LKE: A Self-Configuring Scheme for Location-Aware Key Establishment in Wireless Sensor Networks

Fang Liu, *Student Member, IEEE,* and Xiuzhen Cheng, *Member, IEEE*

***Abstract*— Symmetric key agreement is significant to security provisioning in sensor networks with resource limitations. A number of pairwise key pre-distribution protocols have been proposed, but their scalability is often constrained by the conflict between the desired probability of sharing keys between neighboring nodes and the resilience against node capture attacks under a given budget for storing keying information within each sensor. In this paper, we propose LKE, a self-configuring in-situ key establishment scheme targeting large-scale sensor networks. LKE employs location information for a deterministic key space generation and keying information distribution. For uniformly distributed networks, LKE exhibits strong resilience against node capture attacks and achieves a high key-sharing probability (close to 1) at the expense of a small amount of memory overhead. An improvement over LKE, termed as iLKE, is also proposed. iLKE is topology-adaptive, and therefore works well for both uniform and non-uniform network models. We conduct both theoretic analysis and simulation study to evaluate the performances of LKE and iLKE.**

***Index Terms*— Security, in-situ key establishment, wireless sensor networks.**

I. Introduction

SECURE communication is critical for many sensor network applications. Nevertheless, the constrained capabilities of smart sensors (battery supply, CPU, memory, etc.) and the harsh deployment environment of a sensor network (wireless, ad hoc, etc.) make this problem very challenging. Researchers in this field expect 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. For example, a middle-ranged processor such as the Motorola MC68328 “DragonBall” consumes 42mJ (840mJ) for RSA encryption (digital signature) and 0.104mJ for AES when the key is of size 1024 bits [5]. Based on this observation, researchers have proposed a number of pairwise key establishment protocols recently [6], [9], [10], [12], [14], [15]. However, these methods may not scale well or may require strict deployment knowledge for better scalability. Further, most of them are probabilistic-based, requiring a non-negligible amount of security information to be preloaded into the memory of a sensor, thus wasting storage space since many information may never be used during the lifetime of the sensor.

As claimed by [11], the probabilistic-based key predistribution schemes explore the tradeoff of security and memory consumption, since the amount of preloaded information is constrained by the memory budget within each sensor. A stronger security results in a higher memory consumption. This seems unavoidable in all predistribution schemes [17], [19], due to the randomness since no sensor network topology information is available before deployment. In this paper, we propose LKE and iLKE, two truly in-situ schemes for bootstrapping keys in sensor networks that remove the randomness and achieve good security with a small amount of memory consumption.

LKE is designed for location-aware key establishment in large-scale sensor networks. In LKE, a fraction of sensors are self-elected to become *service sensors*, which are in charge of key space generation and keying information distribution. The majority of the sensors, namely *worker sensors*, get keying information from service sensors in the neighborhood. Two worker sensors can compute a common key as long as they obtain keying information from the same service sensor. Keying information distribution and pairwise key derivation are both based on location information through a deterministic procedure, which is very efficient for path key establishment between two sensors sharing no common key space. LKE places no special requirement on worker sensors. We further propose iLKE, an improved scheme that is topology-adaptive, working well for both uniform and non-uniform network distribution. Simulation study indicates that both schemes achieve a high level of key-sharing probability and strong resilience with a tradeoff of a small amount of storage overhead per node in a uniformly distributed network, while iLKE outperforms LKE with a bit more memory overhead in a non-uniformly distributed network.

Compared with the existing schemes proposed for shared key establishment for sensor networks, LKE and iLKE have the following characteristics or advantages:

- LKE (iLKE) divides sensors into a grid structure and disseminates keying information accordingly. It is purely localized, having high scalability in network size, achieving high key-sharing probability in the induced key-

Manuscript received July 8, 2006; revised October 30, 2006; accepted December 6, 2006. The associate editor coordinating the review of this paper and approving it for publication was M. Guizani. This research was supported by the NSF under CAREER Award CNS-0347674 and grant CCF-0627322.

F. Liu is with the Department of Computer Science, University of Texas - Pan American. (e-mail: fliu@cs.panam.edu).

X. Cheng is with the Department of Computer Science, The George Washington University, Washington, DC 20052 USA (e-mail: cheng@gwu.edu).

Digital Object Identifier 10.1109/TWC.2008.060455.

sharing graph with low storage overhead in each sensor, and showing strong resilience in against node capture attacks.

- LKE (iLKE) employs location information for a deterministic key space generation and keying information distribution, which makes path key establishment much more efficient compared with the existing key predistribution schemes.
- iLKE is a topology-adaptive procedure that achieves high connectivity and strong resilience in both uniform and non-uniform networks. This feature is particularly attractive since it is difficult to obtain *a priori* knowledge of post-deployment configuration for many sensor network applications.

This paper is organized as follows. Related work and network model are sketched in Section II and Section III, respectively. We propose LKE, the location-aware key establishment scheme in Section IV, and iLKE, the enhanced topology-adaptive scheme in Section V. We evaluate both schemes in Section VI, and conclude our paper in Section VII.

II. Related Work

In this section, we summarize a number of most related works. For a more comprehensive literature survey, we refer the readers to [4].

The basic *random keys scheme* is proposed by Eschenauer and Gligor in [12], 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 [6] which requires $q > 1$ number of common keys for two nodes to establish a shared key. In [6], [12], a path key can be established for two sensors that demand secure communication but have no common keys in their key rings. A drawback of this mechanism is that the path key is exposed to all intermediary nodes. To overcome this problem, Zhu *et al.* [23] propose to break the secret (the shared key) into multiple shares and each share is delivered to the destination along a different logical path. The secret is restored at the destination when a number of shares are received.

None of the above mentioned random keys schemes guarantees that a key is shared by only one pair of sensors. Therefore compromising one sensor may threaten links that are incident to uncompromised nodes. This problem has been tackled by Chan *et al.* in [6] and [7], which propose the *random pairwise keys scheme*. In this scheme, every node receives a number of unique keys, with each shared with another node that is randomly selected before deployment. This pairing is done based either on node ids [6], or on virtual grid locations [7]. Similar to the random keys schemes, the random pairwise keys schemes do not scale well to large-scale sensor networks. Neither do they have good key-sharing probability due to the high randomness in preloading keying information before deployment.

To improve security, two *random key spaces schemes* [9], [14] have been proposed. These two schemes are very similar in nature, except that the key spaces are defined differently. [9] is based on symmetric matrix [2], while [14] is based on symmetric polynomial [3]. In [9], a key space is constructed based on Blom's method [2], and a shared key between two nodes corresponds to one entry of a symmetric matrix. In [14], a key space is defined by a symmetric bivariate λ-degree polynomial [3], and the shared key of two sensors is the value obtained by plugging the two ids into a polynomial. 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 [9], [14], LKE achieves much better performance in key-sharing probability and storage overhead, as indicated in our simulation study.

To achieve a better scalability, the *group-based schemes* [10], [16], [22] are proposed which effectively reduce the randomness inherent to the key predistribution schemes mentioned above. Du *et al.* [10] 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 [16], [22], sensors are grouped based on IDs, and nodes within the same deployment group or the same cross group are preloaded with pairwise keys before deployment. The schemes in [16], [22] release the strong topology assumption adopted by [10], but still require flooding for path key establishment. Compared with [10], [16], [22], our scheme can support more efficient path key establishment. Further, LKE reduces the randomness to a much stronger degree, since it is also an in-situ key establishment scheme like SBK [17] and iPAK [19].

III. Preliminaries, Assumptions, and Models

A. Preliminaries

Our scheme works fine with both key space models introduced by [3], [10]. We employ the polynomial key space model [3] as an example. A polynomial key space utilizes a bivariate λ-degree polynomial $f(u,v) = f(v,u) = \sum_{i,j=0}^{\lambda} a_{ij}u^j v^j$ over a finite field F_s, where s is a prime that is large enough to accommodate a cryptographic key. By plugging in a value z_i (e.g. z_i can be the id, location, etc.) associated with sensor i, we obtain the *polynomial share* allocated to i. In this paper, we choose $z_i = Hash(x_i, y_i)$, where (x_i, y_i) is the physical position of sensor i. Therefore sensor i receives the polynomial share $f(z_i, v)$ from the key space $f(u,v)$. Thus two sensors i and j knowing each other's position information can compute the shared key $f(z_i, z_j)$ if they have polynomial shares from the same key space $f(u,v)$.

B. Network Model

We consider a large-scale stationary sensor network deployed in outdoor environments. Sensors are able to position themselves through any of the techniques proposed in literature (e.g. [8], [18]), and they communicate with each other following a geographic routing protocol (e.g. [13]).

TABLE I

PRELOADED SYSTEM PARAMETERS.

λ	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 sensor competition

We assume homogeneous sensors densely deployed in a given region. Sensors are preloaded with several system parameters, and differentiate themselves as either worker sensors or service sensors 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.

C. Adversary Model

We assume sensors are not tamper-resistant. The compromise of a sensor releases all its security information to the adversary. Similar to [1], we assume that an adversary can only passively monitor a small proportion of the communications at any time, and no global adversary that can monitor all of the messages at all times exists. This assumption is realistic since a sensor deployed in a security-critical environment must be designed to survive at least a short interval when captured by an adversary as argued by [1] and [24]; otherwise, the whole network can be easily taken over by the opponent.

We further assume that a cryptographically secure key k_0 is preloaded to all sensors such that all communications in the key establishment procedure of LKE can be protected by a popular symmetric cryptosystem such as AES or Triple-DES. A sensor is allowed to participate in the key establishment procedure of LKE if and only if it knows k_0, therefore k_0 is adopted mainly to protect against false sensor injection attacks[1]. Note that k_0 is strong enough such that it is almost impossible for an adversary to recover it before the key establishment procedure is complete, and the release of k_0 after the key establishment procedure does not negatively affect the security of LKE since all sensitive information involved in the key establishment procedure is protected via a different technique in LKE.

IV. 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 sensor in the *node self-configuration* phase. A worker sensor first obtains a polynomial share from a service sensor through a secure channel in the *polynomial share distribution* phase, then computes shared keys with the other nodes during *pairwise key establishment* phase.

A. Pre-distribution

Three pre-configured system parameters, λ, L and δ, as listed in Table I, are preloaded to each sensor. The security parameter λ, indicating the collusion resistance degree of the key space carried by a service sensor, is determined by the memory budget of a sensor (to be explained in Section VI-C). The grid size L determines the coverage area of a service sensor, which is expected to cover λ worker sensors. The competition area with a radius of δ specifies the service sensor election region within which nodes can communicate with each other directly. Therefore L and δ are initialized according to the following criteria:

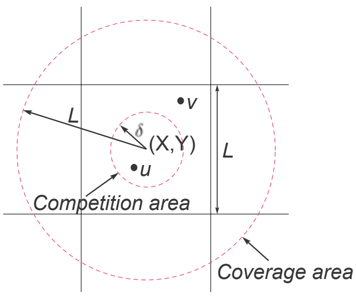


Fig. 1. LKE: A virtual grid, with each grid size of L, is computed based on location information. Sensor u is selected from the competition area and will take care of key establishment for nodes residing in the coverage area. u is the home service sensor of v.

$$\pi L^2 = \lambda \times A/N, \tag{1}$$
$$\delta = R/\sqrt{5}, \tag{2}$$

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.

B. Node Self-Configuration

Right after deployment, a sensor positions itself and determines its role according to its location information. Only sensors from limited regions are eligible for being service sensors, and a localized competition procedure will be conducted for the final role determination. The details of the node self-configuration procedure are elaborated in Algorithm 1.

Based on location information, a virtual grid structure is first computed. As illustrated in Fig. 1, each grid contains a *competition area*, the disk region within a radius of δ from the grid center. At most one service sensor will be selected from the competition area. Each service sensor will establish a key space and serve those worker sensors residing in the *coverage area*, the disk region centered at the grid center with a radius of L. A service sensor is the *home service sensor* of a worker sensor if they reside in the same grid.

The virtual grid structure can be computed as follows. Let (x, y) be the location of sensor S. The *home grid*, where S resides in, can be labelled with the grid center (X, Y), where

$$X = (\lfloor x/L \rfloor + 1/2) \times L, \tag{3}$$
$$Y = (\lfloor y/L \rfloor + 1/2) \times L. \tag{4}$$

Next S computes its distance to (X, Y). If the distance is less than δ, S is eligible to compete for being a service sensor.

An eligible sensor first waits a random delay. If it receives no competition message from others, it announces its decision

[1] An adversary deploys either service sensors or worker sensors.

to be a service sensor. Otherwise, the sensor self-configures as a worker sensor. Note that all the eligible sensors 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 that 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 λ-degree polynomial $f_{X,Y}(x,y) = \sum_{i,j=0}^{\lambda} a_{ij}x^i y^j$ 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 $p \equiv q \equiv 3$ mod 4, where p and q constitute Rabin's public cryptosystem [21] with a public key of $n = p \times q$ and a private key of (p, q). All ineligible sensors and those that have failed in the competition configure themselves as worker sensors.

Algorithm 1 Node Self-configuration

```
function ρ=NodeConfig($\lambda, \delta, L$)    ▷ $\rho$: the selected role
    $(x, y) \leftarrow self\text{-}positioning$    ▷ Localization
    $X \leftarrow (\lfloor x/L \rfloor + 1/2) \times L$    ▷ Get home grid id
    $Y \leftarrow (\lfloor y/L \rfloor + 1/2) \times L$
    $D \leftarrow \sqrt{(x-X)^2 + (y-Y)^2}$    ▷ Get distance
    if $D \leq \delta$ then    ▷ Eligible for the competition
        $TTL \leftarrow rand$    ▷ Wait a random time
        $elapse(TTL)$
        if $not\ recv(competition\_msg)$ then
            $broadcast(competition\_msg)$    ▷ Succeed
            $\rho \leftarrow ServiceNode$
            $\{s, p, q\} \leftarrow getPrimes$
            $f_{X,Y} \leftarrow getPolynomial(\lambda, s)$    ▷ Get a
                symmetric $\lambda$-degree bivariate polynomial
            $PSD(f_{X,Y}, x, y, L)$    ▷ Algorithm 2
        else    ▷ Fail the competition
            $\rho \leftarrow WorkerNode$
        end if
    else    ▷ Not eligible
        $\rho \leftarrow WorkerNode$
    end if
    return $\rho$
end function
```

C. 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 sensors in the neighborhood, which is composed of the following three steps:

1) Key Space Advertisement: A service sensor S announces its existence through beacon broadcasting when its key space is ready. The beacon message includes: i) the key space id (X, Y), which is also the id of S's home grid, ii) (x_0, y_0), the location of sensor S, and iii) the public key n, where $n = p \times q$, p and q are the two primes generated in the previous step. This message will be forwarded to all sensors within S's coverage area.

2) Secure Channel Establishment: Any worker sensor 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, Y) is actually smaller than δ. For each valid announcement, a computationally asymmetric channel based on Rabin's cryptosystem [21] is established for polynomial share distribution. After obtaining the public key n, a worker sensor picks up a random key K_s and computes $E_n(K_s||R) = (K_s||R)^2\ mod\ n$, where R is a predefined bit pattern for ambiguity resolution in Rabin's decryption. $E_n(K_s||R)$, along with the location information, is transmitted to the corresponding service sensor. After Rabin's decryption, the service sensor obtains $D_{p,q}(E_n(K_s||R)) = K_s||R$, where K_s will be utilized to protect the polynomial share transmission from the service sensor to the work sensor.

Note that Rabin's cryptosystem [21] is a computationally asymmetric public cryptosystem. Its encryption operation involves only one squaring, which is extremely fast (several hundreds of times faster than that of RSA). But its decryption time is comparable to that of RSA. The security of Rabin's scheme is based on the factorization of large numbers, thus it is comparable to that of RSA too. Therefore by adopting Rabin's scheme, LKE shifts a large amount of computation overhead to service sensors, which intend to be sacrifices, to conserve the resource in worker sensors, and meanwhile achieves strong protection to the keying information dissemination (see Subsection IV-C.3).

3) Polynomial Share Acquisition: After agreeing on a shared key K_s with a worker sensor i at (x_i, y_i), the service sensor first computes a location-aware polynomial share $f^i_{X,Y} = f_{X,Y}(z_i, y)$ where $z_i = Hash(x_i, y_i)$, then transmits $f^i_{X,Y}$ to i. This message is protected by K_s eatablished in the previous step. The behavior of a service sensor for polynomial share distribution is summarized by Algorithm 2. Any two worker sensors receiving polynomial shares from the same service sensor can compute a shared key directly for secure data exchange in the future.

Algorithm 2 Polynomial Share Distribution

```
procedure PSD($f_{X,Y}, x_0, y_0, L$)▷ $(x_0, y_0)$ is the position
        of the service sensor
    $n \leftarrow p \times q$
    Broadcast $(x_0, y_0, n)$ within $L$-distance    ▷ Key space
        advertisement
    if $recv(request, x_i, y_i, E_n(K_s))$ then    ▷ Distribute
        polynomial share to node $(x_i, y_i)$
        $K_s \leftarrow D_{p,q}(E_n(K_s))$    ▷ Decrypt $K_s$
        $k_i \leftarrow Hash(x_i, y_i)$
        $f^i_{X,Y}(y) \leftarrow f_{X,Y}(k_i, y)$    ▷ Compute polynomial
            share for $(x_i, y_i)$
        $send(x_0, y_0, E_{K_s}(f^i_{X,Y}(y)))$
    end if
    $elapse(TTL)$
end procedure
```

After disseminating the polynomial shares to all worker sensors in the coverage area, *the service sensor erases all stored key space information for security enhancement.*

D. 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 or not based on their location information. Such a deterministic procedure results in an efficient pairwise key establishment procedure.

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 $K_{ij} = f^i_{X,Y}(k_j) = f_{X,Y}(k_i, k_j)$, where $k_i = Hash(x_i, y_i)$, $k_j = Hash(x_j, y_j)$.
- Node i sends to node j the message encrypted with K_{ij} along with (x_i, y_i). After receiving the message, node j computes $K_{ji} = f^j_{X,Y}(k_i) = f_{X,Y}(k_j, k_i)$, 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.

2) Path Key Establishment: If two sensors do not share any key space but desire a pairwise key, intermediary nodes can 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 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 area of its home grid (X_i, Y_i), and selects a location (x_t, y_t) within the disk region that is *closest* to node j.
- Node i computes K_{it}, the shared key with location (x_t, y_t), then use K_{it} to encrypt K_{ij}, a random number selected as the path key.
- Node i sends K_{ij} to (x_t, y_t) securely. In case that no sensor exists at (x_t, y_t), the underlying geographic routing protocol ensures that a nearby sensor at (x'_t, y'_t) receives the message. This sensor requests node i to resend the message encrypted with $K_{it'}$, the shared key between node i and the sensor at (x'_t, y'_t).
- The sensor at (x_t, y_t)(or (x'_t, y'_t)) gets the key K_{ij} and continues the procedure 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 key pre-distribution schemes [16], [22], which 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.

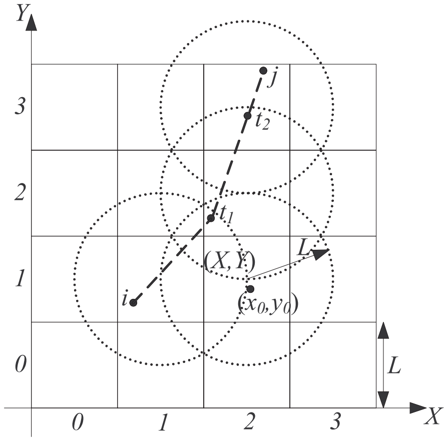


Fig. 2. Path Key Establishment in LKE: $t_1(t_2)$ is selected as the intermediary node since it is closest to the destination j within the coverage area of i's(t_1's) home service sensor.

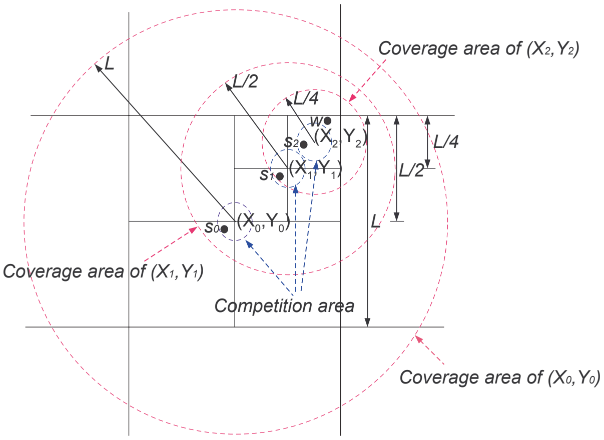


Fig. 3. In iLKE, the size of a grid and the associated coverage area is determined by the *level* property, but the competition area is of a fixed size. A worker sensor may have at most one home service sensor at each level. Sensor $s_0/s_1/s_2$ is the home service sensor of sensor w at $level = 0/1/2$, respectively.

V. iLKE: The Improved Topology-adaptive LKE Scheme

The security of LKE relies on the underlying key space model. Note that the exemplified polynomial key space has the property of *λ-collusion resistance*, which means that as long as no more than λ sensors covered by the same key space are compromised, the pairwise key between any two non-compromised sensors remains secure. Meanwhile, the grid size L is set such that each key space in LKE is expected to serve λ nodes in a uniformly distributed network (see Eq. (1)). Thereafter, the resilience of LKE degrades gradually with the increase of compromised nodes when sensors are uniformly distributed. However, in the case of non-uniform deployment, those key spaces serving more sensors may show fragile resilience. To conquer this problem, we propose iLKE, an improved scheme that employ adaptive grid partition based on network density.

In iLKE, each virtual grid is further associated with a new property *level*. The grids in LKE have $level = 0$. Grids at

$level > 0$ are generated only when necessary. As illustrated in Fig. 3, a grid at level i is of size $L/2^i$, whose coverage area is of a radius $L/2^i$. Each grid contains a competition area, a disk region of a radius δ. At most one service sensor is selected from the competition area, which will serve at most λ worker sensors in the vicinity. Note that for a grid at level $i > 0$, the service sensor competition cannot happen until being *triggered* by some sensor that cannot be served by its $(i-1)$*th-level home service sensor* due to the limitation on the capacity of a key space.

For a sensor S at (x, y), the *ith-level home grid* (X_i, Y_i) can be derived based on the following criteria:

$$X_i = (\lfloor x \times 2^i/L \rfloor + 1/2) \times L/2^i, \tag{5}$$

$$Y_i = (\lfloor y \times 2^i/L \rfloor + 1/2) \times L/2^i. \tag{6}$$

Similarly to LKE, iLKE is consisted of four phases: *pre-distribution, node self-configuration, polynomial share distribution,* and *pairwise key establishment*. The difference lies in the second and the third phases:

- *Node Self-Configuration*: If a node S fails in the competition for being a service sensor at $level = 0$, it is still possible to win at $level > 0$ if it is eligible. A node self-configures to be a work sensor if it fails in all competitions.
- *Polynomial Share Distribution*: To obtain a better resilience, iLKE requires that each service sensor disseminates at most λ polynomial shares to its coverage area. Thus in a dense region, sensor u may receive the key space existence notification from its ith-level home service sensor S_i but be declined its request for a polynomial share. In this case, u initiates the $(i+1)$th-level home service sensor competition at its $(i+1)$th-level home grid by broadcasting a *trigger message*. This broadcasting is controlled by a random delay for collision avoidance, and is squelched when hearing another trigger message. As illustrated in Fig. 4, the process terminates when no service sensor could be elected because of a void competition area, or all the nodes in the coverage area are assigned polynomial shares.

Remark: In iLKE, the adaptive grid partition and service sensor generation will terminate at level i under two conditions: (i) each sensor in the coverage area of an ith-level service sensor is assigned a polynomial share, (ii) no $(i+1)$th-level service sensor can be selected due to a void competition area. For a given network, only these two cases exist when i increases to a certain value I_0, since the coverage area is small enough (with a radius $L/2^{I_0}$) and contains less than λ nodes (case (i)), or no nodes exist in the δ-region (case (ii)). Therefore, the adaptive grid partition will converge.

VI. Security and Performance Analysis

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

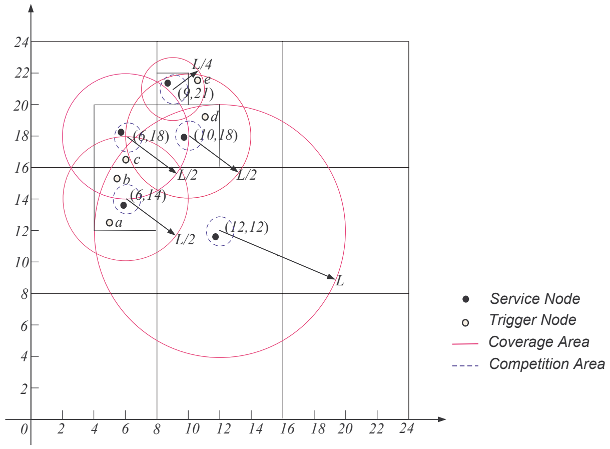


Fig. 4. In iLKE, the ith-level $(i > 0)$ service sensor competition is triggered by nodes receiving key space broadcasting message but not being served by the $(i-1)$th-level home service sensor. Sensors a, b, c, d work as trigger nodes since they cannot be served by their home grid $(12, 12)$ at level 0. Three $1st$-level service sensors (a, b belong to the same $1st$-level home grid) are elected at $(6, 14), (6, 18), (10, 18)$, respectively. No further grid partition happen at $(6, 14), (6, 18)$, since nodes can be fully served in this region. But a $2nd$-level service sensor is elected at $(9, 21)$ since node e receives key space advertisement from $(10, 18)$ but cannot get any keying information.

A. Simulation Settings

For most of the following experiments, we consider a sensor network deployed over a field of 1000 by 1000. The number of sensors, denoted by N, in each scenario is 2000 or 3000, with each node capable of a fixed transmission range of 40.

We consider the following two network models in our simulation study:

- Uniform deployment: N sensors are uniformly distributed throughout the whole deployment region.
- Group-based gaussian deployment: N sensors are divided into 3×3 groups. Nodes within a group follow a 2-dimensional gaussian distribution, with the pdf:

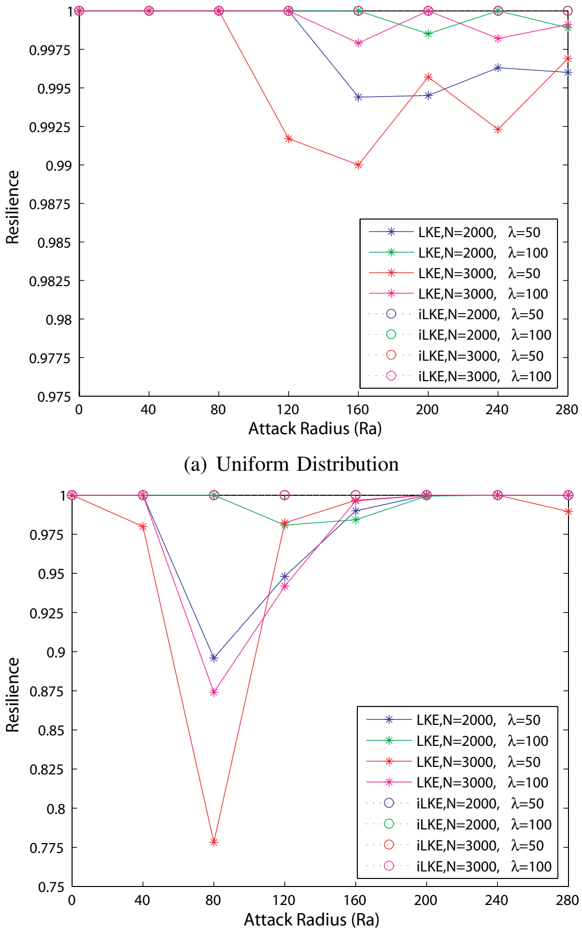
$$f(x, y) = \frac{1}{2\pi\sigma^2} e^{-[(x-x_c)^2+(y-y_c)^2]/2\pi\sigma^2}, \tag{7}$$

where $\mu = (x_c, y_c)$ is the center of the group deployment region, $\sigma = .25\sqrt{G}$, G is the deployment area for a group. Each group contains the same amount of sensors, and covers the same deployment area.

In security analysis, we consider a smart attack model where an adversary attacks nodes within a limited region. Since LKE and iLKE regulate that the keying information be distributed within a pre-defined region, a smart attack would be more destructive than an oblivious attack where an adversary randomly captures nodes throughout the whole deployment field. For simplicity, we assume a circular attack region that is of a radius R_a and centered at (x_{center}, y_{center}), the center of the deployment area.

B. Resilience

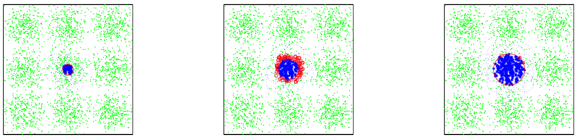
As analyzed in Section V, the resilience of LKE degrades gradually with the increase of the number of compromised nodes in a uniformly distributed sensor network, because of the λ-*collusion resistance* of the underlying key space and the objective to cover λ worker sensors in each key space as



(a) Uniform Distribution

(b) Gaussian Distribution

Fig. 5. LKE, iLKE: Resilience against node capture attacks.



(a) $R_a = 40$ (b) $R_a = 80$ (c) $R_a = 120$

Fig. 6. The resilience of LKE fluctuates with the increase of the attach radius. The exemplified network is in a two-dimensional Gaussian distribution. The */o/. nodes represent captured/indirectly-compromised/unaffected nodes, respectively. The dashed circles denote the associated key spaces for nodes within the attack region (denoted by a solid circle).

specified in Eq. (1). iLKE retains the same resilience in a non-uniform network with adaptive grid partition. By triggering more service sensors to be generated if necessary, iLKE retains a perfect resilience since each key space accommodates at most λ worker sensors.

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 R_a, the resilience of LKE against node capture attacks is defined to be the fraction of the compromised links incident to at least one compromised sensor among all the compromised links. 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. 5(a), the resilience of LKE degrades gradually 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. As for iLKE, the resilience remains constant as 1. No matter how large the attack region is, no secret information will be released about the communication links among uncaptured nodes. Both LKE and iLKE can achieve a "perfect" resilience (close or equal to 1) in uniformly distributed networks, while LKE exhibits small fluctuation when compared to iLKE. Such fluctuation is attributed to the topology that is not perfectly uniform in the simulation, and therefore it is possible for some key space to serve more than λ worker sensors. It is even more obvious in a network where sensors follow group-based Gaussian distribution. As illustrated in Fig. 5(b), iLKE still holds a constant resilience as 1, while the performance of LKE is fluctuant with the increase of the attack radius R_a. Note that LKE sets key spaces in a grid structure, thus the increase of R_a will not necessarily increase the *fraction of additional compromised links*[2] if the adversary has already captured more than λ nodes in a key space. An example is shown in Fig. 6, the increase of R_a from 80 to 120 does not increase the fraction of additional compromised links, since all the newly captured nodes can also be compromised when $R_a = 80$.

C. 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 sensor.

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

$$\tau \approx \frac{\lambda \times (\lceil \sqrt{A}/L \rceil)^2}{N} \approx \frac{\lambda \times A/L^2}{N} = \pi \qquad (8)$$

Each polynomial share is computed from a bivariate λ-degree polynomial over a finite field F_s, and takes up $(\lambda + 1)\log s$ 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 sensor is:

$$m \approx \tau \times (\lambda + 1)\log s \approx \pi \times (\lambda + 1)\log s, \qquad (9)$$

which equals the amount of space for storing $\pi \times (\lambda + 1)$ keys.

Fig. 7 plots our analytical and simulation results for τ, the number of polynomial shares stored in a worker sensor. The two schemes exhibit similar storage overhead in a uniform network distribution, while iLKE burdens worker sensors

[2] We refer *additional compromised links* as those links whose associated communicating parties are compromised but not directly captured.

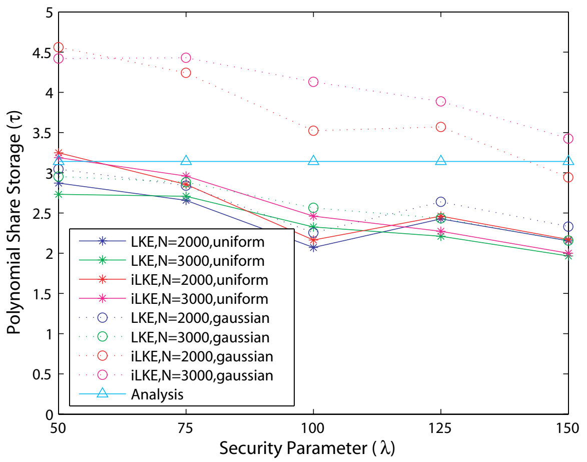


Fig. 7. LKE, iLKE: Keying information storage in a worker sensor.

with a larger storage overhead in a group-based Gaussian distributed network since iLKE requires at most λ worker sensors to be served in a key space and thus incurs more service sensors to be generated. However, as analyzed in Subsection VI-B, the increase of storage overhead contributes to a much stronger resilience to node capture attacks.

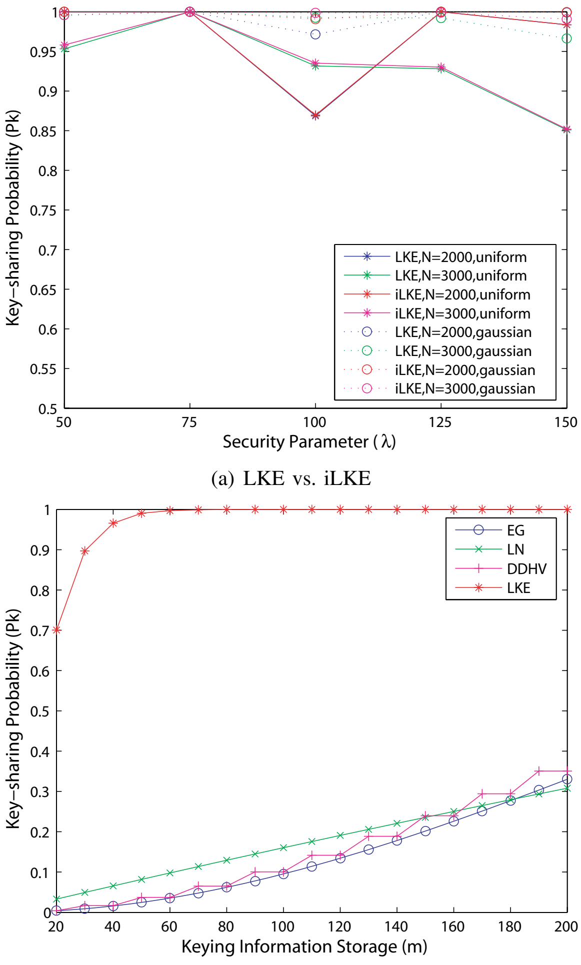
D. Key-sharing Probability

The effectiveness of a key distribution scheme is also dependent on the key-sharing probability, denoted as p_k, which is the probability that two neighboring worker sensors are able to establish a shared key. Both LKE and iLKE are expected to provide high key-sharing probability 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. Fig. 8(a) plots the simulation results for LKE and iLKE in both network models.

A nice property of LKE is that it ensures a high key-sharing probability but its storage overhead is low in a worker sensor. Fig. 8(b) plots the relationship between the probability of establishing a shared key between two neighboring nodes and the number of keys stored in each node. We measure the p_k of LKE and compare it with that of the basic random key predistribution scheme (EG) [12], the random polynomial-based key space predistribution scheme (LN) [14], and the random symmetric matrix based key space predistribution scheme (DDHV) [9]. The settings in EG and DDHV are the same as those in [10]. 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 VI-C). Fig. 8(b) shows that LKE can reach a high key-sharing probability at the expense of a small amount of storage overhead.

E. Communication Overhead

Since LKE and iLKE are two in-situ key establishment schemes, 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 the two schemes. However, polynomial shares are only transmitted within a local region restricted by a radius L ($L/2^i$ for a grid at level i in iLKE), and are helpful to realize a deterministic keying information 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.



(a) LKE vs. iLKE

(b) EG, DDHV, LN vs. LKE

Fig. 8. LKE, iLKE: key-sharing probability.

In LKE (iLKE), each sensor can easily derive the overlapping region covered by both the service sensor 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 [6], [9], [10], [12], [16], [22], LKE produces much less amount of traffic, contributing greatly to network lifetime elongation.

F. Computation Complexity

For LKE and iLKE, the computational expenses on a worker sensor come from two stages: (i) to establish a secure channel to the associated service sensor and decrypt the received polynomial shares during *polynomial share distribution*, (ii)

to calculate shared keys with other worker sensors in *pairwise key establishment*.

To obtain polynomial shares securely from a service sensor, a worker sensor needs to encrypt a secret key K_s with Rabin's algorithm (one squaring only) and decrypt the received polynomial share with a symmetric cryptography algorithm (AES, DES, etc.). Note that the asymmetric Rabin's cryptosystem, with a comparable security with RSA, shifts a large amount of the computational overhead to service sensors and thus lengthens the lifetime of worker sensors.

To compute a pairwise key with sensor j at (x_j, y_j), sensor i instantiates the λ-degree polynomial share with $k_j = Hash(x_j, y_j)$, which requires λ modular multiplications and λ modular additions. The computation process has been tailored for sensor networks by [14] which greatly reduces the computation overhead by transforming onto a smaller finite field.

VII. Conclusion

The design of LKE targets large-scale sensor networks with severely constrained resources. 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 keying information distribution, which help to conserve resources in worker sensors. A distinctive feature of LKE is that location information is employed for node role differentiation and for polynomial share determination and distribution. LKE is a deterministic procedure that greatly reduce the communication overhead in path key establishment. In a uniformly distributed network, LKE exhibits strong resilience in against node capture attacks and high key-sharing probability (close to 1) at the expense of a small storage overhead in worker sensors. We also propose an enhanced scheme, iLKE, which is topology-adaptive, working well for both uniformly and non-uniformly distributed networks. Simulation study indicates that both LKE and iLKE have a good performance in terms of key-sharing probability, keying information storage overhead, and resilience against node capture attacks.

References

[1] R. Anderson, H. Chan, A. Perrig, “Key infection: smart trust for smart dust,” in *Proc. IEEE ICNP’04*, pp. 206–215.

[2] R. Blom, “An optimal class of symmetric key generation systems,” in *Proc. EUROCRYPT 84 Workshop on Advances in Cryptology: Theory and Application of Cryptographic Techniques*, 1985, pp. 335–338.

[3] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, “Perfectly-secure key distribution for dynamic conferences,” in *Proc. CRYPTO’92: the 12th Annual International Cryptology Conference on Advances in Cryptology*, 1992, pp. 471–486.

[4] S. A. Camtepe and B. Yener, “Key distribution mechanisms for wireless sensor networks: a survey,” RPI Technical Report TR-05-07, 2005.

[5] D. W. Carman, P. S. Kruss, and B. J. Matt, “Constraints and approaches for distributed sensor network security,” *NAI Labs Technical Report #00-010*, 2000.

[6] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks,” in *Proc. S&P’03: the 24th IEEE Symposium on Security and Privacy*, 2003, pp. 197–215.

[7] H. Chan and A. Perrig, “PIKE: peer intermediaries for key establishment in sensor networks,” in *Proc. IEEE Infocom 2005*, March 2005.

[8] X. Cheng, A. Thaeler, G. Xue, and D. Chen, “TPS: a time-based positioning scheme for outdoor sensor networks,” in *Proc. IEEE INFOCOM 2004*, March 2004.

[9] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, “A pairwise key pre-distribution scheme for wireless sensor networks,” in *Proc. ACM CCS’03*, pp. 42–51.

[10] W. Du, J. Deng, Y. S. Han, S. Chen, and P. Varshney, “A key management scheme for wireless sensor networks using deployment knowledge,” *IEEE Trans. Dependable and Secure Computing*, vol. 3, no. 2, pp. 62–77, 2006.

[11] W. Du, R. Wang, and P. Ning, “An efficient scheme for authenticating public keys in sensor networks,” in *Proc. ACM MobiHoc’05*, pp. 58–67.

[12] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in *Proc. ACM CCS’02*, pp. 41–47.

[13] B. Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks,” in *Proc. ACM MobiCom 2000*, pp. 243–254.

[14] D. Liu and P. Ning, “Establishing pairwise keys in distributed sensor networks,” in *Proc. ACM CCS’03*, pp. 52–61.

[15] D. Liu and P. Ning, “Location-based pairwise key establishments for static sensor networks,” in *Proc. ACM SASN’03*, pp. 72–82.

[16] D. Liu, P. Ning, and W. Du, “Group-based key predistribution in wireless sensor networks,” in *Proc. ACM WiSe’05*.

[17] F. Liu and X. Cheng, “A self-configured key establishment scheme for large-scale sensor networks,” in *Proc. Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems* (MASS 2006), pp. 447–456, Oct. 2006.

[18] F. Liu, X. Cheng, D. Hua, and D. Chen, “TPSS: a time-based positioning scheme for sensor networks with short range beacons,” in *Proc. ICCNMC’05*, LNCS 3619, pp. 33–42, 2005.

[19] L. Ma, X. Cheng, F. Liu, J. Rivera, and F. An, “iPAK: an in-situ pairwise key bootstrapping scheme for wireless sensor networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 8, pp. 1174–1184, Aug. 2007.

[20] A. J. Menezes, P. C. V. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, Inc., 2001.

[21] 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.

[22] L. Zhou, J. Ni, and C. V. Ravishankar, “Efficient key establishment for group-based wireless sensor deployments,” in *Proc. ACM WiSe’05*, pp. 1–10.

[23] S. Zhu, S. Xu, S. Setia, S. Jajodia, “Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach,” in *Proc. IEEE ICNP’03*, Nov. 2003.

[24] S. Zhu, S. Setia, and S. Jajodia, “Leap: efficient security mechanisms for large-scale distributed sensor networks,” in *Proc. ACM CCS’03*, pp. 62–72.

Fang Liu is an Assistant Professor in the Department of Computer Science at the University of Texas - Pan American. She received her D.Sc. degree in Computer Science from The George Washington University in August 2007. Her current research interests include wireless and mobile computing, wireless security, algorithm design and analysis.

Xiuzhen (Susan) Cheng is an Assistant Professor in the Department of Computer Science at the George Washington University. She received her MS and PhD degrees in Computer Science from the University of Minnesota - Twin Cities in 2000 and 2002, respectively. Her current research interests include Wireless and Mobile Computing, Sensor Networks, Wireless Security, Statistical Pattern Recognition, Approximation Algorithm Design and Analysis, and Computational Medicine. Dr. Cheng has served in the editorial board of technical journals such as Ad Hoc Networks Journal. She was the Program Co-Chair of the first International Conference on Wireless Algorithms, Systems, and Applications (WASA06). Dr. Cheng worked as a program director in the National Science Foundation for six months in 2006. She received the NSF CAREER Award in 2004.