times q$ and a private key (p, q) , which will be used to establish a secure channel such that a polynomial share can be transferred securely from a service sensor to a worker node. All ineligible sensors and those that have failed in the competition configure themselves as worker nodes. The details of the node self-configuration process are elaborated in Algorithm 1.

Algorithm 1 Node Self-configuration

```

1: function  $\rho = \text{NodeConfig}(t, \delta, L, T_0)$     ▷  $\rho$ : the selected role
2:    $(x, y) \leftarrow \text{self-positioning}$           ▷ Localization
3:    $X \leftarrow \lfloor x/L \rfloor$                       ▷ Get grid id
4:    $Y \leftarrow \lfloor y/L \rfloor$ 
5:    $X_0 \leftarrow (X + 1/2) \times L$              ▷ Get grid center
6:    $Y_0 \leftarrow (Y + 1/2) \times L$ 
7:    $D \leftarrow \sqrt{(x - X_0)^2 + (y - Y_0)^2}$   ▷ Get distance
8:   if  $D \leq \delta$  then                        ▷ Eligible for the competition
9:      $TTL \leftarrow \text{rand}$                     ▷ Wait a random time
10:     $\text{elapse}(TTL)$ 
11:    if not  $\text{recv}(\text{competition\_msg})$  then
12:       $\text{broadcast}(\text{competition\_msg})$         ▷ Succeed
13:       $\rho \leftarrow \text{ServiceNode}$ 
14:       $\{s, p, q\} \leftarrow \text{getPrimes}$ 
15:       $f_{X,Y} \leftarrow \text{getPolynomial}(t, s)$   ▷ Get a symmetric
         $t$ -degree bivariate polynomial
16:       $\text{PSD}(f_{X,Y}, x, y, T_0)$                 ▷ Algorithm 2
17:    else                                     ▷ Fail the competition
18:       $\rho \leftarrow \text{WorkerNode}$ 
19:    end if
20:  else                                       ▷ Not eligible
21:     $\rho \leftarrow \text{WorkerNode}$ 
22:  end if
23:  return  $\rho$ 
24: end function

```

4.3 Polynomial Share Distribution

In the third phase, a public key assisted *Polynomial Share Distribution (PSD) protocol* is designed to securely disseminate polynomial shares from a service sensor to worker nodes in the neighborhood. PSD is composed of three steps:

4.3.1 Key Space Advertisement

A service node announces its existence through beacon broadcasting when its key space is ready. The beacon message includes (see Fig. 1): a) the key space id (X, Y) , which is also the grid id of

the service node, b) the location of the service node (x_0, y_0) , and c) the public key n , where $n = p \times q$. Nodes receiving the message the first time forward it if its distance to the grid center (X_0, Y_0) is $\leq L$.

id	$source$	n
------	----------	-----

id : the key space id, represented by the grid id (X, Y)
 $source$: the location of the source service node, (x_0, y_0)
 n : the public key, $n = p \times q$

Figure 1: The message format of the key space advertisement

4.3.2 Secure Channel Establishment

Any worker node receiving the key space advertisement first testifies the validity by checking whether the distance from the declared source position (x_0, y_0) to the grid center (X_0, Y_0) is smaller than δ . Recall that the underlying position verification scheme ensures that (x_0, y_0) is the real source position. Such verification prevents an adversary from infusing false information.

For each valid announcement, a computationally asymmetric channel based on Rabin's cryptosystem [17], which shifts the most computation to the service node, is established for polynomial share distribution. After obtaining the public key n , a worker sensor at (x_i, y_i) picks up a random number K_s and computes $E_n(K_s || B) = K_s^2 \pmod{n}$, where B is a predefined bit pattern for ambiguity resolution in Rabin's decryption. $E_n(K_s || B)$, along with (x_i, y_i) , is transmitted to the corresponding service node, which will compute $D_{p,q}(E_n(K_s || B))$ by applying the decryption algorithm. Now K_s is known to both the worker node and the service node, and can be used as the secret key of a secure channel for polynomial share dissemination in the next step.

4.3.3 Polynomial Share Acquisition

After agreeing on a shared key K_s with a worker node (x_i, y_i) , the service node first computes a location-aware polynomial share $f_{X,Y}^i = f_{X,Y}(k_i, y)$ where $k_i = \text{Hash}(x_i, y_i)$, then transmits $f_{X,Y}^i$ encrypted with K_s based on any popular symmetric encryption algorithm (AES, DES, etc). The behavior of a service node for polynomial share distribution is summarized by Algorithm 2. Any two worker nodes receiving polynomial shares from the same service node can compute a shared key directly for secure data exchange in the future.

Finally, after the timer T_0 expires, the key bootstrapping procedure terminates and service sensors erase their key space information.

4.4 Pairwise Key Establishment

LKE employs location information not only for service sensor election but also for polynomial share generation and distribution. Two sensors can determine whether they share a common key space based on their location information. Such a deterministic procedure results in an efficient pairwise key establishment scheme.

4.4.1 Direct Key Computation

Assume node i at (x_i, y_i) wants to communicate with node j at (x_j, y_j) , and i and j share at least one common key space.

- Node i selects one of the common key spaces, say (X, Y) , and computes $K_{ij} = f_{X,Y}^i(k_j) = f_{X,Y}(k_i, k_j)$, where $k_i = \text{Hash}(x_i, y_i)$, $k_j = \text{Hash}(x_j, y_j)$.

Algorithm 2 Polynomial Share Distribution

```

1: procedure PSD( $f_{X,Y}, x_0, y_0, T_0$ )  $\triangleright (x_0, y_0)$  is the position of
   the service node
2:    $TTL \leftarrow T_0$ 
3:    $n \leftarrow p \times q$ 
4:   Broadcast ( $x_0, y_0, n$ ) within  $L$ -distance  $\triangleright$  Key space
   advertisement
5:   while  $TTL > 0$  do
6:     if  $recv(request, x_i, y_i, E_n(K_s))$  then  $\triangleright$  Distribute
       polynomial share to node  $(x_i, y_i)$ 
7:        $K_s \leftarrow D_{p,q}(E_n(K_s))$   $\triangleright$  Decrypt  $K_s$ 
8:        $k_i \leftarrow Hash(x_i, y_i)$ 
9:        $f_{X,Y}^i(y) \leftarrow f_{X,Y}(k_i, y)$   $\triangleright$  Compute polynomial
       share for  $(x_i, y_i)$ 
10:       $send(x_0, y_0, E_{K_s}(f_{X,Y}^i(y)))$ 
11:    end if
12:     $elapse(TTL)$ 
13:  end while
14:   $Delete(f_{X,Y}(x, y))$   $\triangleright$  Expiration
15: end procedure

```

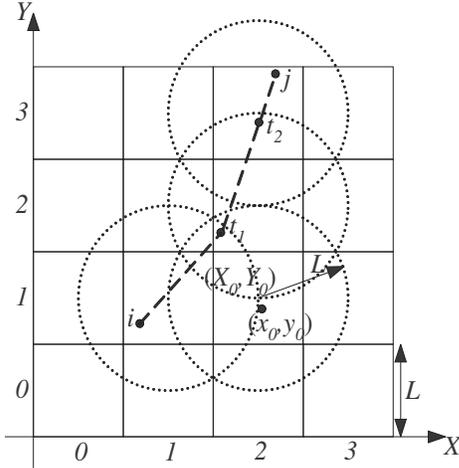


Figure 2: The LKE scheme: A virtual grid, with each grid size of L , is computed based on location information. A service node is selected around the center of each grid who will take care of key establishment for nodes residing within a distance L .

- Node i sends to node j the message encrypted with $K_{i,j}$ along with (x_i, y_i) . After receiving the message, node j computes $K_{ji} = f_{X,Y}^j(k_i) = f_{X,Y}(k_j, k_i)$, where $k_i = Hash(x_i, y_i)$, $k_j = Hash(x_j, y_j)$. Since $f_{X,Y}$ is symmetric, $f_{X,Y}(k_i, k_j) = f_{X,Y}(k_j, k_i)$. Hence, $K_{ij} = K_{ji}$ and node j can decrypt the message.

4.4.2 Path Key Establishment

If two sensors do not share any key space, intermediary nodes must be exploited for path key establishment. For this purpose flooding is often employed in existing key pre-distribution schemes, which is too expensive for large scale sensor networks. While in LKE, the deterministic location-aware procedure makes it very efficient to set up a path key.

Assume node i and node j need to establish a path key for secure communication.

- Node i computes the coverage region of the service node from its home grid, which is centered at the grid center with a radius of L , and selects a location (x_{t1}, y_{t1}) that is *closest* to node j .
- Node i computes K_{it1} , the shared key with (x_{t1}, y_{t1}) , then use K_{it1} to encrypt K_{ij} , a random number selected as the path key.
- Node i sends K_{ij} to (x_{t1}, y_{t1}) securely. In case that no sensor exists at (x_{t1}, y_{t1}) , the underlying geographic routing protocol ensures that a nearby sensor (x'_{t1}, y'_{t1}) will receive the message. Then (x'_{t1}, y'_{t1}) requests node i to resend the message encrypted with $K_{it'1}$, the shared key between node i and (x'_{t1}, y'_{t1}) .
- If (x_{t1}, y_{t1}) (or (x'_{t1}, y'_{t1})) resides in the same key space with node j , K_{ij} can be transmitted securely to the destination and the path key establishment is finished. Otherwise, the same procedure is employed until K_{ij} reaches node j successfully.

An example is shown in Fig. 2, in which two intermediary nodes t_1 and t_2 are found for path key establishment between i and j .

Note that LKE determines the valid region, not a specific sensor, to search for intermediary nodes. Therefore two communicating sensors can employ different intermediaries in different sessions. This results in better resilience against traffic analysis attacks compared with group-based schemes [14, 19] that rely on node id for shared key identification. Furthermore, the above pairwise key establishment procedure can be secured with the introduction of nonces to avoid replay attacks.

5. SECURITY AND PERFORMANCE ANALYSIS

We evaluate the security of LKE in terms of resilience against node capture attack and connectivity of the induced key-sharing graph, and measure the performance in terms of storage and communication overheads. Since service nodes are designed as sacrifices, which do not obviously affect the lifetime of a large-scale sensor network, we care about the performance of worker sensors only.

For most of the following experiments, we consider a sensor network deployed over a field of 100 by 100. The number of sensors in each scenario varies from 300 to 900, with each node capable of a fixed transmission range of 10.

5.1 Resilience

Note that a polynomial key space has the property of *t-collision resistance*, which means that as long as no more than t sensors are captured within the same key space, the shared keys between any pair of non-captured sensors remains secure. Meanwhile, each key space in LKE is expected to serve t nodes in a uniformly distributed network. Thereafter, the resilience of LKE degrades slowly with the increase of the number of captured nodes when sensors are uniformly distributed. In the case of non-uniform deployment, those key spaces with denser sensors may show fragile resilience. Adaptive grid partition based on network density may conquer this problem, which will be explored as a future research.

In the simulation, we consider a smart attack where an adversary compromises all nodes within a disk of radius R_a , and measure the resilience with the following metric:

- **Resilience:** Given an attack radius, the *resilience* of LKE against node capture attacks is defined to be the fraction of the *directly-compromised* links among all the *compromised* links. A directly-compromised link is incident to at least one captured node. When more than t sensors owning the keying information from the same key space are captured, all the links secured by the key space are compromised. Note that the metric *resilience* is in the range $(0, 1]$, where a value closer to 1 represents a better resilience.

As illustrated in Fig. 3, the resilience of LKE degrades slowly with the increase of the attack radius in a uniformly distributed network. An adversary can learn almost nothing about the uncompromised sensors from those being captured. The resilience is close to 1, indicating a “perfect” resilience. Note that the minor fluctuation of the result is attributed to the fact that the topology is not perfectly uniform in our simulation, and therefore it is possible for some key space to serve more than t worker sensors.

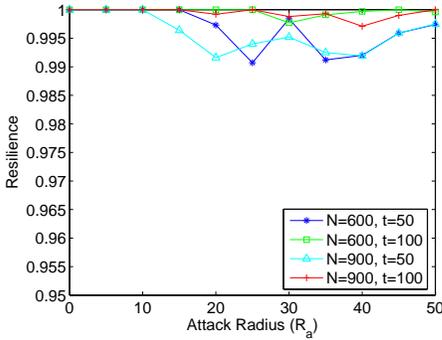


Figure 3: LKE: Resilience against node capture attacks

5.2 Storage Overhead

In LKE, each sensor resides in a grid computed from its physical location. The grid size L , derived from the network density information, also determines the region to be served by a service sensor. Thus, network density can be employed to estimate the average number of polynomial shares stored in each worker node.

Assume N sensors are uniformly distributed in a deployment area A . The grid size L is set such that $\pi L^2 = t \times A/N$. The number of worker nodes to be covered by a key space can be estimated as $\pi L^2 \times N/A = t$. Hence, the average number of polynomial shares stored in each worker sensor, denoted by τ , can be estimated as:

$$\tau \approx \frac{t \times (\lceil \sqrt{A}/L \rceil)^2}{N} \approx \frac{t \times A/L^2}{N} = \pi \quad (3)$$

Each polynomial share is computed from a bivariate t -degree polynomial over a finite field F_s , and takes up $(t+1) \log s$ memory spaces, where s is a prime number that is larger than 2^{len} , len is the length of a cryptographic key. Hence, the memory spaces for keying information stored in a worker node is:

$$m \approx \tau \times (t+1) \log s \approx \pi \times (t+1) \log s, \quad (4)$$

which equals the amount of space for storing $\pi \times (t+1)$ keys. Fig. 4 plots our analytical and simulation results for τ , the number of polynomial shares stored in a worker sensor.

It is interesting to observe that π upper bounds the storage overhead τ in our simulation. The reason is because the variation of t in Eq. 3 is much smaller compared to that of the term $(\lceil \sqrt{A}/L \rceil)^2$, which is amplified to a larger degree due to the ceiling function.

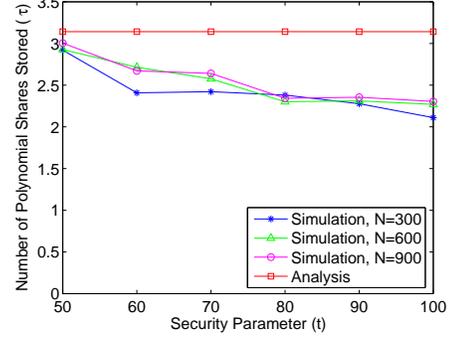


Figure 4: LKE: Polynomial share storage in a worker sensor (analysis vs. simulation)

5.3 Connectivity

The effectiveness of a key distribution scheme is also dependent on the connectivity of the final key-sharing graph $G_K(V, E)$, where V is the set of sensors in the network, and E is the set of edges incident to two neighboring nodes that can securely communicate (i.e. share at least one common key space). LKE is expected to provide high connectivity of the key-sharing graph since the coverage areas of key spaces in proximity overlap. Each sensor can compute a pairwise key directly with nodes in the same grid, or establish a path key with nodes in a neighboring grid with the help of an intermediary node residing in the overlapping region of the two associated key spaces.

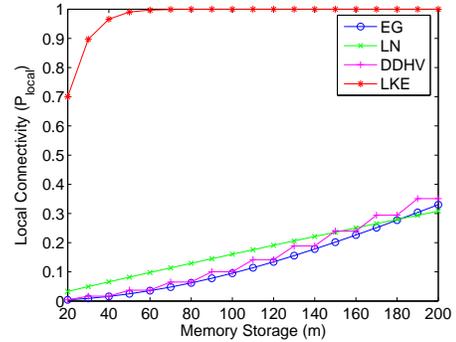


Figure 5: EG, DDHV, LN vs. LKE: P_{local} Comparisons

A nice property of LKE is that its key-sharing graph has high connectivity but its storage overhead is low in a worker node. Fig. 5 plots the relationship between the probability of establishing a shared key between two neighboring nodes, denoted as P_{local} , and the number of keys stored in each node. We measure the P_{local} of LKE and compare it with that of the basic random key predistribution scheme (EG) [9], the random polynomial-based key space predistribution scheme (LN) [12], and the random symmetric matrix based key space predistribution scheme (DDHV) [7]. The settings in EG and DDHV are the same as those in [8]. In EG, the key pool is of size 100,000. In DDHV, the security parameter λ is set to 19, and there are 241 key spaces in total. For LN and LKE, both are considered in a network of size 600, with each node storing 3 polynomial shares (we select 3 since it is a typical value for LKE regardless of network conditions, as illustrated in Section 5.2). Fig. 5

shows that LKE can reach a high connectivity at the expense of a small amount of storage overhead.

5.4 Communication Overhead

Since LKE is an in-situ key establishment scheme, messages are transmitted for keying information distribution as well as pairwise key establishment. Compared to the existent key predistribution schemes, the additional traffic may appear to be a deathful weakness for LKE. However, polynomial shares are only transmitted within a local region restricted by a radius L , and are helpful to realize deterministic key distribution based on network connectivity. The amount of unnecessary keying information carried by a worker sensor is greatly reduced, and it is much more efficient to establish a path key between two communicating sensors multi-hop away.

In LKE, each sensor can easily derive the overlapping region covered by both the service node from its home grid and that of an adjacent grid, then choose an arbitrary node from the region to establish a path key. Compared with the existent key pre-distribution schemes that require flooding to search for an intermediary sensor for path key establishment [5, 7–9, 14, 19], LKE produces much less amount of traffic, contributing greatly to network lifetime elongation.

6. CONCLUSION

The design of LKE targets large-scale sensor networks with resource constraints. In this scheme, sensors determine their roles and configure themselves automatically based on a pure localized algorithm. Only service sensors are in charge of key space generation and distribution, which help to conserve resources in worker sensors. A distinctive feature of LKE is that location information is employed for sensor role differentiation and for polynomial share determination and distribution. LKE is a deterministic procedure that greatly reduce the communication overhead in path key establishment. Simulation study indicates that LKE has a good performance in terms of key-sharing probability between neighboring sensors, memory consumption, and resilience against node capture attacks.

7. ACKNOWLEDGMENT

The research of Dr. Xiuzhen Cheng is supported by the NSF CAREER Award No. CNS-0347674.

8. REFERENCES

- [1] R. Anderson, H. Chan, A. Perrig. Key Infection: Smart Trust for Smart Dust. In *IEEE ICNP'04 Proceedings*, pp. 206-215.
- [2] C. Blundo, A. D. Santis, A. Herzberg, S. Kuttan, U. Vaccaro, and M. Yung. Perfectly-secure key distribution for dynamic conferences. In *CRYPTO'92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pp. 471-486, 1992.
- [3] S. A. Camtepe and B. Yener. Key Distribution Mechanisms for Wireless Sensor Networks: a Survey. *RPI Technical Report TR-05-07*, March 23, 2005.
- [4] D. W. Carman, P. S. Kruss, and B. J. Matt. Constraints and approaches for distributed sensor network security. *NAI Labs Technical Report #00-010*, Sept. 2000.
- [5] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *S&P'03: Proceedings of the 24th IEEE Symposium on Security and Privacy*, pp. 197-215, 2003.
- [6] X. Cheng, A. Thaeler, G. Xue, and D. Chen. TPS: a time-based positioning scheme for outdoor sensor networks. In *IEEE INFOCOM 2004 Proceedings*, Vol. 4, pp.2685-2696, HongKong, March 7-11, 2004.
- [7] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *ACM CCS'03*, New York, NY, pp. 42-51, 2003.
- [8] W. Du, J. Deng, Y. S. Han, S. Chen, and P. Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In *IEEE Transactions on Dependable and Secure Computing*, 2005.
- [9] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *ACM CCS'02 Proceedings*, pp. 41-47.
- [10] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *ACM MobiCom 2000 Proceedings*, pp. 243-254.
- [11] L. Lazos and R. Poovendran. SeRLoc: Robust localization for wireless sensor networks. In *ACM Transactions on Sensor Networks (TOSN)*, Vol. 1, No. 1, pp. 73-100, August 2005.
- [12] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In *ACM CCS'03 Proceedings*, pp. 52-61.
- [13] D. Liu and P. Ning. Location-based pairwise key establishments for static sensor networks. In *SASN'03 Proceedings*, pp. 72-82.
- [14] D. Liu, P. Ning, and W. Du. Group-based Key PreDistribution in Wireless Sensor Networks. In *ACM WiSe'05 Proceedings*.
- [15] F. Liu, X. Cheng, D. Hua, and D. Chen. TPSS: A Time-based Positioning Scheme for Sensor Networks with Short Range Beacons. In *ICCNMC'05 Proceedings*, LNCS 3619, pp. 33-42, August 2005.
- [16] A. J. Menezes, P. C. V. Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Inc., 2001.
- [17] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. *Technical Report MIT/LCS/TR-212*, Laboratory for Computer Science, MIT, 1979.
- [18] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *ACM Wise 2003 Proceedings*, pp. 1-10, San Diego, CA, 2003.
- [19] L. Zhou, J. Ni, and C. V. Ravishanker. Efficient Key Establishment for Group-based Wireless Sensor Deployments. In *ACM WiSe'05 Proceedings*, pp. 1-10.