Two Variations of Peer Intermediaries for Key Establishment in Sensor Networks

Jingyuan Rao, Nanchang University School of Software, China

Min Tu, Jiangxi Police College, China

Xuanjin Yang, Southeast University, China

ABSTRACT

Recently, the issue pertinent to sensor network security is a popular topic. Especially, due to the restriction of energy supplied by battery power of the sensors, symmetric cryptography is one of the suitable choices for the secret communication between sensors, while asymmetric cryptography is not appropriate because of its heavy loading on computation should consume a lot of energy. As a result of using symmetric cryptography, there are numerous research papers focusing on designing efficient key management scheme in sensor networks. PIKE designed by Chan and Perrig is a scheme using peer intermediaries and their pre-installed keys to deliver secret message from one sensor to another. However, they did not consider the case that the sensors are scattered in nonuniform way. Moreover, O(logn) is enough for sensor networks to achieve expander topology while PIKE has O($\sqrt{}$n) storage overhead. This article gives generalizations of PIKE to offer more choices for developers under different requirements. The Constant Storage Protocol, abbreviated as CSP, costs constant memory storage and is more suitable for group-based deployment.

KEYWORDS

Key Management, Secret Communication, Sensor Network Security, Symmetric Cryptography, Wireless Sensor Network

1. INTRODUCTION

1.1. Background

Sensor networks may be deployed at any area we want to monitor, including the hostile environment. Under this situation, the communication between any two sensors can be eavesdropped by the adversary to compromise the confidentiality of transmitted message. Thus, secure communication is an essential issue in sensor networks. To provide secure communication, exchanged message between sensor nodes should be encrypted and authenticated. In wireless sensor networks, sensors have limited energy supply that usually cannot be renewed. The lifetime of a sensor network is constrained by the amount of energy that sensors can use to perform operations such as sensing, processing, and transmission. When cryptography is used to ensure the confidentiality of transmitted data, the encryption/decryption procedure needs exquisite designs in order to save energy. Because asymmetric cryptography such as

DOI: 10.4018/IJDCF.2020070101

This article, originally published under IGI Global's copyright on July 1, 2020 will proceed with publication as an Open Access article starting on January 27, 2021 in the gold Open Access journal, International Journal of Digital Crime and Forensics (converted to gold Open Access January 1, 2021), and will be distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

RSA or Elliptic Curve cryptography (ECC) needs extensive computations, they are usually considered to be unsuitable for resource-limited sensor networks (Eschenaur & Gligor, 2002). In this paper, we force on designing key management schemes to achieve secure communication in sensor networks.

1.2. Related Work

We mainly review key management methods in sensor networks without using asymmetric cryptography. The trusted-server scheme, called Kerberos (Steiner et al., 1988), needs support of trusted infrastructure, is unsuitable for sensor networks because the highly frequent forced communication patterns around those servers will become noticeable targets such that the adversary can easily interference with those servers. A more intuitive key-distribution is full pairwise key scheme. In this scheme, each sensor carries $n-1$ secret pairwise keys, where n is the total number of sensors, each of which is known only to a pair of sensors that share this key. Full pairwise key distribution has excellent resilience to node capture attack because capturing of any one sensor does not compromise the security of the sensors that have not been captured. However, such a scheme costs a lot of memory storage and is considered impractical. In contrast with full pairwise key scheme, in master key scheme, all sensors contain only one master secret key, which is used this key to communicate with all other sensors. Although this scheme significantly saves memory storage, its security is very poor. If the adversary captures any one of sensors, then the security of whole network will be broken.

Recently, the most popular key management strategy for secure sensor networks is random key predistribution scheme, which was first proposed by Eschenauer and Gligor (2002). The idea behind random key predistribution exploits the fact that a random graph achieves connectivity with high probability if each node's degree is above a threshold. In random key predistribution, a key pool S is first prepared for key distribution. Then, m distinct keys are chosen from S randomly and assigned to each sensor. These m keys are called the key ring of a sensor. If two sensors have at least one common key in their key rings, then they can communicate with each other using this common secret key. Later on, several researches focus on how to improve the resilience of the original random key predistribution scheme to node capture attack. Chan et al. (2003) suggested that two sensors can directly achieve secure communication if they share q common keys in their key rings, where q is an integer. They also suggested that the keys in a sensor network should be pair wise in order to enhance resistance to capture of sensors by adversary. In Liu and Ning (2003) proposed a class of random key predistribution scheme. In their scheme, several bivariate polynomials having symmetry property, instead of keys, are embedded into sensors. Any two sensors that aim to proceed secret communication must negotiate in advance to get a common bivariate polynomial. Once the common bivariate polynomial is found, the result of substituting the identities of the other sensors and its own identity into the polynomial is the shared key between them. Du et al. (2003) proposed a multi-space random key predistribution scheme, which can adjust the resilience of sensor network to node capture attack. In addition, Wacker et al. (2005) proposed a Recursive Key Establishment Protocol (RKEP) for secure sensor networks. In RKEP, a transmitted message is divided into many parts, which will be sent to the target sensor using different paths. Traynor et al. (2005) the authors examined the efficiency of random key predistribution over heterogeneous sensor networks.

A promising key establishment scheme called Peer Intermediate for Key Establishment (PIKE) was proposed by Chan and Perrig (2005). In PIKE, sensors are arranged to constitute a mesh structure and each sensor is given a 2-D label (i, j) according to its location in the mesh. For key establishment, two sensors will share a unique key if they are either located at the same row or column. Thus, in PIKE, each sensor stores $2(\sqrt{n}-1)$ keys, where n denotes the dimension of a square mesh structure. If two sensors (i_1, j_1) and $(\mathrm{i}_2, \mathrm{j}_2)$ want to communicate with each other, PIKE will perform the following two steps:

- If the two sensors are located at the same row or column, then they share a secret key. Under this circumstance, they can directly use this key for data transmission;
- Otherwise, PIKE will first choose an intermediary ((i_1, j_2) or (i_2, j_1)) and then send data from source sensor, via this intermediary, to target sensor. In other words, at most one intermediary is required for indirect transmission in PIKE.

In addition, there are only two different paths between source and target in the two-dimensional structure of PIKE. This characteristic leads to several other advantages. For example, no additional control message is needed for establishing the shared key between two sensors which aim to interchange messages. However, this is a scheme best suited for uniform deployment because it does not utilize any special property of nonuniform deployment. The disadvantage is that the memory usage of PIKE will increase accordingly with the increase of sensors.

In our paper, a scheme called snake PIKE is additionally added into our comparison. The reason why we also consider this scheme is to reduce the constraint, "indexing", induced by design of PIKE. Since the indexing method used by PIKE is row major, we change the indexing method into snake major to avoid the inherent drawback of row major that a gap happens when there are consecutive elements located in two different rows.

To deal with the weakness of uniform deployment, Du et al. (2004) proposed group-based deployment, where sensors may be deployed on certain locations of a monitored region and form a non-uniform distribution. In group-based deployment, sensors have more chances to communicate with the sensors in the same subgroup and have less chance to communicate with the sensors in different subgroups. Because Du et al. (2004) assume that each sensor has equal probability to diffuse in the target region, the distribution of sensor locations of the same subgroup in group-based deployment is two-dimension Gaussian distribution. In sensor network, long distance peer-to-peer secure communication between sensors is hardly to come up with. For this reason, group-based deployment plays an important role in sensor networks.

1.3. Our Approach

In this paper, we propose a group-based sensor deployment protocol with key establishment satisfying the trad-off between the number of intermediaries and the number of transmissions. We assign a key ring for each sensor node to satisfy the security requirement of sensor networks. Here, the security requirement is secure communication between sensors. We propose a generalized random key predistribution protocol that can set each node's key ring freely in order to balance the number of intermediaries and communication overhead. Because all sensors are labelled from 0 to $n - 1$, with a single index, it is quite simple and convenient for users to label the indies of sensors. The proposed protocol, called Constant Storage Protocol (CSP), costs constant memory storage and is more suitable for group-based deployment. The remainder of this paper is organized as follows. CSP is described in Sec. 2. Experimental results and conclusions are given in Sec. 3 and Sec. 4, respectively.

2. CONSTANT STORAGE PROTOCOL

We exploit the concept of peer intermediary (Chan & Perrig, 2005) and generalize it to develop a new key management protocol under group-based deployment. We will examine several issues including group connectivity, number of different available paths from source to destination, communication overhead, and number of edges in some specified sub-group in CSP.

2.1. Description of CSP

The basic idea of PIKE (Chan & Perrig, 2005) is to embed the keys according to the locations of sensors. It should be noted that the number of keys is the same for each sensor because for a given regular mesh structure, where sensors are deployed, the number of vertical and horizontal neighbors for all sensors are the same. This implies that the number of embedded keys increases with the number of deployed sensors. Thus, this weakness motivates us to design a generalized PIKE regardless of the locations of sensors. In our protocol, users can setup each node's keys freely. Let a graph G_k denote the deployment of sensors, where the vertices of G_k are sensors and an edge (x, y) exists in G_k if and only if sensor x and sensor y share at least one common keys. Since we embed the same number of keys in each sensor, each node's degree is k in G_k. As a result, G_k is a regular graph for CSP.

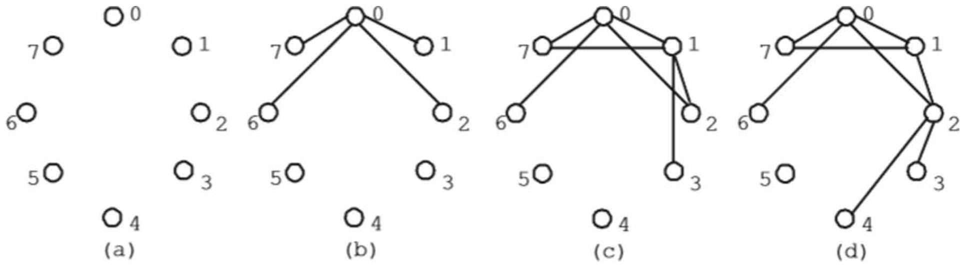
The proposed constant storage protocol is described in Algorithm 1. We assume that there are in total n sensors in the sensor network and each sensor contains k keys.

Algorithm 1: Constant Storage Protocol

```
Require: Each sensor has its own identity. Assume that we have n sensors and the identities range
    from 0 to n−1.
Require: Sensors are arranged in a clockwise manner based on their identities.
Require: Each sensor stores exactly k keys. Let C be a set.
for $i = 0$ to $n - 1$ do
    for $j = 1$ to $\frac{k}{2}$ do
        if the edge $(i, i + j \pmod{n})$ does not exist then
            Add an undirected edge $(i, i + 1 \pmod{n})$.
        end if
        if the edge $(i, i - j \pmod{n})$ does not exist then
            Add an undirected edge $(i, i - j \pmod{n})$.
        end if
    end for
end for
for each edge do
    Separately embed a key into the nodes in the ends of the edge.
end for
```

Consider the case that for a given sensor with label i, it shares keys, $K_{i,i-\frac{k}{2}}$, $K_{i,i-\frac{k}{2}+1}$, …, $K_{i,i-1}$, $K_{i,i+1}$, …, and $K_{i,i+\frac{k}{2}}$ with the sensors having labels $i-\frac{k}{2}$, $i-\frac{k}{2}+1$, …, i - 1, i + 1, …, and $i+\frac{k}{2}$ respectively. Figure 1 shows an example for the case with n = 8 and k = 4. The keys in the

Figure 1. An example for CSP with 8 sensors, each of which is assigned 4 keys. (a) The initialization of algorithm; (b) The first step of the graph construction, starts the process in the sensor with identity 0; sensor 0 shares keys with sensor 1, 2, 6, 7 by $K_{0,1}$, $K_{0,2}$, $K_{0,6}$, $K_{0,7}$. (c)(d) The second step (respectively, third step), starts the process in the sensor with identify 1(respectively, 2). In CSP, K_{pq} is equal to K_{qp}, where p and q are identifies of sensors.



sensor networks assigned in this way can also be pairwise because we can assign different keys to each edge.

In CSP, the message can be conveyed from the source sensor to the destination sensor in either clockwise or counterclockwise. In order words, there are more than two different transmission paths between the source and target. This advantage definitely increases the security of networks. Figure 1 also shows how the source sensor sends the message to the destination sensor using peer intermediaries. For example, when the identities of the source and destination sensor are 0 and 5, respectively, there are several routes can be used to deliver the information. Sensor 0 can clockwise send messages to sensor 5 using sensors 2 and 4 as peer intermediaries successively, or can counterclockwise send messages using only sensor 6 as peer intermediary. Note that if the identity difference between the source sensor and destination sensor is within k, then according to the construction of CSP they have had shared a common key. Therefore, they do not need other sensors as intermediaries. In addition, a transmission path between any two nodes in the network cross at most n k intermediaries.

In the ideal case, from the viewpoint of key sharing when a node would like to send message to another sensor, it has k neighbors. Note that source sensor knows the IDs of their neighbor sensors. The source sensor chooses one of these neighbor sensors depended on which of these neighbor sensors has the best routing metric such as practical distance and the strength of signal.

2.2. Incremental Deployment

If the area that we want to monitor is quite small, or we just want to focus on a specified region, then it is sufficient to deploy a subset of sensors. When it is required to expand the monitored region, we just deploy remaining sensors incrementally. For example, let the sensors be labeled as 0, 1, ..., n−1. Let S be the set containing all sensors, and let S_1 and S_2 be the subsets of S. Let the labels of sensors in S_1 be 0, 1, …, $n_1 - 1$, and let those of sensors in S_2 be $n_1 + 1, \ldots, n_2 - 1$. We can first deploy S_1, and then deploy S_2 in the future if necessary.

Since the number of keys assigned for each sensor is k, there are at least $\frac{n_1}{\mathrm{k}}$ intermediaries in S_1. The problem we need to concern here is if S_1 is deployed ahead of S_2, then how is the connectivity between S_1 and S_2.

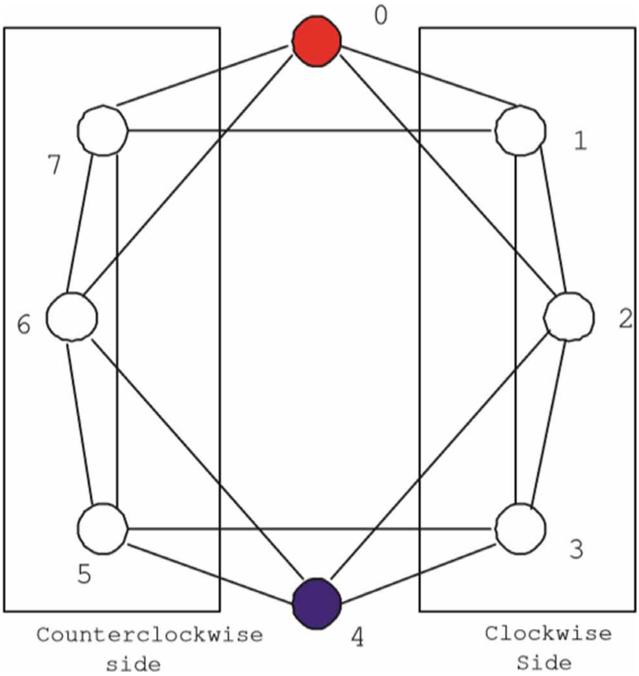
Since we know that the node n_1 contains k keys, the node $n_1, n_1 + 1, \ldots, n_1 + \frac{\mathrm{k}}{2} - 1$, must share common keys with $n_1 - 1$. By the labeling way we use, the above nodes must be in S_2. Similarly, the nodes $n_1 + \frac{\mathrm{k}}{2} - 2, n_1 + \frac{\mathrm{k}}{2} - 3, \ldots, n_1 - 1$ share common keys with node $n_1 - 2$. So, in one group, there are at least $\frac{\mathrm{k}}{2}$ nodes can communicate with the other groups, those nodes can be used as bridges between two groups. The keys between two different groups are $\frac{k^2 + 2k}{8}$. So, we can modulate the number of keys and bridges by setting up the number of keys contained in each node. According to this discussion, CSP is suitable for group deployment. We will analyze more properties about CSP for group deployment later.

2.3. The Number of Different Available Paths

In this section, we will analyze the number of different available paths between the source and target nodes. First, we introduce the property, cross side, of a path connecting two nodes.

Definition 1 Given a circular graph and two nodes in the graph, the remaining nodes will be divided into two sides, clockwise side and counterclockwise side, as shown in Figure 2. The path connecting these two nodes has the property that at least one node belongs to the clockwise side and at least one node belongs to the counterclockwise side. This property is called cross side here.

Figure 2. Cross side in a graph, where the source node is labeled as 0 and the target node is labeled as 4



The following lemma describes the number of different available paths of length r connecting the source nodes and target node t.

Lemma 2 Let the identities of the source node and target node be s and t respectively, where t ≥ s. The total number of different available paths with fixed number of communications r that have cross side connecting the source node s and target node *t* is:

$$\sum_{j=0}^{r}\left(-1\right)^{j}\binom{r}{j}\left(\binom{r+t-s-\frac{k}{2}-1}{t-s-\frac{k}{2}}+\binom{r+n-t+s-\frac{k}{2}-1}{n-t+s-\frac{k}{2}}\right)$$

where r is a positive integer.

Proof. The identity distance between the identity of the source node s and the target node t is t−s. Hence, given the fixed number of communications, r, each transmission path of length x_i falls between 1 and $\frac{k}{2}$, i.e., $1 \le x_i \le \frac{k}{2}$. We have $x_1 + x_2 + \dots + x_r = t - s$. Under the situation of considering counter clockwise side, we have $x_1 + x_2 + x_3 + ... + x_r = n - t + s$. After some calculations, the number of different available paths connecting two nodes in clockwise and counter clockwise directions is $\sum_{j=0}^{r}\left(-1\right)^{j}\binom{r}{j}\binom{r+t-s-\frac{k}{2}-1}{t-s-\frac{k}{2}}$ and $\sum_{j=0}^{r}\left(-1\right)^{j}\binom{r}{j}\binom{r+n-t+s-\frac{k}{2}-1}{n-t+s-\frac{k}{2}}$, respectively.

Consequently, the sum of two values:

$$\sum_{j=0}^{r}\left(-1\right)^{j}\binom{r}{j}\binom{r+t-s-\frac{k}{2}-1}{t-s-\frac{k}{2}}+\sum_{j=0}^{r}\left(-1\right)^{j}\binom{r}{j}\binom{r+n-t+s-\frac{k}{2}-1}{n-t+s-\frac{k}{2}}$$

$$=\sum_{j=0}^{r}\left(-1\right)^{j}\binom{r}{j}\left(\binom{r+t-s-\frac{k}{2}-1}{t-s-\frac{k}{2}}+\binom{r+n-t+s-\frac{k}{2}-1}{n-t+s-\frac{k}{2}}\right)$$

is the number of different available paths with fixed number of communications r.

Suppose that the difference between source node and target node is sufficiently less than $\frac{n}{2}$, we can only consider the paths in clockwise side. In this case, the length of path connecting source node s and target node t is at least $\frac{t-s}{\frac{k}{2}}$, because an edge in the path crosses at most $\frac{k}{2}$ nodes. We know that if r $\leq \frac{t-s}{\frac{k}{2}}$ then there are no such paths existed. Apparently, the number of shortest path connecting source and target node is:

$$\sum_{j=0}^{m}\left(-1\right)^{j}\binom{m}{j}\binom{m+t-s-\frac{k}{2}-1}{t-s-\frac{k}{2}}$$

where m = $\frac{t-s}{\frac{k}{2}}$. By the above lemma, there are a lot of apparent paths between s and t. In fact, the more path between s and t can enhance the security during transmission.

2.4. Communication Overhead

Here we analyze communication overhead for CSP. We first take the unit square as the simplest case and calculate the average distance between two points in it.

Lemma 3 Given a unit square V. Let $x = (x_1, x_2)$ and $y = (y_1, y_2)$ be uniformly selected from V. The average distance between x and y is about 0.521.

Proof. Let V be a unit square centered at original and let x_1, x_2, y_1, y_2 be four random variables which are independent identically distributed in $\left[\frac{-1}{2}\ \frac{1}{2}\right]$. Let s_x, s_y, t_x, t_y be four random variables defined as: $s_x = x_1 - x_2$, $s_y = y_1$-y_2, $t_x = x_1 + x_2$ and $t_y = y_1 + y_2$. It is trivial to obtain $t_x \in$[max{$s_x - 1$, -s_x - 1}, min{-s_x + 1, s_x + 1}] = [$\left|s_x\right|$-1, -$\left|s_x\right|$+1], and $t_y \in$[$\left|s_y\right|$ − 1, -$\left|s_y\right|$ + 1].

Therefore, the average distance between x and y is:

$$E\left[\sqrt{\left(x_1-x_2\right)^2+(y_1-y_2)^2}\right]=\int_V\sqrt{\left(x_1-x_2\right)^2+\left(y_1-y_2\right)^2}\,dx_1dx_2dy_1dy_2$$

$$=\frac{1}{4}\int_V\sqrt{s_x^2+s_y^2}\,ds_xds_ydt_xdt_y \tag{1a}$$

By considering s_x, s_y ∈[0,1], Equation (1a) can be rewritten as:

$$=4\int_{\theta=0}^{\frac{\pi}{2}}\int_{r=0}^{r(\theta)}\left(r\cos\theta-1\right)\left(r\sin\theta-1\right)\times r\times rdrd\theta \tag{2a}$$

where $s_x = r\cos\theta$, $s_y = \mathrm{r}\sin\theta$, $\theta\in$ [0, $\frac{\pi}{2}$], and r ∈ [0, r(θ)].

Again, by Equation (2a):

$$2\text{ x }4\int_{\theta\in\left[0,\frac{\pi}{4}\right]}\int_{r\in[0,\sec\theta]}r^4\cos\theta\sin\theta-r^3\cos\theta-r^3\sin\theta+r^2drd\theta$$

$$=8\int_{\theta=0}^{\frac{\pi}{4}}\left(\frac{1}{5}sec^3\theta\sin\theta-\frac{1}{4}sec^3\theta-\frac{1}{4}sec^4\theta\sin\theta+\frac{1}{3}sec^3\theta\right)d\theta \tag{3a}$$

After some numerical calculations, Equation (3a) is calculated as:

$$8\{\frac{-1}{20}\left(\frac{2\sqrt{2}}{3}-\frac{1}{3}\right)+\frac{1}{12}\left[\frac{1}{2}\left(\sqrt{2}+\ln\left(1+\sqrt{2}\right)\right)\right]\}=8\{\frac{-2\sqrt{2}+1}{60}+\frac{\sqrt{2}+\ln\left(1+\sqrt{2}\right)}{24}\}\approx 0.521$$

According to lemma 2, if the sensors are uniformly distributed on a square area A, then the average distance between any two sensors is about 0.521$\sqrt{A}$. According to (Li et al., 2001), there are O($\sqrt{n}$)) hops required to traverse this distance. As a result, the hop-distance between two sensors in the network is $\frac{\alpha}{\sqrt{n}}$, where α is a constant depending on the transmission range of each sensor and the environment factors of the target region. In CSP, a path between two sensors may pass through $\frac{\mathrm{n}}{k}$ intermediaries. The expected communication overhead of a path is $\frac{0.521\left(\frac{\mathrm{n}}{\mathrm{k}}+1\right)\sqrt{A}}{\mathrm{O}\left(\sqrt{n}\right)}$ hops away. By this results, the maximum transmitting distance between any two sensors in our protocol is proportional to O($\sqrt{n}$) hop-distance.

2.5. Security and Resilience Against Node Compromise

In CSP, a path between two nodes is compromised if any one of its endpoint nodes between them is compromised, or if any one of the intermediary nodes is compromised. Let p denote the fraction that the total number of nodes in the network is compromised. The fraction of total paths compromised is about $1\text{-}(1 - p)^{2+m}$, where m is the number of intermediary nodes. If p is sufficiently small, the value approximates $(2 + m)p$. Under the ideal case, if two nodes can exchange messages directly without passing through any intermediaries, this value is 2m (Chan & Perrig, 2005). So, the number of intermediary nodes should not be very large under hostile environment. In CSP, the number of intermediaries $\frac{n}{k}$, so the number of keys contained in each sensor should be adjusted in order to accustom the environment.

The above results also indicate an important property of secure networks. The adversary may use a variety of attacking methods to compromise some nodes. If some nodes are compromised by the adversary, the networks will lose some secure links. So, it is necessary to define the fraction of communications compromised. By (Chan & Perrig, 2005), the fraction of communications compromised is total number of compromised links divide total number of links. By previous discussion, the number of compromised links are about twice as much as the number of compromised nodes under ideal case. In Section 3, we will do some comparisons for CSP and other proposals.

2.6. Comparison With Previous Key Establishment Scheme

In the two-dimensional case of PIKE, it has a fascinating property that for any two nodes they are either directly connected or connected with a path of length 2 so that the transmission range of each sensor can reach all the other sensors in the network. As a n-dimension case for PIKE, there is a path having length at most n connecting two sensors for each other. Although CSP constructed as above loses the advantage that for any two nodes they are either directly connected or connected with a path with length 2 from the viewpoint of key sharing between sensors, the resulting graph of CSP has another favor. It has a kind of cluster-like property, that for any specified node it is more heavily connected with its neighbor sensors. It is important to have cluster-like property for the connectivity in some cases. For example, the sensor networks based on group-based deployment is a case. Group-based deployment is a kind of non-uniform deployment having the property that sensors in the networks are grouped and distributed in some specified locations. In this case, for those specified locations sensors are more closed. Due to this fact, if the key establishment scheme has cluster-like property, the connectivity in the subgroups are higher than one without cluster-like property with higher probability. Because of this fact, in the following we state two lemmas describing the behavior of the number of corresponding edges in two kinds of deployments, uniform deployment and group-based deployment respectively.

Before stating the lemma, under the assumption that the total number of sensors is n we first define the C-consecutive group as a group of sensors consisting of C sensors whose identities ranged from i to $i + C - 1$ (mod n), and the C-monotone increasing sequence as a sequence consisting of C integers $a_1, a_2, \ldots, a_C$ where $a_{m+1} - a_m = 1$ for $m = 1,\ldots, C - 1$.

The reason why we care about the edge of a C-consecutive group is that, if the group-based deployment is used, a series of sensors are deployed on the specified point in the target region, forming a set of subgroups which is composed of a number of sensors for each subgroup. The characteristic of such deployment is that from the viewpoint the transmission range, in contrast to uniform deployment the sensors in the same subgroup is more likely to cover each other. In addition, the assumption of group-based deployment is that the sensors have inclination that is more likely to communicate with the other sensors in the same subgroup. According to these two characteristics and the observation that the so-called secure communication between two sensors exists provided that these two cover and share at least one key each other, if there are more edges, that means the level of key sharing, in a subgroup, the secure communication is easier to achieve.

Lemma 4 Given the construction of CSP, sensors and the expected degree of each sensor, for any C-consecutive group that $C > \frac{k}{2}$ with $a_1 = i$, it contains:

$$\frac{\mathrm{k}}{2}x + \frac{k^2}{8} - \frac{k}{4} \text{ Edges}$$

where:

$$x = C - \frac{k}{2}$$

Proof. Without loss of generality, we assume that the identities of the sensors in *C*-consecutive group are form a *C*-monotone increasing sequence. Obviously, for the first x nodes in *C*-consecutive group, they have exactly $\frac{k}{2}$ edges connecting to the nodes whose identities are larger than themselves. In addition, due to the bound C, there are some nodes having less edges connecting to the nodes whose identities are larger than themselves. The number of such nodes is exactly $\frac{k}{2}$. In addition, the number of edges connected between these nodes and their clockwise neighbors is decreased by one each time. It can be formulated by $\frac{\mathrm{k}}{2} - 1 + \frac{\mathrm{k}}{2} - 2 + \ldots 2 + 1 = \frac{k^2}{2} - \frac{\mathrm{k}}{4}$. To sum up, there are $\frac{\mathrm{k}}{2}x + \frac{k^2}{8} - \frac{\mathrm{k}}{4}$ edges, where $x = C - (\frac{\mathrm{k}}{2})$.

Here, given the length of *C*-consecutive group we also calculate the number of edges in the *C*-consecutive group for PIKE. The lemma describing it goes as follows.

Lemma 5 Given the PIKE construction with n × n square, for any C-consecutive group with $a_1 = i$, it contains $\frac{1}{2}\left(\begin{array}{l}\left(n - c_1\right)\left(n - c_1 + 1\right) + c_2\left(c_2 - 1\right) + n\left(n - 1\right)\left(r_2 - r_1 - 1\right) \\ + n\left(r_2 - r_1\right)\left(r_2 - r_1 + 1\right) - 2\left(r_2 - r_1\right)\left(n - c_2 + c_1 - 1\right) + 2\delta_c\left(c_1 - c_2 - 1\right)\end{array}\right)$ edges, where r_1 is the minimal integer such that $r_1 n \geq i$, r_2 is the minimal integer such that $r_2 n \geq i + C$, $c_1 = i - (r_1 - 1) \times n$, $c_2 = i + C - (r_2 - 1) \times n$ and δ_c is 1 if $c_1 > c_2$, 0 otherwise.

Proof. At first, we calculate the number of horizontal incident edges. At first, the row the first node in consecutive group placed is called head row, r_1, and the row the last node in consecutive group placed is called tail row, r_2, and the others are called torso row. Moreover, the column the first node in consecutive group placed is called head column, c_1, and the column the last node in consecutive group placed is called tail column, c_2, and the others are called torso column. We divide the computation into three parts. We count the number of nodes in the head row and further estimate the number of its horizontal incident edges. The same way also applied to the tail row and torso row. Afterward, we can obtain $(\mathrm{n}-\mathrm{c}_1)(\mathrm{n}-\mathrm{c}_1+1)$ for head row, $c_2(c_2-1)$ for tail row and $\mathrm{n}(\mathrm{n}-1)(r_2-r_1-1)$ for torso row. In addition, we must calculate the number of vertical incident edges. We start by counting the number of edges in the square formed by the r_1-th row and r_2-th row, which is $\mathrm{n}(r_2 - r_1)(r_2 - r_1 + 1)$. Then we subtract the number of edges which connect with at least one node which is not in the consecutive group from $n(r_2 - r_1)(r_2 - r_1 + 1)$. Nevertheless, when $c_1 > c_2$, some edges connected with the nodes within the middle part between c_1-th column and c_2-th column will be double-subtracted. On account of this reason, if $c_1 > c_2$ then we add the value $2(c_1 - c_2 - 1)$ to it. Last, because of the double counting for each edge, the value derived above should divided by 2.

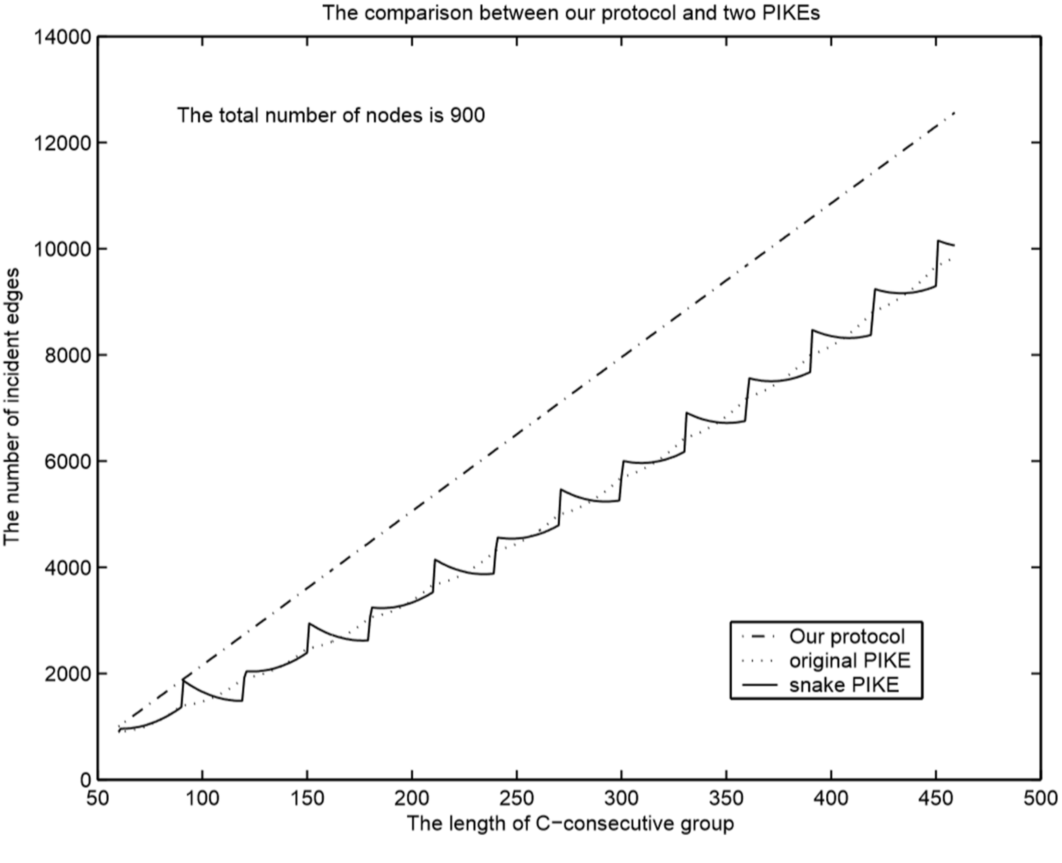
In later section, we will compare the above two results using numerical experiments.

3. EXPERIMENTAL RESULT

In section 3, a series of experiments were conducted to verify the advantages of our method in comparison with two variations of PIKE under group deployed. The two variations of PIKE include original PIKE (Chan & Perrig, 2005) and snake PIKE that we have introduced in section 1.

Since in group-based deployment the sensors are deployed nonuniformly into several subgroup and the sensors in the same subgroup have more chance to communicate with each other, this motivate us to use "number of incident edges" to measure the communication efficiency. In fact, the number of incident edges corresponds to number of key sharing. Figure 3 shows that in group-based deployment the number of edges obtained in the proposed scheme is more than that in both variations of PIKE. On the other hand, once we have more edges in a deployment, the sensors in the same subgroup are likely to communicate with each other using the keys already stored in their key rings rather than using intermediary sensor to retransmit messages, which imply higher communication overhead.

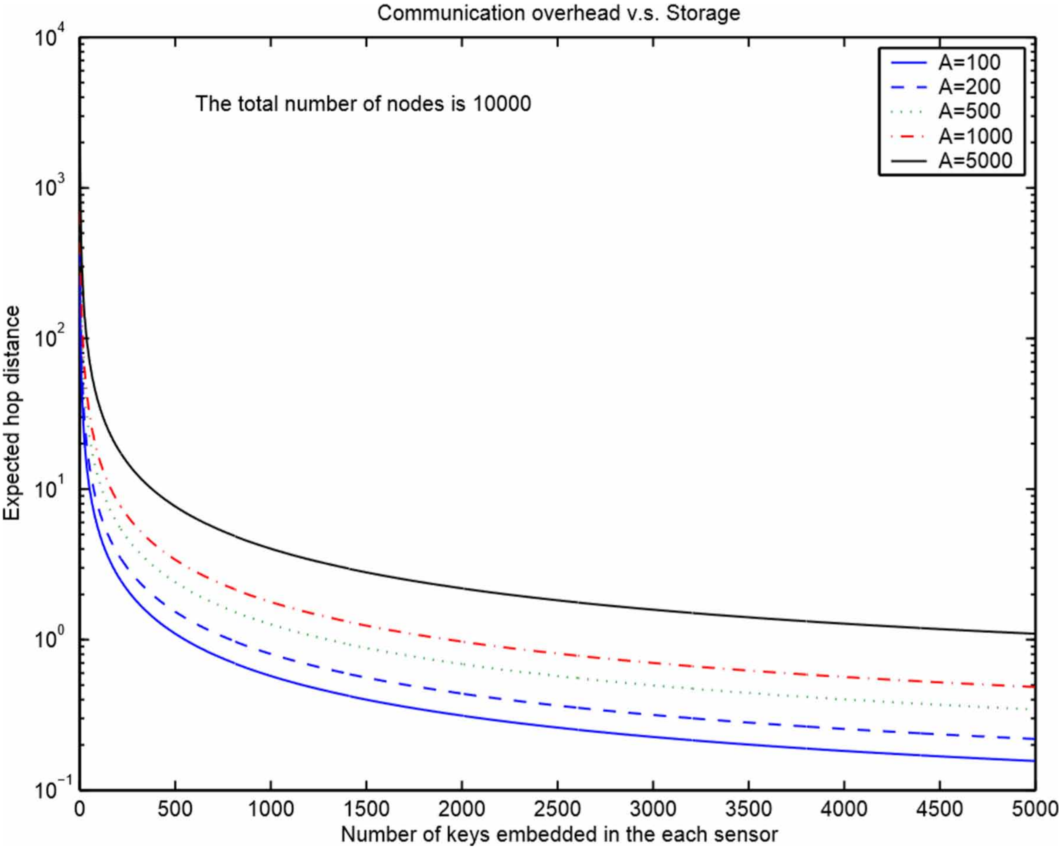
Figure 3. The comparison between the proposed scheme and two variations of PIKE. In this experiment, the number of sensors is 900, the cardinality of a subset of sensors is ranged from 60 to 460.



In other words, the communication overhead can be reduced. Moreover, the number of edges also related to the size of a subgroup.

Figure 4 shows the relationship between communication overhead and the number of keys embedded in each sensor. From Section 2.4, we know the communication overhead is determined by the number of sensors, the size of target region, and the degree of each sensor. The free variable we can control here is the degree of each sensor. Since the expected communication overhead of a link is:

Figure 4. The representation between the communication overhead and the number of keys embedded in each sensor



$$\frac{0.521\left(\frac{\mathrm{n}}{\mathrm{k}}+1\right)\sqrt{A}}{\sqrt{n}}$$

hops away, we respectively adjust the area size, the total number of sensors and the difference between source node and target node to observe the relation between them. We further observe that regardless of the size of a target region increasing the degree of each sensor can have the same proportional effect of reducing the communication overhead.

As we know at first, the resilience is worse than original PIKE. However, it still has satisfiable requirement. As shown in our experiment, when the intermediaries is 10, the fraction of total communications compromised is below 0.5 after approximately $\frac{200}{10000} = \frac{2}{100}$ of sensors were captured. However, under this circumstance, the keys embedded in each sensor is less than 1000.

4. CONCLUSION AND FUTURE WORKS

In this paper, we illustrate a key management scheme, called CSP, and discuss its properties. Now we compare the two protocol with PIKE (Chan & Perrig, 2005). In the following, we assume the total number of sensors be n. In PIKE (Chan & Perrig, 2005), each node contains $2(\sqrt{n}-1)$ keys, and total number of keys that this scheme needs is $n(\sqrt{n}-1)$. In worst case, a link in PIKE (Chan & Perrig, 2005) may need one intermediary. In CSP, the number of keys contained in each node, the total number of keys in the network, and the most intermediaries of each link needs is *k*, *nk*, and $\frac{\mathrm{n}}{\mathrm{k}}$. The total number of keys and maximal number intermediaries should be kept smaller to save the cost and get better secure properties. So we calculate the product P_A, as a measure to identify the balance between storage and communication overhead, which is the total number of keys times the maximal number of intermediaries for a specified scheme A. By the above discussion, the value of P_A should be as small as possible. P_{PIKE} is equal to $n\left(\sqrt{n}-1\right)$, $P_{CSP} = n^2$. CSP is a suitable protocol for group-based deployment.

In this paper, we do not consider the properties about transmission range of each node. However, in real model, if two sensors want to build a secure link, they must satisfy the restrictions on both the transmission range and sharing at least one common key. So, we should combine this two requirements simultaneously in the future.

ACKNOWLEDGMENT

Authors would like to thank the anonymous referees for their constructive comments and valuable suggestions. This work is supported by the Collaborative Innovation Center for Economics crime investigation and prevention technology, Jiangxi Province, China.

REFERENCES

Bettstetter, C. (2002, June). On the minimum node degree and connectivity of a wireless multihop network. Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing (pp. 80-91). ACM. doi:10.1145/513800.513811

Chan, H., Perrig, A., & Song, D. (2003). Random Key Predistribution Schemes for Sensor Networks. *Proceedings of the IEEE Symposium on Security and Privacy*. Academic Press.

Chan, H., & Perrig, A. (2005, March). PIKE: Peer intermediaries for key establishment in sensor networks. []. IEEE Press.]. *Proceedings - IEEE INFOCOM*, *1*, 524.

Chan, S.-P., Poovendran, R., & Sun, M.-T. (2006). A Key Management Scheme in Distributed Sensor Networks Using Attack Probabilities. Proc. IEEE GLOBECOM 2005. IEEE Press.

Du, W., Deng, J., Han, Y. S., Chen, S., & Varshney, P. K. (2004). A key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge. Proceedings of IEEE INFOCOM 2004 (Vol. 1). IEEE Press.

Du, W., Deng, J., Han, Y., & Varshney, P. (2003). A Pairwise Key PreDistriubtion Scheme for Wireless Sensor Networks. *Proceedings of the Tenth ACM Conference on Computer and Communications Security (CCS)* (pp. 42-51). ACM.

Eschenauer, L., & Gligor, V. (2002). A Key-management Scheme for Distributed Sensor Networks. *Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS)* (pp. 41-47). ACM. doi:10.1145/586110.586117

Li, J., Blake, C., DeCouto, D., Lee, H. I., & Morris, R. (2001). Capacity of ad hoc wireless networks. *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*. ACM.

Liu, D., & Ning, P. (2003). Establishing Pairwise Keys in Distributed Sensor Networks. *Proceedings of the 10th ACM Conference on Computer and Communications Security*. ACM. doi:10.1145/948109.948119

Rong, Y., Choi, H., & Choi, H.-A. (2004). Dual Power Management for Network Connectivity in Wireless Sensor Networks. *Proceedings of the 18th International Parallel and Distributed Processing Symposium*. Academic Press.

Santi, P., Blough, D. M., & Vainstein, F. (2001, October). A probabilistic analysis for the range assignment problem in ad hoc networks. Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing (pp. 212-220). ACM. doi:10.1145/501416.501446

Steiner, J. G., Neuman, B. C., & Schiller, J. I. (1988, February). Kerberos: An Authentication Service for Open Network Systems. *Proceedings of Usenix Winter* (pp. 191-202). Academic Press.

Traynor, P., Choi, H., Cao, G., Zhu, S., & Porta, T. L. (2006). *Establishing Pairwise Keys in Heterogeneous Sensor Networks*. Proceedings of IEEE INFOCOM. IEEE Press.

Wacker, A., Knoll, M., Heiber, T., & Rothermel, K. (2005). A new approach for establishing pairwise keys for securing wireless sensor networks. *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM.

Jingyuan Rao attended Nanchang University School of Software. Research interests include: data deduplication, sensor network, information security.

Xuanjin Yang was boarn in Nanching. He earned his Master of Education. His research interests include: education informatization.