Key Predistribution Schemes for Sensor Networks Using Finite Plane Geometry

Hisashi MOHRI†, Member, Ritsuko MATSUMOTO†, Nonmember, and Yuichi KAJI†(a), Member

SUMMARY This study is to investigate new schemes for distributing cryptographic keys in sensor networks. Sharing a key is the very first step to realize secure communication over an untrusted network infrastructure, but commonly used cryptographic techniques cannot be employed for sensor networks due to the restriction of computational resources of sensor nodes. A practical solution to this issue is to predistribute cryptographic keys in sensor nodes before they are deployed. A focal point in this solution is the choice of keys that are assigned to a sensor node. Eschenauer et al. considered to choose keys randomly, and Chan et al. also followed the random choice approach. We consider in this paper a new approach in which keys are assigned according to a basic algebraic geometry. The performance of the proposed scheme is investigated analytically.

key words: sensor networks, key agreement method, finite geometry, management of cryptographic keys, ubiquitous system

1. Introduction

This paper discusses schemes for two neighbor sensor nodes (nodes that can directly communicate with each other) to agree a cryptographic key. Sharing a cryptographic key between two parties is the very first step to realize secure communication over an untrusted communication channel. Without sharing a key, it will be difficult to realize secure, authenticated and trustful communications because we will not be able to use modern cryptographic techniques. Key agreement has been one of significant subjects in cryptology, and nowadays several solutions are known widely for the “usual” computer networks. Those known key agreement schemes commonly make use of techniques of public-key cryptography or similar methods, and require terminals to perform rather large and complicated computation. Consequently, it is widely considered that those techniques for key agreement are not suitable for sensor networks that consists of resource-restricted sensor nodes.

A primitive but practical solution for the key agreement in sensor networks is to predistribute keys to sensor nodes before they are deployed. The random key distribution is an implementation of key predistribution proposed by Eschenauer [6]. In the random key distribution, a key manager determines a set of keys beforehand. The set of keys are sometimes called a key pool. Keys that are embedded in a sensor node is randomly chosen by the key manager before the node is deployed to a field. Two sensor nodes can agree a cryptographic key if and only if they happen to have keys in common. Several extensions of this simple scheme have been investigated, which we will review in the next section.

The primal objective of this study is to construct a better implementation of the key predistribution scheme than the random key distribution scheme of [6]. We consider three quantitative measures to evaluate key predistribution schemes; the node memory size, the connectivity and the key-survivability. The node memory size (or memory size for simplicity) is the number of keys that a node needs to remember, the connectivity is the probability that randomly chosen two sensor nodes have one or more keys in common, and the key-survivability is the probability that the key which has been agreed by two nodes stays secure even if an intruder mounts the node capture attack, in which an intruder captures sensor nodes deployed in a field, and retrieves keys embedded in the nodes. There are obvious tradeoff relations between the three quantitative measures. For example, if we try to increase the connectivity without changing the memory size, then the key-survivability decreases accordingly in general. A key predistribution scheme needs to provide good tradeoff points with respect to the three measures, and it is preferred that the tradeoff points are controllable. The random key distribution scheme is, however, not good with respect to these issues, because the scheme is so simple that we have little way to control its performance.

In this study, we consider to actively control keys to be embedded in sensor nodes so that both the connectivity and the key-survivability are larger than the random key distribution scheme. For this sake, we consider to use simple geometric properties of lines and points. We associate each node with a line over a two-dimensional finite plane, and manage keys so that two nodes can agree a key if and only if the associated lines intersect with each other. Two randomly chosen lines intersect with each other unless they are parallel, thus two nodes succeed in key agreement with high probability. We propose a scheme that is directly based on this idea, and call the scheme a basic scheme in this manuscript. The basic scheme can realize high connectivity with small memory size, but it is difficult to flexibly control the tradeoff points in general. To overcome this problem, we investigate two extensions of the basic scheme. The extended schemes have more flexibility than the basic scheme, though they are not as advantageous as the basic scheme compared to the random key distribution scheme. We evaluate the basic and extended schemes analytically
and compare the result with that of the random key distribution scheme.

2. Related Work

A naive scheme for key agreement in sensor networks is to use a global key. The key manager determines a unique “global key”, and distribute the global key to all the sensor nodes. A node can agree a key with any other node in the system and therefore the connectivity is 1. On the other hand, the key-survivability of the scheme is quite poor because an intruder can obtain all the secret of the system by capturing just one node. Another straightforward scheme is to assign different keys for different pairs of nodes, and provide a sensor node with the keys that the node commits to. This scheme gives high key-survivability, but the memory size increases as the network size (the number of nodes in the network) increases. Usually a sensor network consists of many sensor nodes that have small memory size, and therefore this scheme is not suitable for sensor networks.

In the random key distribution scheme [6], as briefly introduced in the previous section, the connectivity and the key-survivability are controlled by two parameters; the size of the key pool $|K|$ and the node memory size $m$ (the number of keys stored in a node). We can show easily that the connectivity and the key-survivability (under the situation that $c$ nodes are captured) of the random key scheme are given as

$$p_{\text{con}} = 1 - \prod_{i=0}^{m-1} \frac{|K| - m - i}{|K| - i},$$

$$p_{\text{surv}} = \left(1 - \frac{m}{|K|}\right)^c,$$

respectively. The random key distribution scheme is sometimes used as a “building block” of more sophisticated key predistribution schemes. For example, Chan et al. consider to improve the trade-off points of the random key distribution scheme by making use of multiple keys that are shared by two nodes [3], but they basically use exactly the same random key distribution scheme in its key distribution phase. The scheme proposed by Du et al. [4] also uses the random key distribution scheme partly.

There are several studies that take different approaches from the random key scheme. One major approach among them is to make use of the deployment location of sensor nodes. In the random key approach, it is implicitly assumed that we cannot predict where a sensor node is deployed. In some applications, however, this is not the case. We may be able to statistically predict the deployment location of each sensor node. Such statistical information is helpful to realize secure and efficient key agreement, as investigated in [1], [8], [10], [13]. The relation between the location-based and the location-free approaches will be discussed in Sect. 5.2.

There also exist studies in which much emphasis is devoted to establishing path-keys. Assume that two nodes, say $a$ and $b$, fail to agree a cryptographic key. In this case, we can ask other nodes to help $a$ and $b$ setting up a key: Determine a sequence of nodes $a = n_0, n_1, \ldots, n_{r-1}, n_r = b$ so that $n_i$ and $n_{i+1}$ with $0 \leq i \leq r - 1$ can agree a key, and ask the intermediate nodes $n_i$ to relay secret information that enables key agreement between $a$ and $b$. The key established between $a$ and $b$ in this way is called a path-key. Liu proposes in [9] a scheme which makes use of bivariant polynomials. The scheme has relatively low connectivity, but high probability that randomly chosen two nodes succeed to agree a path-key. Du et al. consider similar approach [5], but they use the Blom’s scheme [2] instead of bivariant polynomials. To the authors’ understanding, a path-key cannot be an alternative to the link-key that is directly agreed between two nodes. Computing a path-key requires sensor nodes to pay large computational and communication overheads, and its security will be seriously damaged if there are selfish or dishonest nodes in the network. Consequently it is quite misleading to compare the connectivity with respect to link-keys and the connectivity with respect to path-keys.

Other well-known studies on the key management in sensor networks include SPINS by Perrig et al. [12]. SPINS is a collection of protocols for key management between sensor nodes and a base station, and does not intend for the key agreement between two sensor nodes. LEAP is another widely known scheme for key management in sensor networks [14]. LEAP is a general mechanism that allows a group of nodes to agree a key, but the security of LEAP strongly depends on an uncommon assumption; a node is equipped with a precise timer, a (global) master key, and a special mechanism that erases the master key at a certain timing. The node makes use of the master key to share cryptographic keys with other nodes, and then erases the master key to avoid troubles in case the node is captured. If the key-erasing mechanism does not work as expected and an intruder happen to capture a node in which the master key remains not-erased, then the entire system might be paralyzed completely.

Light mechanisms for key agreement have been studied for long years. For example, Matsumoto and Imai have proposed the concept of the Key Predistribution Scheme (KPS) in late 80s [11], mainly intended for applications in smartcards. The KPS itself is a general concept, and [11] gives an implementation of KPS that utilizes matrix computation. The concept of KPS is also effective for sensor networks, but concrete implementations of KPS that is suitable for sensor networks is still an open problem. Gong and Wheeler have proposed a key management scheme based on algebraic geometry [7]. The idea to use lines and point is similar to our work, though, the used techniques and the obtained results of [7] are quite different from ours. Some characteristics of the scheme in [7] are problematic in sensor network applications. For example, the scheme in [7] cannot support a large network that consists of nodes with quite small memory size.
3. Proposed Scheme

3.1 Preliminary

Let \( p \) be a prime number, \( Z_p = \{0, \ldots, p-1\} \) and \( Z_p^2 = \{(x, y) | x, y \in Z_p\} \). We call \( Z_p^2 \) a plane and elements in \( Z_p^2 \) points. For \( a \) and \( b \) in \( Z_p \), a line is a collection of points defined as \( l(a, b) = \{(x, y) | x \in Z_p, y = ax + b \ (mod \ p)\} \). We can easily show the following lemma.

**Lemma 3.1:** Consider two lines \( l(a_1, b_1) \) and \( l(a_2, b_2) \):

1. If \( a_1 = a_2 \) and \( b_1 \neq b_2 \), then \( l(a_1, b_1) \cap l(a_2, b_2) = \emptyset \) (we say that the two lines are parallel).
2. If \( a_1 \neq a_2 \) then \( l(a_1, b_1) \cap l(a_2, b_2) \) contains exactly one point in \( Z_p^2 \) (we say that the two lines intersect, and the point is an intersection point).

**Proof:** Let \( a_1 = a_2 \) and \( b_1 \neq b_2 \), and assume without loss of generality that \( b_1 \leq b_2 \). Suppose that there is \((x, y)\) such that \( (x, y) \in l(a_1, b_1) \cap l(a_2, b_2) \), so \( a_1 x + b_1 \equiv a_2 x + b_2 \ (mod \ p) \). This implies that \( b_2 - b_1 \equiv 0 \) mod \( p \). Because \( (b_2 - b_1) \in Z_p \), \( b_2 - b_1 \equiv 0 \) if and only if \( b_2 = b_1 \). This contradicts our assumption, and (1) is proved. Next, let \( a_1 \neq a_2 \), and assume without loss of generality that \( a_1 < a_2 \). For \( x \in Z_p \), define as \( d_x = (ax + b) - (a_1 x + b_1) \) (mod \( p \)). To prove (2), we show that \( d_{xa}, \ldots, d_{xp-1} \) is the permutation of \( 0, \ldots, p-1 \). It is clear that \( d_x \in Z_p \) for any \( x \in Z_p \), and it suffices to show that \( d_{xa}, \ldots, d_{xp-1} \) are all different. Suppose for contradiction that \( d_x = d_{x'} \) for \( x < x' \), then

\[
(a_2 x + b_2) - (a_1 x + b_1) \\
\equiv (a_2 x' + b_2) - (a_1 x' + b_1) \mod p.
\]

Thus, \( (a_2 - a_1)(x' - x) \equiv 0 \) mod \( p \). However, the product of \((a_2 - a_1)\) and \((x' - x)\) cannot be divisible by \( p \), because \( p \) is prime, \( (a_2 - a_1) \in Z_p \) and \((x' - x) \in Z_p \). Therefore, no value appears twice in \( d_{xa}, \ldots, d_{xp-1} \). \( \square \)

**Example 3.2:** Consider a plane \( Z_p^2 \). Figure 1 shows three lines \( l(1,0), l(2,3) \) and \( l(1,2) \) over \( Z_5^2 \). The lines consist of the following points:

- \( l(1,0) = \{(0,0), (1,1), (2,2), (3,3), (4,4)\} \)
- \( l(2,3) = \{(0,3), (1,0), (2,2), (3,4), (4,1)\} \)
- \( l(1,2) = \{(0,2), (1,3), (2,4), (3,0), (4,1)\} \)

The line \( l(1,0) \) intersects with the line \( l(2,3) \) at the point \((2,2)\), and the line \( l(2,3) \) intersects with the line \( l(1,2) \) at the point \((4,1)\). The line \( l(1,0) \) is parallel to the line \( l(1,2) \). \( \square \)

Let \( L \) be the class of all lines, then \( L \) contains \( p^2 \) lines in total. Let \( l \) be a line in \( L \), then there are \( p \) lines (including \( l \) itself) that are parallel to \( l \), and all the other \( p^2 - p = p(p-1) \) lines intersect with \( l \). Thus if we choose another line from \( L \) randomly, then the line intersect with \( l \) with probability \( p(p-1)/p^2 = (p-1)/p \), which approaches to 1 as \( p \) increases.

3.2 Basic Scheme

To make the later discussion clear, we describe a key predistribution scheme by a four-tuple \((K, N, m, k)\), where \( K \), the key pool, is a collection of keys, \( N \) is the set of nodes, \( m \) is the number of keys that are assigned to a sensor node, and \( k \) is a key assignment function mapping \( N \) to a subset of \( K \) with cardinality \( m \). For example, in the random key distribution scheme [6], \( k \) is a random function that associates a node with a randomly chosen subset of \( K \).

Now we define a new key predistribution scheme and call it the **basic scheme** in this manuscript. First, choose a prime number \( n \), and associate each point in \( Z_p^2 \) with a randomly chosen cryptographic key. The key that is assigned to a point \( \pi \) denoted as \( k(\pi) \). The key pool \( K \) is defined as \( K = \{k(\pi) | \pi \in Z_p^2 \} \). For a node \( n \in N \), let \( l(n) \) be a line that is chosen randomly from \( L \). The key assignment function \( k \) is then defined as

\[
k(n) = \{k(\pi) | \pi \in l(n)\}.
\]

Intuitively, a node is associated with a randomly chosen line in \( L \) and the node has keys that are on the line. Because a line contains \( p \) points, a node is assigned with \( p \) keys. The obtained key predistribution scheme is thus \((K, L, m, k)\).

It is easily understood that two sensor nodes \( n_1 \) and \( n_2 \) share a key in common if and only if lines \( l(n_1) \) and \( l(n_2) \) intersect. Thus the connectivity of the basic scheme is

\[
p_{con} = (p-1)/p.
\]

(1)

To discuss the key-survivability, assume that \( l(n_1) \) and \( l(n_2) \) intersect at a point \( \pi \), and \( n_1 \) and \( n_2 \) share the key \( k(\pi) \). The number of lines that pass through \( \pi \) is \( p \), and therefore if we choose a line from \( L \) randomly and uniformly, then the chosen line passes through \( \pi \) with probability \( p/p^2 = 1/p \). This probability coincides with the probability that a sensor node which is captured by an intruder happen to have \( k(\pi) \). Consider that an intruder have captured \( c \) sensor nodes. The key \( k(\pi) \) stays secure (not know to the intruder) if and only if all the \( c \) nodes do not include \( k(\pi) \). This happens with
probability \((1 - 1/p)^c\). Thus the key-survivability of the basic scheme under \(c\) captured nodes is
\[ p_{sur} = (1 - 1/p)^c. \]

3.3 Evaluation of the Basic Scheme

The performance of the basic scheme is evaluated in this section. The basic scheme is intended to replace for the random key distribution scheme in \([6]\), and to be used as a “building block” of other advanced schemes. Thereby we compare the basic scheme and the random key distribution scheme, and do not consider other advanced schemes in this section. As we noted in the introduction, there are tradeoff relations between the memory size, the connectivity and the key-survivability. To make the discussion clear, we meanwhile set the memory size to a constant number, and observe the relation between the connectivity and the key-survivability.

According to some literatures \([3]\), \([6]\), we assume that a node can have about 100 keys. In the basic scheme, the memory size equals to the prime number \(p\), and hence choosing \(p = 101\) seems reasonable for this setting. This choice of \(p\) makes the connectivity \(p_{con} = 0.99\). In the random key distribution scheme, the connectivity is controlled by the size of the key pool \(|\mathcal{K}|\) and the memory size \(m\). To make \(p_{con} = 0.99\) and \(m = 101\), the key pool must contain 2311 or less keys. Under the choice of these parameters, we compare the survivabilities of the two schemes under a node capture attack. The numerical result is depicted in Fig. 2. The \(x\)-axis of the graph is the number of compromised nodes, and the \(y\)-axis is the key-survivability. As the number of compromised nodes increases, the key-survivability decreases in general. We can see from the figure that the key-survivability of the random key distribution scheme drops rapidly as the number of compromised nodes increase, whereas that of the basic scheme decreases slowly. Compare, for example, the key-survivability of the two schemes when 50 nodes are compromised. The figure shows that only 10% of keys survive in the random key distribution scheme, while about 60% of keys survive in the basic scheme. This means that the basic scheme offers more robustness against node capture attacks than the random key distribution scheme, with the same connectivity and the same number of keys in each node.

To compare the two schemes from another direction, consider how many keys are needed in the random key distribution scheme to achieve the same performance as the basic scheme. Assume that we would like to realize a key agreement scheme with the connectivity \(p_{con} = 0.99\) and the key-survivability \(p_{sur} \geq 0.9\) for up to 10 compromised nodes. In the basic scheme, choosing \(p = 101\) achieves the required performance as illustrated in Fig. 2. In this case, a sensor node needs to have 101 keys. To achieve the same performance by using the random key distribution scheme, we need to choose so that \(|\mathcal{K}| = 41304\) and \(m = 454\). The random key distribution scheme requires a sensor node to have four times more keys than the basic scheme. Other comparisons for other choice of \(p_{sur}\) and \(c\) are presented in Fig. 3. We can see that the basic scheme requires smaller number of keys than the random key scheme. The number of keys has strong relationship to the memory size of a sensor node, and thus to the manufacturing cost and the energy efficiency of a node. We could see clear advantage of the basic scheme against the random key distribution scheme.

We would like to remark that the basic scheme also has an advantage in the communication overhead for the key agreement. In the random key distribution scheme, two nodes need to exchange what keys they do have. This will be done by exchanging the indices of the keys, but the communication cost for this operation should be proportional to the number of keys in the nodes. In the basic scheme, the keys are assigned in a structured and systematic manner, which helps reducing the communication overhead. Remember that a line over a plane is defined by just two parameters, \(a\) and \(b\). Once two node, \(n\) and \(n'\), exchange their parameters, then the nodes can compute the intersection point of \(l(n)\) and \(l(n')\) by themselves. The communication for exchanging the parameters can be done with constant amounts of clear-text.
messages. This is a significant advantage in sensor networks in which energy consumed for computation and communication is so expensive.

4. Extended Schemes

We have seen in the previous section that the basic scheme is advantageous to the random key distribution scheme. On the other hand, the authors recognize that the basic scheme does not have sufficient flexibility to control the connectivity and the key-survivability. The scheme has just one parameter, the prime number \( p \), and choosing \( p \) determines the memory size (the number of keys assigned to a node), the connectivity and the key-survivability. In general, the prime memory size (the number of keys assigned to a node), the prime number \( p \) does not have sufficient flexibility to control the connectivity and the key-survivability. On the other hand, the authors recognize that the basic scheme can be much smaller than 1 to obtain more key-survivability. Unfortunately, it is difficult in the basic scheme to make \( p \) small value without affecting other criteria. In this section, we investigate extensions of the basic scheme so that we can have more flexibility and controllability.

4.1 Extended Scheme 1

The first type of extensions is to use lines partially. In the basic scheme, a sensor node \( n \) is associated with a line \( l(n) \), and all keys in \( \{k(\pi) | \pi \in l(n)\} \) are assigned to \( n \). Now, instead of assigning all keys on the line, consider to assign a subset of \( \{k(\pi) | \pi \in l(n)\} \) to \( n \). Let \( m \) be an integer with \( m < p \) and define the key assignment function \( k \) as

\[
k(n) = \{k(\pi) | \pi \in l_m(n)\},
\]

where \( l_m(n) \) is a randomly chosen subset of \( l(n) \) with cardinality \( m \). Thus, the number of keys that a node needs to remember is smaller than the prime number \( p \). This may reduce the memory requirement of sensor nodes, or allow us choosing the prime number \( p \) bigger than the memory restriction of sensor nodes.

Two nodes share a key if and only if the lines associated with the nodes intersect, and both of the two nodes have the key at the intersection point of the lines. Thus the connectivity of this extension scheme is

\[
p_{\text{con}}' = \frac{p_m(m/p)^2}{(p-1)m^2/p^2},
\]

where \( p_m \) is defined in (1). Let \( k(\pi) \) be the key shared by the two nodes. A randomly chosen sensor node \( n \) contains \( k(\pi) \) if and only if the line \( l(n) \) passes through \( \pi \) and \( l_m(n) \) happens to contain \( \pi \). The probability that \( n \) have \( k(\pi) \) is therefore \( (p/p^2)(m/p) = m/p^2 \). The key-survivability of the extended scheme for \( c \) captured nodes is

\[
p_{\text{surv}}' = (1 - m/p^2)^c.
\]

4.2 Extended Scheme 2

The above described extension can be regarded as a combined scheme of the basic scheme and the random key distribution scheme, where the random key distribution scheme is used as an “inner scheme”. We can consider another combined scheme in which the random key distribution scheme is used as an “outer scheme”. Let \( S_0 = (K_0, N, m_0, k_0) \) be a random key predistribution scheme. Assume that \( K_0 \) contains \( r \) keys and keys in \( K_0 \) are numbered from 1 to \( r \). The number associated with a key \( k \in K_0 \) is denoted by \( i(k) \). Also define \( r \) basic schemes \( S_i = (K_i, N, m, k_i) \) with \( 1 \leq i \leq r \) where \( K_i \cap K_j = \emptyset \) if \( i \neq j \). Now combine these schemes and define

\[
S_0 \circ \{S_1, \ldots, S_r\} = \bigcup_{i=1}^r K_i, N, m_0m, k'
\]

where

\[
k'(n) = \bigcup_{k \in K_0(n)} k_{i(k)}(n).
\]

In this extension, we prepare \( r \) independent key distribution schemes \( S_1, \ldots, S_r \). The “parent scheme” \( S_0 \) assigns each node with keys in \( K_0 \). If \( k \in K_0 \) is assigned to a node by \( S_0 \), we use a “child scheme” \( S_{i(k)} \) to determine keys for the node \( n \). In other words, \( S_0 \) is used to determine which schemes should be used to assign keys for a node. The scheme \( S_0 \) assigns \( m_0 \) keys to a node, and each \( S_i \) with \( 1 \leq i \leq r \) assigns \( m = p \) keys if it is chosen by \( S_0 \). Therefore a node receives \( m_0m = mp \) keys in total.

In this scheme, two nodes share a key if and only if (i) there is a scheme \( S_i \) with \( 1 \leq i \leq r \) which is commonly used by the two nodes, and (ii) the two nodes agree a key by using the scheme \( S_i \). Remark that \( S_0 \) is a random key distribution scheme and it can assign, to two nodes, multiple keys in common. In this case, the two nodes use multiple child schemes in common. The probability that two nodes use \( i \) child schemes in common is

\[
\left(\begin{array}{c} r-i \\ m_0-i \end{array}\right)\left(\begin{array}{c} 2(m_0-i) \\ m_0-i \end{array}\right) = \left(\begin{array}{c} m_0 \\ i \end{array}\right)\left(\begin{array}{c} r-m_0 \\ m_0 \end{array}\right)^{-1}(r-m_0)^m_0 / (r_m_0).
\]

Two nodes fail to agree a key if all the \( i \) child schemes fail to assign common keys. Therefore the connectivity of this extended scheme is

\[
p_{\text{con}}'' = \sum_{i=1}^{m_0} \left(\begin{array}{c} m_0 \\ i \end{array}\right)\left(\begin{array}{c} r-m_0 \\ m_0 \end{array}\right)^{-1}(1 - p_{\text{con}}')^i.
\]

where \( p_{\text{con}} \) is defined in (1). If we use the extended scheme 1 for \( S_1, \ldots, S_r \) instead of the basic scheme, then the connectivity changes to a value that is obtained by replacing \( p_{\text{con}} \) with \( p_{\text{con}}' \) in (2). To discuss the key-survivability, assume that two nodes are assigned a common scheme \( S_i \), and share a key \( k(\pi) \) of \( S_i \). The probability that a node
which is captured by an intruder happen to include \(k(\pi)\) is 
\[
(m_0/r)(1/p) = m_0rp.
\]
Thus the key-survivability of this scheme under \(c\) captured nodes is
\[
p_{\text{sur}} = (1 - m_0rp)^c.
\]
If we use the extended scheme 1 for \(S_1, \ldots, S_r\), just replace \(m_0rp\) in the above equation with \(m_0m/rp^2\).

In this extended scheme 2, there is a lot of flexibility for parameter choices. Now, consider that a node can have about 200 keys. In this case, we can arbitrarily choose \(m_0\) and \(p\) to accomplish \(m_0p \approx 200\). After we choose \(m_0\) and \(p\), we can change the connectivity and the key-survivability by changing the parameter \(r\). For example, consider three choices \((p, m_0) = (199, 1), (41, 5), \text{ and } (19, 10)\) which all make \(m_0p \approx 200\). For each choice, we choose \(r = 2, 40\) and \(145\) to make the connectivity \(p_{\text{con}}'' \approx 0.5\). Figure 4 shows the evaluation results of key-survivability for the above three choices of parameters. This shows that the key-survivability gets better as \(p\) increases.

### 4.3 Evaluation of the Two Extended Schemes

This section is to compare the extended schemes 1 and 2 with the random key distribution scheme. We set \(m\), the number of keys in a node, to \(m = 200\) in this section, and observe how the key-survivability changes for \(p_{\text{con}} = 0.33, 0.5\) and \(0.9\). Table 1 summarizes parameters which make each scheme achieve the given connectivity. Figures 5–7 illustrate the key-survivability for each \(p_{\text{con}}\). We can see that the extended schemes show better key-survivability than the random key distribution scheme. The advantage of the proposed schemes is especially significant when \(p_{\text{con}}\) is close to 1. This result suggests that the proposed scheme is more favorable when we need high connectivity.

Numerical results show that the extended scheme 1 has small advantage to the extended scheme 2 with respect to

| \(p_{\text{con}}\) | extended 1 \(p\) | extended 2 \((p, r, m)\) | random key \(|K|\) |
|---|---|---|---|
| 0.33 | 347 | (199, 3, 1) | 99,619 |
| 0.50 | 283 | (199, 2, 1) | 58,295 |
| 0.90 | 211 | (37, 21, 6) | 18,008 |

Fig. 4 The performance of the extended scheme 2 for three choices of parameters with \(p_{\text{con}} = 0.5\).

Fig. 5 The key-survivability of the extended schemes with \(p_{\text{con}} = 0.33\).

Fig. 6 The key-survivability of the extended schemes with \(p_{\text{con}} = 0.5\).

Fig. 7 The key-survivability of the extended schemes with \(p_{\text{con}} = 0.9\).
the key-survivability. We note, however, that the extended scheme 2 have some advantages which are not illustrated in the figures. For example, the extended scheme 2 allows sensor nodes to have two or more keys in common. Hence techniques similar to the $q$-composite key [3] is easily available in the extended scheme 2, and the key-survivability might be improved by using such additional techniques.

5. Discussion

5.1 Choice of Parameters

To use the proposed scheme in real applications, we need to determine the values of parameters. Different choice of parameters gives the scheme different characteristics. We would like to find the optimum values of parameters for a given application, but the choice of parameters strongly depends on many aspects that are difficult to quantify.

For example, we consider how the density of sensor nodes affects the choice of parameters. If sensor nodes are deployed sparsely in a field, then we should choose parameters so that the scheme has high connectivity. In a sparse network, a node is expected to have small number of neighbor nodes. If the connectivity is small in this sparse network, then it can happen with considerable probability that a node cannot agree a key with any of its neighbor nodes, and is isolated (in the sense of secure communication) in the network. Such an isolation of a node happens with probability $(1 - p_{\text{con}})^d$, where $d$ is the average number of neighbor nodes. To make this probability small, we need to let $p_{\text{con}}$ take a value close to 1. This is possible by choosing a large prime number for $p$ in the basic scheme, or by choosing $p$ and $m$ close in the extended scheme 1 (parameter choices of the extended scheme 2 is complicated, but we can determine appropriate parameter values by solving an equation).

The discussion goes differently if sensor nodes are deployed densely. On discussing key agreement, there are two points that we need to remark to compare a sparse network and a dense network. The first point is that, in a dense network, an intruder will be able to find and capture sensor nodes easily, and the node capture attack is more serious than the sparse network case. The second point is that the cost to reinforce a key agreement scheme, for example by using path-keys [5],[9] or the multipath key reinforcement [3], is smaller and more acceptable in a dense network than in a sparse network. Hence, instead of increasing the connectivity, we may choose parameters so that the scheme has large key-survivability, and to use the reinforce techniques to mitigate the low connectivity.

5.2 Relation to the Location-Based Approach

A sensor network is constructed by deploying a number of sensor nodes in a designated area. In many cases, it is difficult to predict (or control) the exact deploy location of each node, and we cannot know in advance which nodes come close to a given node. To realize a key-agreement scheme under this situation, there are two different approaches. In a location-free approach, we investigate a scheme that shows reasonable performance for “any” deploy scenario. Even if nodes are deployed unintentionally or randomly, the scheme is required to show reasonable performance. The scheme investigated in this paper follows the location-free approach. In a location-based approach, we assume that nodes are deployed under a certain condition, and devise a scheme which is optimized for the assumed deploy scenario. For example, [4] considers a location-based scheme assuming that nodes are deployed by dropping from a helicopter. We cannot predict exactly where a node is settled, but a certain distribution function will be available in this case. By using the distribution function, we can choose keys to be embedded in a node “non-uniformly”. In other words, we choose keys so that two nodes which will be deployed closely have keys in common. The idea of the location-based approach is further developed in [8] and [1] for example.

The location-based approach has several advantages against the location-free approach. If nodes are deployed as expected, then a location-based scheme shows better connectivity than a location-free scheme in general. It is also said that, in the location-based approach, the affect of a node capture attack is confined in a small local section of the network. The approach is, furthermore, able to support very large scale network.

In spite of these advantages of the location-based approach, it is significant to investigate the location-free approach. The essential problem of the location-based approach is that the approach is not always available. If we know nothing about the node deployment, then we cannot devise a location-based scheme. Consider for example mobile nodes that autonomously and freely move after they are deployed. Obviously location-based schemes are not available to support such mobile nodes, and we need to consider location-free schemes. In other words, location-based schemes cannot replace for all the location-free schemes, and hence studying good location-free scheme has practical significance.

Comparison of the location-free and location-based approaches suggests possible applications of the proposed scheme. The key agreement scheme investigated in this paper is intended to be general, multi-purpose and fundamental technology that is available in wide range of applications. However, unfortunately, the proposed scheme is not the very best for all applications: For example, location-based schemes will be more useful than the proposed scheme if we can predict the deploy locations of sensor nodes. The proposed scheme is superior to other schemes if we cannot predict the deploy locations in advance, and if high connectivity is strongly required. For example, consider mobile sensor nodes that are carried by users or equipped on a vehicle. In such applications, nodes will move extensively large area, and the network can be regarded as very sparse. The connectivity must be sufficiently large in this case, because an encounter with other nodes is “precious” in such a sparse network and we should not miss the opportunity. The
location-based approach is not available in this case since the move of nodes is not predictable. Within the location-free framework, the proposed scheme show clear advantage to the random key distribution scheme when the connectivity is required to be large. It seems that the proposed scheme is the best choice for such an application.

6. Conclusions and Future Work

We proposed new schemes for predistributing cryptographic keys in sensor networks. The proposed scheme is a direct scheme with which key agreement is accomplished between two involved nodes only. The scheme is well suited for realizing high connectivity, which is strongly desired in sensor networks in which sensor nodes are deployed very sparsely. It is also notable that the proposed schemes do not require special assumptions nor equipments such as a timer and key diminish mechanisms.

The proposed scheme can replace for the naive random key scheme, achieving higher security (with respect to the key-survivability) while other system parameters unchanged. The random key distribution scheme has been used as a fundamental component for constructing more sophisticated key management scheme, and therefore the proposed scheme can make significant contribution in the wide ranges of key management techniques for sensor networks.

The authors consider that there are a lot of points for improving and extending the basic scheme. For example, we considered lines and points over a two-dimensional plane. We can extend the geometry to three or more dimensional space.

Acknowledgement

The authors would like to thank anonymous referees for their constructive comments.

References


Hisashi Mohri received a B.E. degree from Nara University of Education in 2003 and received an M.E. and a Ph.D. degrees from Nara Institute of Science and Technology in 2005 and 2008, respectively. In 2008, he joined Information Technology R&D Center, Mitsubishi Electric Corporation. His research interest includes security in ubiquitous computing environments.

Ritsuko Matsumoto received a B.S. degree from Nara Women’s University in 2005 and a M.E. degree from Nara Institute of Science and Technology in 2007. She joined Sharp Corporation in 2007. Her research topic in her master’s course was security techniques for sensor networks.

Yuichi Kaji was born in Osaka, Japan, on December 23, 1968. He received the B.E., M.E., and Ph.D. degrees in information and computer sciences from Osaka University, Osaka, Japan, in 1991, 1992 and 1994, respectively. In 1994, he joined Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. In 2003 and 2004, he visited the University of Hawaii at Manoa as a visiting researcher. His current research interests include the theory of error correcting codes, fundamental techniques for information security, and the theory of automata and rewriting systems. He is a member of IPSJ, SITA and IEEE.