Journal of Network and Computer Applications 82 (2017) 114–127

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

A novel key generation method for wireless sensor networks based on system of equations

Furui Zhan, Nianmin Yao*, Zhenguo Gao, Guozhen Tan

Dept. Computer Science and Technology, Dalian University of Technology, Dalian, China

ARTICLE INFO

Keywords:
Key generation
System of equations
Key connectivity
Key management
Wireless sensor networks

ABSTRACT

Many key management schemes were proposed for protecting wireless sensor networks (WSNs). While applying key management to the network, it is important to ensure that the efficiency of the network is not greatly affected by key connectivity. Poor connectivity might lead to many messages forwarding. Consequently, a large amount of energy of the involved nodes would be consumed during message forwarding, which is not suitable for the resources-constraint sensor nodes. In this work, we analyze the impact of key connectivity on the efficiency of communication. Then, a novel key generation method based on system of equations is proposed to improve key connectivity of key management. The involved equations are applied to establish secret keys and each node uses these keys for protecting their communication. The system of equations is constructed to have one and only one solution so that the unique solution can also be used to establish a shared hidden key for enhancing the association among nodes. As a result, neighbors can directly communicate with each other through the shared hidden key even though they do not have common keys. To differentiate from normal keys, keys generated by the proposed method are called associated-keys. According to the analyses, we recommend that systems of linear equations (linear systems) instead of systems of polynomial equations are used to realize the proposed method with respect to the computation complexity. Furthermore, we illustrate that linear systems of two variables are sufficient to generate keys for large scale of networks. The Exclusion Basis System (EBS) is used as a instance to illustrate the implementation of key management with associated-keys. The theoretical analyses and simulation results show that key management schemes with associated-keys have better key connectivity than the corresponding schemes with normal keys. Meanwhile, other performance metrics are unaffected.

1. Introduction

Nowadays, wireless sensor networks (WSNs) are applied into various fields (Rashid and Rehmani, 2015), such as military, transportation and healthcare. In these applications, the efficiency and security of communication are very important. Typically, key management is used as a critical security service for protecting WSNs (Ying et al., 2011).

According to Eltoweissy et al. (2006), a key management process consists of four components: key analysis, key assignment, key generation, and key distribution. The existing key management schemes can be classified into various categories, such as probabilistic schemes and deterministic schemes (Zhang and Varadharajan, 2010; He et al., 2013). For these schemes, key connectivity is an important metric which indicates the ability of secure communication after key management is applied. Accordingly, key connectivity significantly affects the efficiency and security of networks. For many key management schemes, key connectivity is lower than 1. Although key connectivity can achieve 1 in some schemes, these schemes either have many constraints or sacrifice other metrics, e.g., poor scalability or requirement of deployment knowledge. When key connectivity of the applied key management schemes is lower than 1, it is impossible to ensure that common keys can be discovered among all neighbors. As a result, many messages forwarding need to be processed, which leads to consuming a large amount of energy and other precious resources of sensor nodes. What's more, during messages forwarding, the authentication of nodes have to be implemented for guaranteeing the security of these processes. When key management is used for clustered WSNs, the implementation of clustering might be affected if its key connectivity is lower than 1. Therefore, the referred key connectivity is a critical metric of key management.

In this work, we focus on key connectivity of key management. Moreover, to enhance the key connectivity without sacrificing other metrics, a novel key generation method based on system of equations is

* Corresponding author.
E-mail addresses: izfree@mail.dlut.edu.cn (F. Zhan), lucos@dlut.edu.cn (N. Yao), gzg2012@dlut.edu.cn (Z. Gao), gztan@dlut.edu.cn (G. Tan).

http://dx.doi.org/10.1016/j.jnca.2017.01.019
Received 25 March 2016; Received in revised form 15 November 2016; Accepted 18 January 2017
Available online 21 January 2017

proposed. The main contributions of our work are described as follows:

- We analyze the impact of key connectivity on WSNs. Specifically, we illustrate the case where two neighbors, who do not have common keys, want to communicate with each other.
- To enhance key connectivity, we propose a novel key generation method based on system of equations. The system of equations is defined as eligible system (ES) when it has one and only one solution. Each equation in the applied eligible system is applied to generate a secret key for the network. The generated keys are called associated-keys in contrast to normal keys. As a result, the unique solution can be used to establish a shared hidden key for nodes and neighbors can establish secure link by the shared hidden key even when they do not have common keys. Both system of polynomial equations and system of linear equations are illustrated to implement the proposed method. Taking into account computation complexity, we recommend to use system of linear equations to generate secret keys for the network.
- We use linear system of two variables to illustrate the implementation of the proposed method and use the generated keys to achieve key management. Moreover, the Exclusion Basis System (EBS) (Eltoweissy et al., 2004) in conjunction with associated-keys is used as an instance of key management system.
- The theoretical analyses and simulations are conducted to evaluate the proposed method. During simulations, associated-keys are applied into different EBS (Eltoweissy et al., 2004) and Unital schemes (Bechkit et al., 2013) to create new key management schemes. Then, these schemes are compared with the corresponding schemes with normal keys. The results show that the proposed method can be used to enhance key connectivity of key management without sacrificing other metrics.

The remainder of this paper is organized as follows: in Section 2, we review the related work on key management. The impact of key connectivity is analyzed in Section 3. The key generation method based on system of equations is illustrated in Section 4. Section 5 describes the implementation of the proposed method. In Section 6, simulations are conducted to evaluate the performance of the proposed method. Finally, the conclusions of this work are described in Section 7.

2. Related work

Many key management schemes have been proposed for WSNs. Although some schemes apply asymmetric cryptography were proposed (Malan et al., 2004; Rajendiran et al., 2011; Nam et al., 2014), most schemes applied symmetric cryptography with respect to computation complexity and energy consumption. In this section, only the key management schemes based on symmetric cryptography are described.

2.1. RKP and RKP-based schemes

Eschenauer and Gligor proposed a random key pre-distribution scheme (RKP scheme) for wireless sensor networks (Eschenauer and Gligor, 2002). The scheme consists of three components: key pre-distribution, shared-key discovery and path-key establishment. In key pre-distribution phase, a large key pool is initialized and the identifiers of keys are determined. Each node randomly selects k keys to store. In the shared-key discovery phase, each node exchanges the identifiers of keys with neighbors and identifies the shared keys with neighbors. The path-key establishment phase is implemented if the shared keys cannot be found between the communicating parties. In this phase, several intermediate nodes capable of directly communicating with them are selected to accomplish the communication. This scheme is energy efficient, but the storage overheads are high. The key connectivity of this scheme can be figured out as

$$p' = 1 - \frac{(1 - \frac{k}{|S|})^{2(|S|-k+\frac{1}{2})}}{(1 - \frac{2k}{|S|})^{(|S|-2k+\frac{1}{2})}}$$

where $|S|$ denotes the size of key pool, and k is the number of keys stored in each node. p' is the key connectivity of this scheme. It can be found that the resulting connectivity is lower than 1.

Based on Eschenauer and Gligor (2002), Chan et al. proposed a modified scheme called q-composite keys scheme (Chan et al., 2003). In this solution, neighbors can establish a secure link only if they share at least q keys and thus the resilience against node capture is enhanced. In Du et al. (2003), a key pre-distribution scheme that combined the RKP scheme and Blom's scheme (Blom, 1985) was proposed to improve the resilience against node capture. Similarly, a key pre-distribution scheme based on the RKP scheme was proposed in Liu et al. (2005), where bivariate t-degree symmetric polynomials instead of matrix were used to generate shared keys between nodes.

2.2. EBS and EBS-based schemes

The Exclusion Basis System (EBS) is a combinatorial optimization methodology for group key management scheme (Eltoweissy et al., 2004). In EBS, each node is assigned k keys out of a pool of size $P = k + m (1 < k, m < n)$, where P is the size of key pool and n denotes the size of the network. That is, m keys are unknown to each node. According to Eltoweissy et al. (2004), the referred parameters have to meet the relationship $\binom{k+m}{k} \geq n$. As a result, if a node is compromised, this node can be evicted by broadcasting the rekeying messages which contain the replacement of k exposed keys and are encrypted by the corresponding m unknown keys. Consequently, the key system is updated.

Younis et al. proposed a location-aware dynamic key management scheme based on EBS (Younis et al., 2006). With the deployment information, the resilience can be enhanced by decreasing the Hamming distances of key strings stored by neighbors. In Eltoweissy et al. (2006), a novel dynamic key management scheme was proposed, which is called localized combinatorial keying (LOCK). This scheme is implemented in clustered WSNs and the polynomial keys are applied to enhance the resilience of key management. Besides, several key management schemes based on EBS have been proposed (Moharrum et al., 2006; Ying et al., 2011; Lo et al., 2009; Syed et al., 2010).

Comparing with RKP-based schemes, EBS-based schemes can efficiently evict the compromised node and update the key system. Therefore, these schemes can provide long-term and flexible protection for WSNs.

2.3. Combinatorial design schemes

Several key management schemes based on combinatorial design were proposed (Camtepe and Yener, 2007; Ruj et al., 2011, 2013; Bechkit et al., 2013). In Camtepe and Yener (2007), Camtepe et al. proposed a key pre-distribution scheme based on Symmetric Balanced Incomplete Block Design (SBIBD). The SBIBD scheme performs good key connectivity. However, this scheme cannot be used for large scale networks. Pairwise and triple key distribution schemes were proposed by Ruj et al. in Ruj et al. (2011), where Steiner trade is applied for key establishment. The scheme is highly resilient against node capture attacks. In Bechkit et al. (2013), Bechkit proved that Ruj's scheme provided a low session key sharing probability and then proposed a new scheme based on unital design theory. The scheme provides high network scalability and good key sharing probability approximately lower bounded by $1 - e^{-1}$. Comparing with random key pre-distribution schemes, the key connectivity in combinatorial design schemes is improved. However, the construction of appropriate a combinatorial design for the given network. In addition, these schemes do not have

good flexibility.

2.4. PRF-based schemes

The pseudo-random functions (PRF) are another popular solutions to implement key management for WSNs. In Zhu et al. (2006), a scheme, LEAP+, was proposed for protecting homogeneous WSNs. In this scheme, four types of keys were established by pseudo-random functions. In addition, two key management schemes with pseudo-random functions were proposed for protecting large-scale WSNs in Das (2012a) and Das (2012b).

Besides, to improve the performance of key management, several schemes employ auxiliary devices (Dong and Liu, 2012; Mi et al., 2010; Das and Sengupta, 2007). In Liu and Ning (2005) and Liu et al. (2008), deployment knowledge is also used to implement key management. Without auxiliary devices or location information, the existing schemes generally enhance some performance metrics with sacrificing other metrics, which is not practical in many scenarios. In this work, we propose a key generation method to enhance the key connectivity without sacrificing other metrics.

3. The impact of key connectivity on WSNs

Key management is applied to establish and maintain secure links among communicating nodes. However, the improvement of security accompanies with the reduction of communication efficiency. Actually, some neighbor nodes cannot find common keys if the network is protected by a key management scheme whose key connectivity is lower than 1. As a result, the direct communication between these neighbors is not available and thus secure key-paths have to be established for these communication. These processes consume large amount of energy of sensor nodes and the authentication of relay nodes must be considered as well. Fig. 1 illustrates the aforementioned case.

As presented in Fig. 1, nodes A, B, C and D are neighbors with each other and the communication radius of each node is R. In this case, only full lines between nodes indicate that the corresponding nodes have common keys and can directly communicate with each other. Then, it can be found that if A wants to communicate with B, only the longest key-path $A \rightarrow C \rightarrow D \rightarrow B$ is available, since other paths $A \rightarrow B$, $A \rightarrow C \rightarrow B$ and $A \rightarrow D \rightarrow B$ are unavailable due to lack of common keys. In this case, several problems have to be considered: 1) the authentication of relay nodes; 2) resources consumption of all involved nodes (including relay nodes) caused by the messages forwarding, i.e., resource-constraint relay nodes have to consume their precious resources to complete the communication, such as energy, bandwidth and memory. Therefore, it is better to find another solution to achieve direct communication rather than communication with relay nodes.

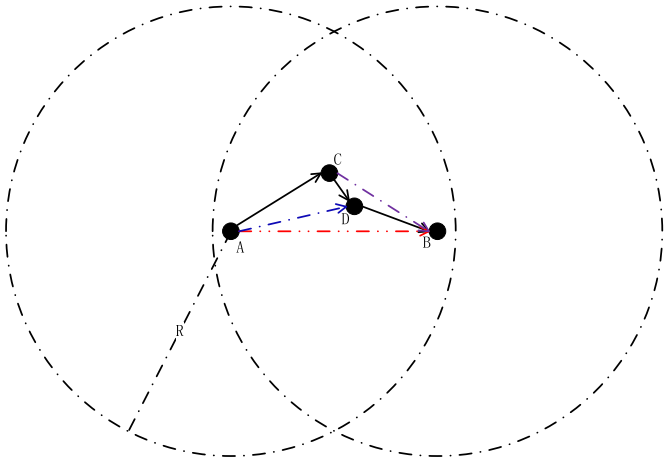

Fig. 1. Impact of key connectivity on communication.

Table 1
A summary of notation.

Symbol	Meaning
n	scale of network
k	size of key ring
m	number of the keys unknown to each node
$keyPool$	key pool of the network
$\|keyPool\|$	size of key pool
$H(\cdot)$	hash function
k_i	the i_{th} key of key pool
N_i	the i_{th} node in the network
kc_{ij}	communication key between nodes N_i and N_j
sk_i	the i_{th} sub-pool of key pool
$\|sk_i\|$	size of sk_i
$intersection_i$	intersection determined by sk_i

4. Preliminaries

To enhance the key connectivity, a novel key generation method based on system of equations is proposed. The proposed method utilizes the system of equations and its solutions to establish secret keys. Then, key management schemes can distribute these keys to nodes and efficiently protect the network. In contrast to normal keys, the keys generated by the proposed method are called the associated-keys. To clearly illustrated the proposed scheme, the notation used in the following sections are described in Table 1.

4.1. System of equations

An equation is an equality containing one or more variables and a system of equations is a collection of two or more equations with a same set of variables. As for any equation, the solution of an equation is the values of the variables for which the equation is true. Accordingly, a solution for a system of equations is an assignment of values to the variables such that all equations are simultaneously satisfied. Therefore, the solutions can be considered as the underlying associations among the involved equations, i.e., the implicit shared resources for the equations.

Assume that a system of equations consists of u equations $\varphi_i(x_1, x_2, \ldots, x_v) = 0$ $(1 \le i \le u)$ and each equation has v variables. The system of equations can be expressed as

$$\Phi^{(v)} = \begin{cases} \varphi_1(x_1, x_2, \ldots, x_v) = 0 \\ \vdots \\ \varphi_u(x_1, x_2, \ldots, x_v) = 0 \end{cases} \quad (1)$$

Where $\Phi^{(v)}$ indicates that the system of equations has v different variables. Then, the solution of $\Phi^{(v)}$ can be summarized as

$$S^{(v)} = \begin{cases} \emptyset \\ \{s(x_1, x_2, \ldots, x_v)^T\} \\ \{s_1^T, s_2^T, \ldots, s_t^T\} \end{cases} \quad (2)$$

Where $S^{(v)}$ denotes the collection of solutions and $s_i(x_{i1}, x_{i2}, \ldots, x_{iv})^T$ is the i_{th} solution of the system of equations. According to the theory of equations, if a system has a finite number of solutions in an algebraically closed extension L of l, this system is zero-dimensional. Further, in this work, we focus on the systems which have one and only one solution and use these systems to generate secret keys for the given network. Consequently, the unique solution can be used as a shard secret to enhance the association between nodes.

According to the requirement, if a system of equations $\Phi^{(v)}$ with u equations has one and only one solution $s(x_1, x_2, \ldots, x_v)^T$, this system is defined as an eligible system $ES(u, v)$, which can be used to implement the proposed key generation scheme.

The eligible system can be used to establish a key pool, if the involved equations in this system are used to generate secret keys. Taking $ES(u, v)$ for example, a key pool ($keyPool = \{\varphi_1, \varphi_2, \ldots, \varphi_u\}$) can be established for the network, where $\varphi_i(x_1, x_2, \ldots, x_v) = 0$ is the i_{th} key k_i. When each node randomly selects k equations from $ES(u, v)$ as its own k secret keys, this key pool can be used to protect a network as large as $\binom{|ES(u,v)|}{k}$, where $|ES(u, v)|$ is the number of equations. What's more, if k equations can figure out the unique solution s, neighbors can communicate with each other through the shared hidden key (the unique solution s) even though common keys (equations) do not exist.

According to the theory of equations, if the number of equations is smaller than the number of the variables, the system is either inconsistent or has infinitely many solutions. Therefore, the involved parameters u, v and k have to satisfy some constraints so that the system of equations can be used as a eligible stem and nodes can efficiently figure out the unique solution. Taking $ES(u, v)$ for example, the relationship between u and v is $u \geq v$, which is the sufficient and necessary condition for linear system and a necessary condition for other systems. In addition, the value of k is also important which determines whether a node can figure out the unique solution. Actually, if the value of k is not big enough (at least larger than the number of variables), nodes that randomly select k equations might calculate infinitely many solutions. Although the unique solution s is included in these solutions, it is prior unknown and nodes cannot efficiently find the unique solution to establish secure links. Therefore, the value of k must ensure that the unique solution can be figured out. Similarly, the relationship $k \geq v$ is the sufficient and necessary condition for linear system and a necessary condition for other systems. In contrast, the relationship between k and u is not determined. Obviously, nodes can randomly select k equations from one $ES(u, v)$, if $k \leq u$. In the case where $k > u$, the entire key pool can be established by multiple eligible systems. In Section 5.3, we further explain the implementation of key generation in detail.

Among various types of equations, polynomial equations and linear equations are two main families of equations. Next, we illustrate the proposed method with polynomial equations and linear equations, respectively. Meanwhile, some illustrations applying geometry interpretations are used to intuitively explain the method.

4.2. System of polynomial equations

Polynomial equation is an equation of the form P=0, where P is polynomial with coefficients in some field, often the field of the rational numbers. A polynomial equation is called multivariate polynomial equation if it involves several variables. A system of polynomial equations is a collection of polynomial equations, which can be expressed as

$$P^{(v)} = \begin{cases} p_1(x_1, x_2, \ldots, x_v) = 0 \\ \vdots \\ p_u(x_1, x_2, \ldots, x_v) = 0 \end{cases} \tag{3}$$

where $P^{(v)}$ is a system of u polynomial equations and each equation has v variables. Likewise, we consider the case that the system has a unique solution $s(x_1, x_2, \ldots, x_v)^T$.

Typically, each equation with v variables can determine a $v-$dimensional geometry, e.g., an equation $p(x, y, z) = 0$ defines a geometry in 3-dimensional space. When the system of equations has one and only one solution, the corresponding geometries have a unique intersection. Figs. 2 and 3 illustrate the geometries corresponding to various systems of polynomial equations, respectively.

In Fig. 2, four 3-dimensional geometries are depicted. These geometries are various spheres and can be determined by the follow system of polynomial equations

$$P^{(3)} = \begin{cases} p_1(x, y, z): x^2 + y^2 + z^2 - 1 = 0 \\ p_2(x, y, z): (x-3)^2 + y^2 + z^2 - 4 = 0 \\ p_3(x, y, z): (x-1)^2 + (y-1)^2 + z^2 - 1 = 0 \\ p_3(x, y, z): (x-1)^2 + (y+\frac{1}{2})^2 + z^2 - 1 = 0 \end{cases} \tag{4}$$

Obviously, these geometries have a unique intersection, i.e., $P^{(3)}$ has one and only one solution $s(x, y, z)^T$

$$s(x, y, z)^T = \begin{cases} x = 1 \\ y = 0 \\ z = 0 \end{cases}$$

Fig. 3 illustrates five 2-dimension geometries which are determined by a system of five polynomial equations. In this system of polynomial equations, each equation has 2 variables. The system of polynomial equations has the form

$$P^{(2)} = \begin{cases} p_1(x, y): (x-1)^2 + y^2 - 1 = 0 \\ p_2(x, y): (x-1)^2 + y - 1 = 0 \\ p_3(x, y): x^2 - y = 0 \\ p_4(x, y): (x-\frac{3}{2})^2 - \frac{y}{4} = 0 \\ p_5(x, y): 2x^2 - y - 1 = 0 \end{cases} \tag{5}$$

Correspondingly, the unique solution of $P^{(2)}$ is

$$s(x, y)^T = \begin{cases} x = 1 \\ y = 1 \end{cases}$$

Taking $P^{(2)}$ for example, a key pool ($keyPool = \{p_1, p_2, p_3, p_4, p_5\}$) can be established for protecting a network. The key pool has five different keys corresponding to the involved equations. As illustrated in Fig. 3, in some cases, the geometries have more than one intersections, such as the combination of p_1, p_2 and p_3. As a result, the communicating nodes have to take some time and resources to find which intersection is used by the other, which is not efficient. Therefore, each node is required to store four equations to calculate the unique solution.

4.3. System of linear equations

In contrast to polynomial equation, linear equation is the equation in which all involved variables are only multiplied by numbers. System of linear equations, also called linear system, consists of two or more linear equations, which can be described as

$$F^{(v)} = \begin{cases} f_1(x_1, x_2, \ldots, x_v): a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,v}x_v = b_1 \\ \vdots \\ f_u(x_1, x_2, \ldots, x_v): a_{u,1}x_1 + a_{u,2}x_2 + \cdots + a_{u,v}x_n = b_u \end{cases} \tag{6}$$

The coefficient matrix A can be found as

$$A_{u\times v} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,v} \\ \vdots & \ddots & \vdots \\ a_{u,1} & \cdots & a_{u,v} \end{bmatrix}$$

According to the properties of the coefficient matrix $A_{u\times v}$, the solutions of $F^{(v)}$ can be determined as follows:

$$S = \begin{cases} \emptyset, \ if \ R(A) \neq R(A, b) \\ \{s(x_1, x_2, \ldots, x_v)^T\}, \ if \ R(A) = R(A, b) = v \\ \{s_G^T + s_P^T\}, \ if \ R(A) = R(A, b) < v \end{cases} \tag{7}$$

Where $R(A)$ is the rank of $A_{u\times v}$ and $R(A, b)$ denotes the rank of the augmented matrix. $s_P{}^T$ is the specific solution, while $s_G{}^T$ is the general solution of the corresponding homogeneous systems $F^{(v)} = 0$.

As mentioned above, our primary concern is the case where $F^{(v)}$ has one and only one solution $s(x_1, x_2, \ldots, x_v)^T$. Then, $F^{(v)}$ is applied to establish a key pool ($keyPool = \{f_1, f_2, \ldots, f_u\}$). Each node randomly selects k keys (equations) to store and uses these keys to connect with

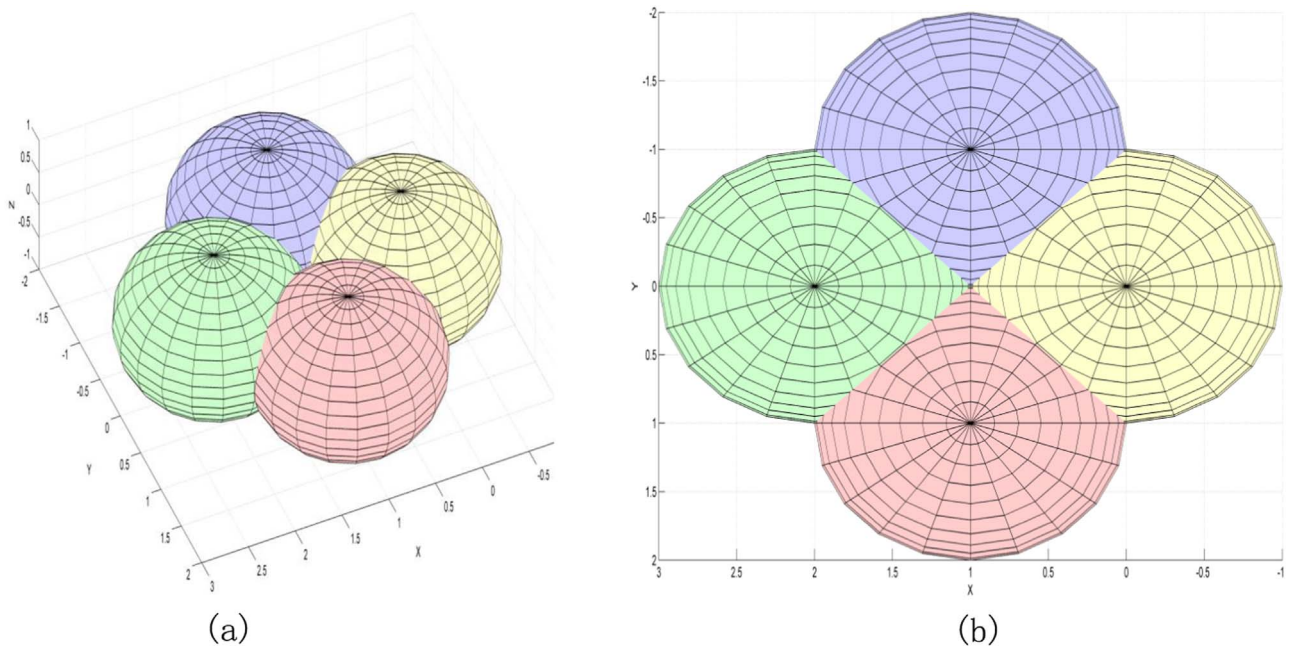

Fig. 2. 3-dimensional geometries corresponding to a system of polynomial equations. In this case, these geometries has one and only one intersection.

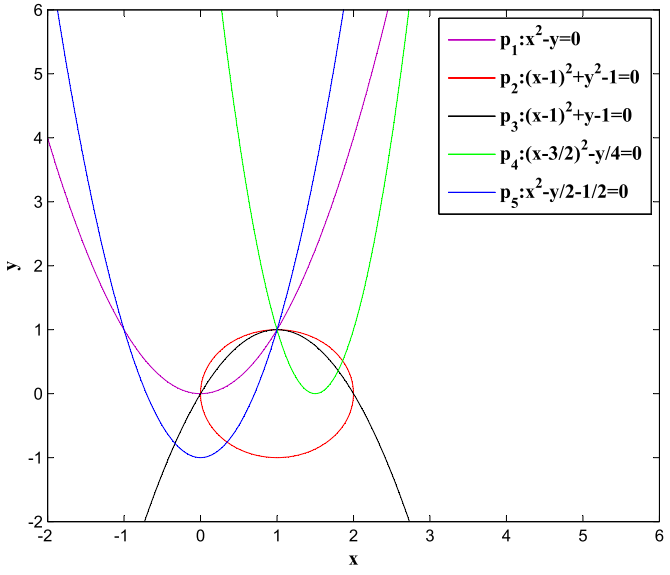

Fig. 3. 2-dimensional geometries corresponding to a system of polynomial equations. In this case, these geometries has one and only one intersection.

other nodes. Consequently, the key pool can be used to protect a network of $\binom{u}{k}$ nodes. To ensure the nodes in the network can figure out the unique solution, the following requirements must be satisfied

$$\begin{cases} k \geq v & (8) \\ u > k & (9) \end{cases}$$

The first condition ensures that each node can calculate the unique solution. The second requirement guarantees that sufficient available combinations of keys can be established for the given network. In addition, to ensure that these keys are different from each other, $F^{(v)}$ must ensure that none of the equations can be derived algebraically from any other equations. Figs. 4 and 5 show some examples with 2-dimensional and 3-dimensional geometries, respectively.

In Fig. 4, five 3-dimensional geometries are depicted. These geometries are determined by the following system of linear equations

$$F^{(3)} = \begin{cases} f_1(x, y, z): x = 0 \\ f_2(x, y, z): y = 0 \\ f_3(x, y, z): z = 0 \\ f_4(x, y, z): x + y = 0 \\ f_5(x, y, z): x - y = 0 \end{cases} \quad (10)$$

The solution of $F^{(3)}$ is

$$s(x, y, z)^T = \begin{cases} x = 0 \\ y = 0 \\ z = 0 \end{cases}$$

Fig. 5 shows several 2-dimensional lines determined by the following linear system

$$F^{(2)} = \begin{cases} f_1(x, y): x - y = 0 \\ f_2(x, y): x + y - 2 = 0 \\ f_3(x, y): y - 1 = 0 \\ f_4(x, y): 2x - y - 1 = 0 \\ f_5(x, y): 2x + y - 3 = 0 \end{cases} \quad (11)$$

The solution of $F^{(2)}$ is

$$s(x, y)^T = \begin{cases} x = 1 \\ y = 1 \end{cases}$$

Taking $F^{(2)}$ for example, $F^{(2)}$ can be applied to establish a key pool ($keyPool = \{f_1, f_2, f_3, f_4, f_5\}$). Obviously, it can be found that k=2 can ensure the unique solution can be calculated. Therefore, if each node randomly selects $k \geq 2$ equations as its keys, the unique solution (intersection) can be figured out to enhance the association among nodes. As a result, a network of $\binom{5}{k}$ nodes can be protected by the generated keys.

4.4. Evaluation of associated-keys

In this section, we evaluate the associated-key through its impacts on key connectivity, security, computation complexity and storage.

Key connectivity In this work, key connectivity is defined as the direct secure connectivity coverage P_{dc}, which is calculated as the probability that a pair of neighbor nodes are able to establish a direct

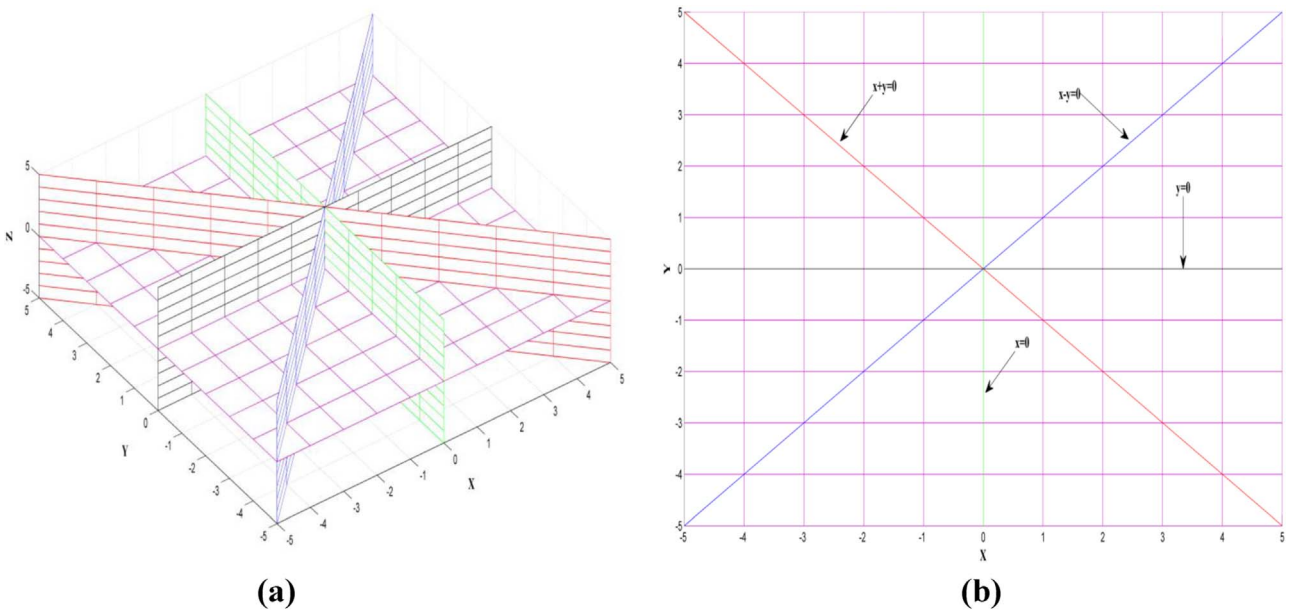

Fig. 4. 3-dimensional geometries corresponding to a system of linear equations.

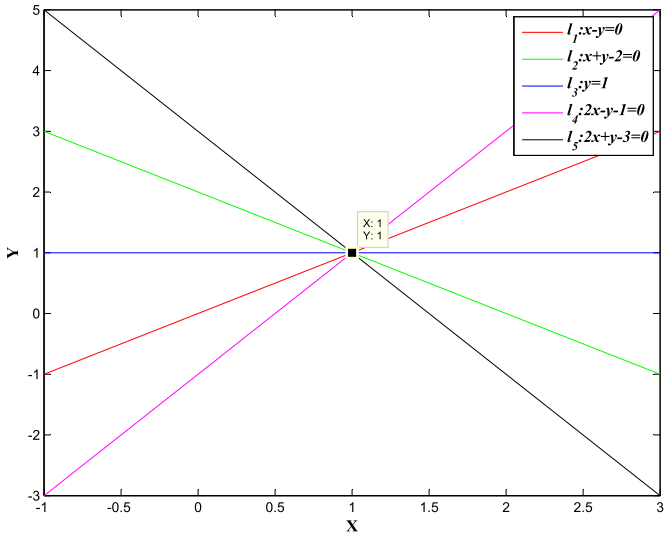

Fig. 5. 2-dimensional geometries corresponding to a system of linear equations.

secure link. For the key management scheme with normal keys, the probability P_{dc} equals the probability that neighbor nodes have common keys P_{ck}. In contrast, for the key management scheme with the associated-keys, the probability P_{dc} is the sum of the probability that neighbor nodes have common keys P_{ck} and the probability P_{ci}, which represents neighbor nodes have shared hidden keys established by the unique solution P_{ci} when they do not have common keys. That is

$$P_{dc} = P_{ck} + P_{ci} \tag{12}$$

Obviously, key connectivity is different when the associated-keys are applied into different key management schemes. The impact of the associated-keys on key connectivity is affected by the following factors:

(1) P_{ck} of different key management schemes is different;
(2) P_{ci} is also different in different key management schemes even though these schemes apply the same eligible systems, since the mapping of keys to nodes during key assignment is different. For example, if a key pool *keyPool* is used in Eschenauer and Gligor (2002) or Camtepe and Yener (2007), the former allows each node can randomly select k keys, while the latter requires that keys are selected according to the principle of SBIBD.

Typically, to balance between key connectivity and security, the entire key pool is formed by multiple sub-pools, i.e., several eligible systems are used to establish the entire key pool. On the one hand, when the number of involved eligible systems is increased, the resilience is enhanced since the impact of each eligible system is reduced; on the other hand, the association is reduced when the number of eligible systems is increased. In the simplest case, the entire key pool is established by only one elidible system. Then, the direct communication can be achieved by all neighbors who can calculate the unique solution. However, when a node is captured and the unique solution is thus exposed, all the other nodes cannot use this solution any more. Therefore, the number of applied eligible systems might be determined according to different requirements.

Security Firstly, the secrecy of associated-keys is analyzed. According to the theory of equations, only the unique solution and k equations cannot extrapolate other used equations in the elidible system, since this system is prior unknown and the appropriate equations that have the same solution are innumerable. Then, it is impossible to make sure which equations are applied to establish other secret keys. That is, the secrecy of associated-keys is same as normal keys.

Secondly, we discuss the impact of associated-keys on the resilience against node capture. Typically, the resilience is calculated as the fraction of uncompromised secure links when x nodes are captured. The key connectivity of key management scheme significantly affects the resilience. As the aforementioned analyses, the key connectivity can be improved when associated-keys instead of normal keys are used for key management. Consequently, the number of compromised links is increased when x nodes are captured. However, the number of total secure links is also increased. Therefore, although the fraction of compromised links might be increased, the number of available links is still larger than it in the corresponding scheme with normal keys since other uncompromised solutions can still establish secure links between uncompromised nodes. Moreover, as a performance metric of key management, the resilience is determined by the used key and key connectivity which might be different in different key management schemes. Therefore, the resilience cannot be calculated unless associated-keys are used in a key management scheme.

Computation complexity According to key analysis and key assignment, the construction of appropriate eligible systems is implemented by the key server before node deployment. For sensor nodes, the additional computation overhead caused by the associated-keys is produced in the case where the hidden key established by the unique solution is used to establish secure link between nodes. The resulting computation overhead is the cost of solving system of equations to find the unique solution. Comparing with system of polynomial equations, solving linear system requires fewer calculations. Several methods, such as Gaussian elimination, Cramer's rule and Matrix solution, were proposed to solve linear system. The computation complexity of Cramer's rule is $O(v^3)$, where v is the number of variables (Habgood and Arel, 2012).

Storage When the applied eligible system has more variables, more equations are required to calculate the unique solution. In addition, comparing with linear system, system of polynomial equations with same variables might need more equations to determine the unique solution. That is, the number of keys required by each node is determined by the number of variables and the form of the involved equations, which significantly affects the application of associated-keys. In Section 5, we use the linear system of two variables to implement the proposed method and prove that such linear systems are sufficient to generate keys for large scale of networks.

5. Implementation by linear system of two variables

5.1. Key generation by linear system of two variables

Comparing with system of polynomial equations, the construction and calculation of linear system are simple and cost less resources of nodes. Therefore, linear system is more appropriate to implement the proposed method. Actually, linear systems of two variables are sufficient to generate secret keys for efficiently protecting a very large scale of network. In this section, to describe the implementation of the proposed method in detail, linear system of two variables is applied to generate secret keys. Likewise, geometry interpretation is used to intuitively illustrate the method.

In this case, a linear system can be expressed as

$$F^{(2)} = \begin{cases} f_1(x, y)\colon a_{1,1}x + a_{1,2}y + b_1 = 0 \\ \vdots \\ f_u(x, y)\colon a_{u,1}x + a_{u,2}y + b_u = 0 \end{cases} \tag{13}$$

The solution set of $F^{(2)}$ is

$$S^{(2)} = \begin{cases} \varnothing \\ \{s(x, y)^T\} \end{cases}$$

Correspondingly, the geometries determined by different linear equations are various straight lines. According to the theory of plane geometry, a line can be determined by two different points. For example, the equation of the line passing through two different points $P(x_p, y_p)$ and $Q(x_q, y_q)$ can be written as

$$(y - y_p)(x_q - x_p) = (y_q - y_p)(x - x_p) \tag{14}$$

If $x_p \neq x_q$, this equation is rewritten as

$$y = (x - x_p)\frac{y_q - y_p}{x_q - x_p} + y_p \tag{15}$$

where $\frac{y_q - y_p}{x_q - x_p}$ is the slope of the corresponding line. As a result, each equation can be figured out by two distinct points on the corresponding line. Therefore, two different points on each line instead of the corresponding equation are applied to realize the proposed method, i.e., each secret key is established by two different points on each line, which is determined by the equation in the linear system.

In order to ensure that each node can figure out the unique solution

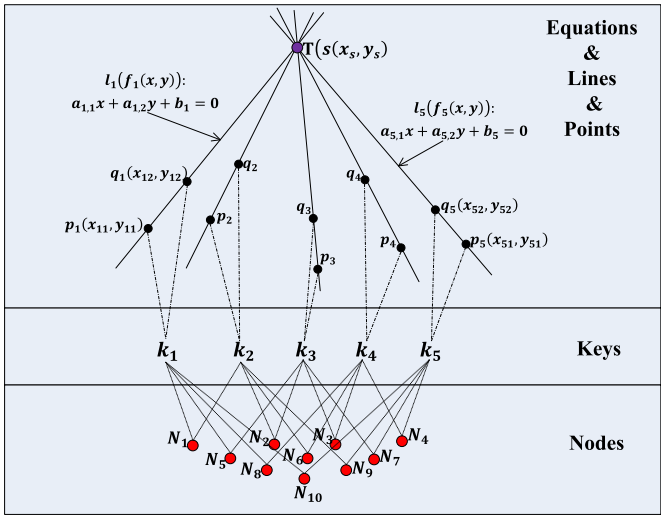

Fig. 6. The relationship among point, key and node.

$s(x, y)^T$, each node needs to store at least two keys, since two different lines have an intersection when they are not parallel with each other. Fig. 6 shows an example to interpret the proposed scheme. In Fig. 6, there are five different lines l_i $(1 \leq i \leq 5)$, which are determined by a linear system of five linear equations. These lines have an intersection T, which means that the corresponding linear system has one and only one solution $s(x_s, y_s)^T$. As mentioned above, we can determine l_i with the following equation:

$$l_i\colon \frac{x - x_{i1}}{x_{i2} - x_{i1}} = \frac{y - y_{i1}}{y_{i2} - y_{i1}},\ x_{i2} \neq x_{i1},\ y_{i2} \neq y_{i1},\ 1 \leq i \leq 5 \tag{16}$$

Then, a key pool which has five keys is established for the network and each key is generated by two different points on the line corresponding to each equation. As illustrated, each node randomly selects two keys to store and a network of ten nodes can be protected by these keys. Fig. 6 illustrates a simplest case where the entire key pool is established by one eligible system. Typically, multiple eligible systems rather than single eligible system are used for key generation and the implementations of such key generation schemes are similar.

5.2. Integration with key management

The proposed scheme is only a method for achieving key generation which is a component of key management. Therefore, the proposed scheme has to be integrated into key management to provide comprehensive protection for WSNs. Comprehensively considering all metrics, such as key connectivity and security, multiple eligible systems are typical used to implement the proposed scheme. In this case, several linear systems of two variables are used to establish various sub-pools to form the entire key pool.

The implementation of key management can be described as follows:

1) During key analysis, the size of key pool $|keyPool|$ and the number of keys assigned for each node are determined according to the scale of network;
2) During key assignment, the mapping of keys to nodes is determined;
3) During key generation, the proposed method is carried out to generate secret keys:
   (i) The collection of intersection $\{intersection_i,\ 1 \leq i \leq t\}$ is determined and the entire key pool is divided into t sub-pools $\{sk_i,\ 1 \leq i \leq t\}$. Each intersection corresponds to a sub-pool. The size of i_{th} $(1 \leq i \leq t)$ sub-pool $|sk_i|$ is

$$|sk_i| = \begin{cases} \frac{|keyPool|}{t}, \ if \ |keyPool| \bmod t = 0 \\ \left\lfloor \frac{|keyPool|}{t} \right\rfloor + \left(1 - \left\lfloor \frac{t-i}{|keyPool| \bmod t} \right\rfloor\right), \ otherwise \end{cases} \tag{17}$$

(ii) According to the size of i_{th} $(1 \le i \le t)$ sub-pool, $|sk_i|$ lines that pass through $intersection_i$ are determined;
(iii) Two distinct points from each line, P_{ij} and Q_{ij}, are selected and the secret key is established as

$$k_{ij} = H(P_{ij} \parallel Q_{ij}),\ 1 \le i \le t,\ 1 \le j \le |sk_i| \tag{18}$$

4) During key distribution, keys (actually the corresponding point pairs) are distributed to nodes according to the mapping of keys to nodes

In formula (18), $P_{ij} \parallel Q_{ij}$ denotes that two elements are linked to each other. As a result, associated-keys can be used to implement key management and such key management scheme can provide efficiently protection for the network.

5.3. An instance of key management with associated-keys

In this section, a key management scheme that combines EBS with associated-keys is used as an instance to illustrate the protection mechanism for the network. The entire process consists of three phases: node pre-deployment, shared-key discovery/path-key establishment and key redistribution.

Node pre-deployment phase In the node pre-deployment phase, the appropriate EBS framework is established according to the scale of the network. Then, according to the proposed key generation method, associated-keys are established and distributed to nodes before deployment. The details of the implementation are described as follows:

(1) During key analysis, according to the scale of network n, the size of key pool $|keyPool|$ and key ring $|keyRing|$ are determined as $\binom{|keyPool|}{|keyRing|} = \binom{k+m}{k} \ge n$;
(2) During key assignment, the mapping of keys to nodes is determined;
(3) During key generation, the proposed method is implemented to generate secret keys as illustrated in Section 5.2. Moreover, the ID of each key is established;
(4) During key distribution, keys (actually the corresponding point pairs) and the corresponding IDs are distributed to nodes according to the mapping of keys to nodes.

In this case, the ID of each key consists of two factors: 1) the index of sub-pool; 2) the index of key in this sub-pool. For example, k_{25} is the 5_{th} key in the 2_{nd} sub-pool. As a result, nodes can make sure which intersections can be figured out according to the IDs of keys, e.g. k_{21} and k_{25} can calculate $intersection_2$. What's more, the adversary who listens to the communication cannot infer the intersection from these IDs. Therefore, secure links can be established by common keys or these intersections.

Shared-key discovery/path-key establishment In the shared-key discovery/path-key establishment phase, neighbors find the shared keys between each other to complete the communication. By broadcasting the IDs of their keys, nodes can know who has common keys with themselves. If two neighbors have common keys, they can directly establish secure communication through these common keys. Otherwise, they need to check the IDs of keys received from the other to make sure if they can figure out same intersections. If they have more than two keys from the same sub-pool, the intersection of this sub-pool can be calculated. Then, they can also establish secure communication with the key established by the intersection. For example, both nodes N_1 and N_2 can compute l intersections $intersection_i$ $(1 \le i \le l)$. Then, they can establish a shared key $kc_{12} = (intersection_1 \parallel intersection_2 \parallel \cdots \parallel intersection_l)$ and use this key to protect their communication.

When the communicating parties do not have common keys and common intersections cannot be calculated, they have to implement path-key establishment. By discovering the relay nodes that could directly communicate with them, they can establish a secure path to complete the communication. When the path-key establishment is failure, the communication cannot be achieved, which means that the network is not connected.

Key redistribution Key redistribution is triggered by two events: 1) node addition; 2) node is captured.

To replace the exhausted nodes or enhance the coverage of the network, new nodes have to be deployed after initial deployment.

1) New nodes are deployed to replace the exhausted nodes. In this case, the number of nodes in the network is not changed. Accordingly, new nodes can use same keys as the corresponding exhausted nodes.
2) New nodes are deployed to enhance the coverage of the network. In this case, the number of nodes in the network is increased. For each new node, according to its expected location and keys of its probable neighbors, keys of the involved sub-pools are used to form the key rings which are appropriate for the new node. If some of these key rings are still available, the new node randomly selects one as its key ring. Otherwise, the following method can be used to provide available key rings: for each involved sub-pool of its probable neighbors, increase a new key in this sub-pool, i.e., increase a line (equation) which passes through the corresponding intersection. Consequently, the appropriate key ring can be established.

If a node in the network is captured, keys stored in this node are exposed. Key redistribution has to be implemented to update the key system and evict the compromised node. In this case, according to the number of intersections that the compromised node could compute, the content of rekeying messages is different.

1) If the compromised node cannot compute the valid intersection, only the exposed keys need to be updated and the nodes which have the exposed keys need to update their keys. In this case, new keys could be simply established by different point pair on the same line. For example, assume that the node N_2 is compromised, and the keys k_{13} and k_{25} are exposed. k_{13} is established by the point pair p_{13} and q_{13} on the line l_{13}. Then, the new key k_{13}' can be established by another point pair p_{13}' and q_{13}' on the line l_{13}. As a result, the exposed key k_{13} can be replaced with k_{13}' by the EBS key update mechanism.
2) If the compromised node can compute several intersections, two types of keys need to be updated: 1) the keys of the compromised node; 2) all sub-pools corresponding to the exposed intersections. In this case, the proposed method has to be implemented by the server once again. New intersections, lines and point pairs are selected to establish new keys according to the number of exposed keys and the size of all involved sub-pools. Then, the generated new keys can be applied to update the key system by the EBS key update mechanism.

5.4. Theoretical analysis

Key connectivity For EBS(n,*k*,m), the probability that any two nodes has common keys P_{ck} is

$$P_{ck} = \begin{cases} 1 & if\ k > m \\ 1 - \prod_{i=0}^{k-1} \frac{m-i}{k+m-i} & if\ k \le m \end{cases} \tag{19}$$

In contrast, for the illustrated scheme, P_{ci} can be calculated as

$$P_{ci} = \begin{cases} \frac{\sum_{i=2}^{k}\left[\binom{u}{i}\binom{k+m-u}{k-i} \times \sum_{j=2}^{k}\binom{u-i}{j}\binom{k+m-u-(k-i)}{k-j}\right]}{\left[\binom{k+m}{k}\right]^2 - 1} & if\ k \le \frac{u}{2} \\ \frac{\sum_{i=2}^{\frac{u}{2}}\left[\binom{u}{i}\binom{k+m-u}{k-i} \times \sum_{j=2}^{\frac{u}{2}}\binom{u-i}{j}\binom{k+m-u-(k-i)}{k-j}\right]}{\left[\binom{k+m}{k}\right]^2 - 1} & else \end{cases} \tag{20}$$

In this case, we assume that each eligible system has u equations and two equations are able to calculate the corresponding solution, which can be easily implemented during key generation. According to the analyses in Section 4.4, key connectivity can be calculated as $P_{dc} = P_{ck} + P_{ci}$. As a result, we can calculate P_{dc} of the illustrated scheme. In the simplest case, the entire key pool is established by only one linear system and only one intersection is used for the illustrated scheme. Obviously, the probability P_{dc} is 1.

Security The secrecy analysis in Section 4.4 demonstrates that the secrecy of associated-keys is same as normal keys. For the illustrated scheme, we mainly analyze the resilience against node capture. In this case, the fraction of compromised keys and intersections when x nodes are captured is analyzed. Let P_{key} denote the fraction of compromised keys, while P_t represent the fraction of compromised intersections.

According to Chan et al. (2003), the fraction of compromised keys when a node is compromised is $\frac{k}{k+m}$. Then, we calculate P_{key} as

$$P_{key} = 1 - \left(1 - \frac{k}{k+m}\right)^x \tag{21}$$

In contrast, the fraction of compromised intersections when a node is compromised is

$$p = \begin{cases} \frac{\sum_{j=2}^{k}\binom{u}{j}\binom{k+m-u}{k-j}}{\binom{k+m}{k}} & if\ k \le u \\ \frac{\sum_{j=2}^{u}\binom{u}{j}\binom{k+m-u}{k-j}}{\binom{k+m}{k}} & else \end{cases} \tag{22}$$

Then, we can calculate P_t as

$$P_t = 1 - (1-p)^x \tag{23}$$

Obviously, we can find that the links established by the compromised intersections are unavailable. However, other intersections are still available to enhance the association between uncompromised nodes. Moreover, the adversary cannot use the compromised keys and intersections to extrapolate other keys and intersections. That is, although some keys and intersections are compromised, key connectivity of the illustrated scheme is still better than the corresponding scheme with normal keys.

Scalability The scalability indicates the maximum scale of the network that can be supported by the generated key ring. The mapping of keys to nodes during key assignment determines the supported scale of the network. Therefore, the scalability of key management is not changed even though different types of keys are used. Therefore, the scalability of the illustrated scheme is $\binom{k+m}{k}$.

Computation complexity In this scheme, the computation of unique solution is simplified to calculate intersections of different lines (linear equations of two variables), where only some basic arithmetic operations are implemented. Therefore, the computation overhead is small and thus the key management scheme with associated-keys can efficiently protect the network.

Storage In this scheme, each node randomly selects k keys (equations) to store. Specifically, linear systems of two variables are used to generate keys, i.e., each key is established by two points on the corresponding line. As a result, each node stores $2k$ different points.

6. Performance evaluation

In order to evaluate the performance of the proposed method, several simulations are conducted. In this work, we assume 200 nodes are randomly deployed in a 300 m×300 m area. The communication radius of each node is 60 m. For different simulations, the EBS (Eltoweissy et al., 2004) and Unital scheme (Bechkit et al., 2013) are implemented to the given network, respectively. To differentiate from the types of keys, key management schemes that apply associated-keys are called Associated-EBS and Associated-Unital. In contrast, the schemes applying normal keys are called Normal-EBS and Normal-Unital. Specifically, the Associated-EBS (200, 3, 9), Associated-EBS (200, 4, 6), Associated-Unital (200, 5, 65) and the corresponding Normal-schemes are implemented to the network, respectively. In addition, the key pool is divided into t=2 or t=3 sub-pools in the simulations where EBS-based schemes are used. In contrast, the key pool is divided into 2, 3, 4 and 5 sub-pools in the simulations with Unital-based schemes. The Unital (200, 5, 65) scheme is the Steiner 2-design with parameters $S(2, q+1, q^3+1)$ and q=4. That is, each node stores 5 keys and the entire key pool has 65 keys. The implementation of the Unital scheme is proposed by Bechkit et al. in Bechkit et al. (2013).

Fig. 7 shows the deployment of nodes in the network. As mentioned above, nodes are randomly deployed into the network. Consequently, the distribution of nodes is uneven. Moreover, all nodes in the network are assumed to have same capabilities, such as computation capability and storage. That is, we assume that all key management schemes are applied into homogeneous WSNs.

Fig. 8 presents the network connectivity when key management is not applied. The lines in the figure indicate that the corresponding nodes are in the communication range of each other, i.e. the connected nodes are neighbors of each other. In this case, due to the fact that key management scheme is not applied to the network, whether two nodes can direct communicate with each other is determined by the communication radius of nodes and the distance between nodes. As illustrated, the communication capability ensures that the network is connected and there is no isolated node. Consequently, the following analyses are focused on the impacts caused by different key management schemes.

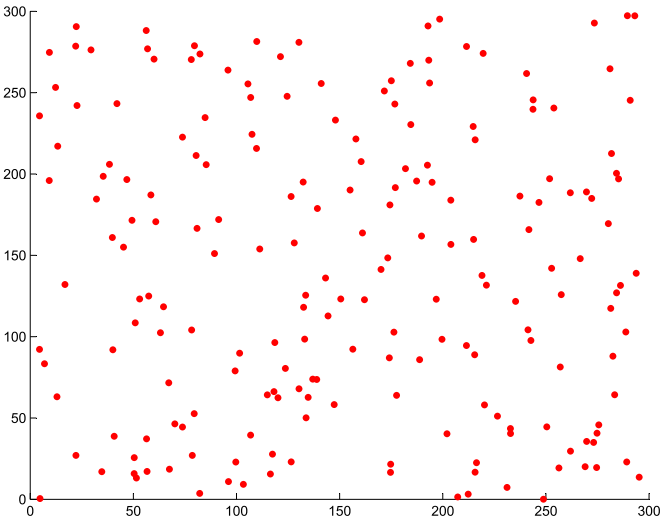

Fig. 7. Node deployment.

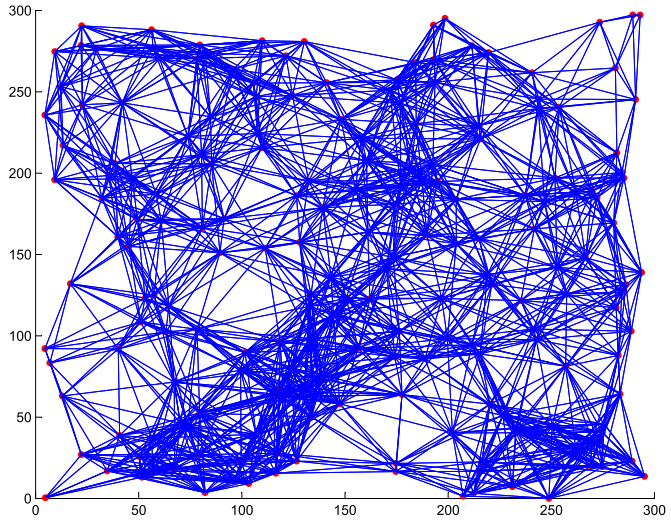

Fig. 8. The network without key management.

6.1. Key connectivity

To analyze the key connectivity of different schemes, several aspects are used, such as the number of neighbors of each node and the average length of key-path.

In Fig. 9, the cases where various EBS-based schemes are applied to the network are illustrated in (a)–(h), respectively. The meaning of blue lines is same as Fig. 8. The green lines indicate that the corresponding neighboring nodes have common keys. In contrast, two nodes linked by red line can figure out same intersections to establish shared hidden keys and implement direct communication. When two neighbors have common keys or shared hidden keys, the blue lines are covered by green lines and red lines, respectively. Then, the number of remaining blue lines reports the number of unavailable links caused by the implementation of key management, which intuitively shows the impact of key management. As illustrated, it can be found that.

1) According to the comparisons of (a) and (c) as well as (b) and (d), the key connectivity is better when more keys are distributed to nodes;
2) According to the comparison of each Normal-scheme and the corresponding Associated-scheme, the shared hidden key established by intersections can be applied to enhance key connectivity;
3) According to the comparisons of (e) and (f) as well as (g) and (h), when more intersections are used in the network, the improvement of key connectivity caused by hidden keys is reduced.

In summary, the associated-key can be applied to improve the key connectivity of key management.

Similarly, Fig. 10 shows the cases where various Unital-based schemes are applied for the network, respectively. During different simulations, the key pool is divided into 2, 3, 4 and 5 sub-pools to establish associated-keys. As expected, the Associated-Unital schemes perform better key connectivity than the corresponding Normal-Unital schemes. The best case is the network protected by the Associated-Unital (200, 5, 65) & t=2 scheme, where almost all physical links are available. In addition, different from Fig. 9, each node stores q=5 keys in all cases and the entire key pool totally has 65 keys. That is, the Unital-based schemes require more keys than EBS-based schemes to protect the same network. As a result, key connectivity is reduced when Normal-Unital schemes instead of Normal-EBS schemes are used. However, the improvement of key connectivity caused by associated-keys is more significant in Unital-based schemes, since a larger probability of calculating shared hidden keys can be achieved when each node has more keys.

Number of neighbors of each node Fig. 11 illustrates the number of neighbors of each node when different EBS-based schemes are applied to the network. In Fig. 11-(a), it can be found that the number of each node's neighbors in each Associated-EBS case is larger than it in the corresponding Normal-EBS case and this number is increased when each node stores more keys. Fig. 11-(b) indicates that the number of neighbors is decreased while the key pool is divided into more sub-pools. The results reflect the fact that the probability of calculating same intersections is reduced as the number of intersections increases.

Similarly, Fig. 12 presents the number of each node's neighbors in the cases where different Unital-based schemes are implemented. Fig. 12-(a) demonstrates the same results as Fig. 11-(a), i.e., the

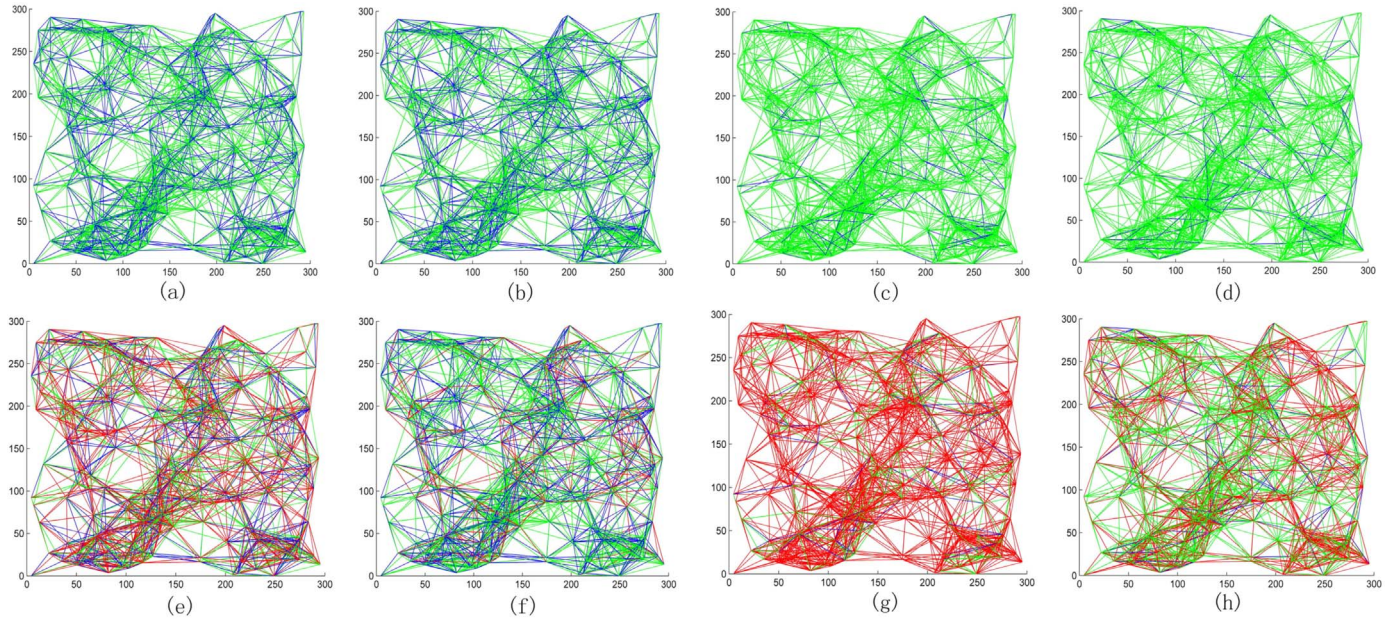

Fig. 9. Key connectivity of different key management schemes. In this case, key management schemes applied in (a)-(h) are {Normal-EBS(200, 3, 9) & t=2; Normal-EBS(200, 3, 9) & t=3; Normal-EBS(200, 4, 6) & t=2; Normal-EBS(200, 4, 6) & t=3; Associated-EBS(200, 3, 9) & t=2; Associated-EBS(200, 3, 9) & t=3; Associated-EBS(200, 4, 6) & t=2; Associated-EBS(200, 4, 6) & t=3}, where t is the number of the used linear systems. Additionally, keys used for key management are normal keys or associated-keys.

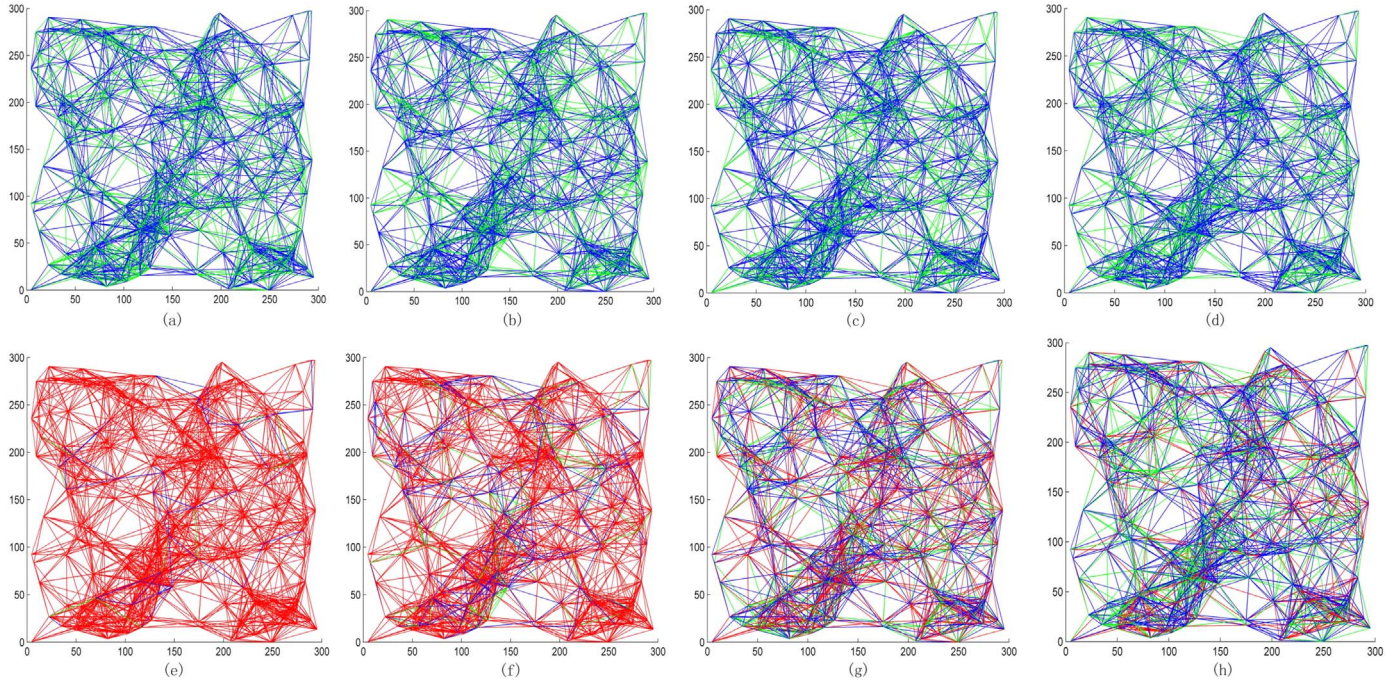

Fig. 10. Key connectivity of different key management schemes. In this case, key management schemes applied in (a)-(h) are {Normal-Unital(200, 5, 65) & t=2; Normal-Unital(200, 5, 65) & t=3; Normal-Unital(200, 5, 65) & t=4; Normal-Unital(200, 5, 65) & t=5; Associated-Unital(200, 5, 65) & t=2; Associated-Unital(200, 5, 65) & t=3; Associated-Unital(200, 5, 65) & t=4; Associated-Unital(200, 5, 65) & t=5}, where t is the number of the used linear systems. Additionally, keys used for key management are normal keys or associated-keys.

number of each node's neighbor in each Associated-Unital scheme is larger than it in the corresponding Normal-Unital scheme. Moreover, when the network is protected by the Associated-Unital (200, 5, 65) & t=2 scheme, neighbors of majority nodes are equal to the largest value which is the number of neighbors of each node without key management.

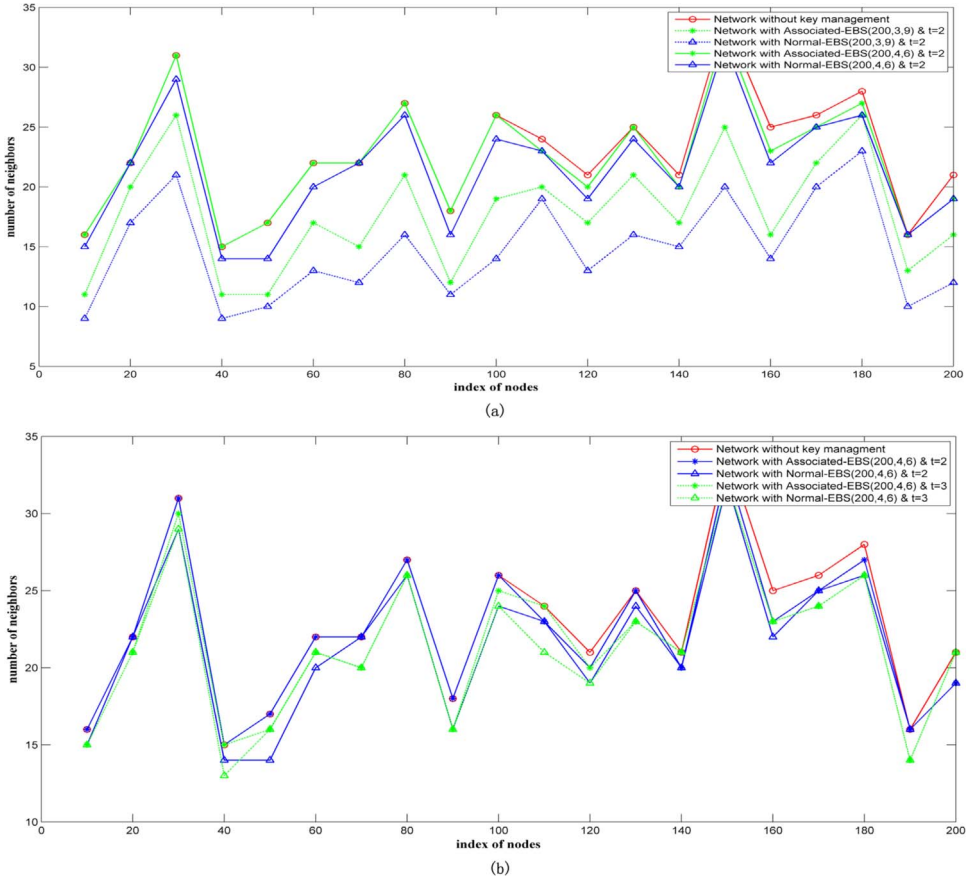

Fig. 11. Number of neighbors of each node in different EBS-based cases.

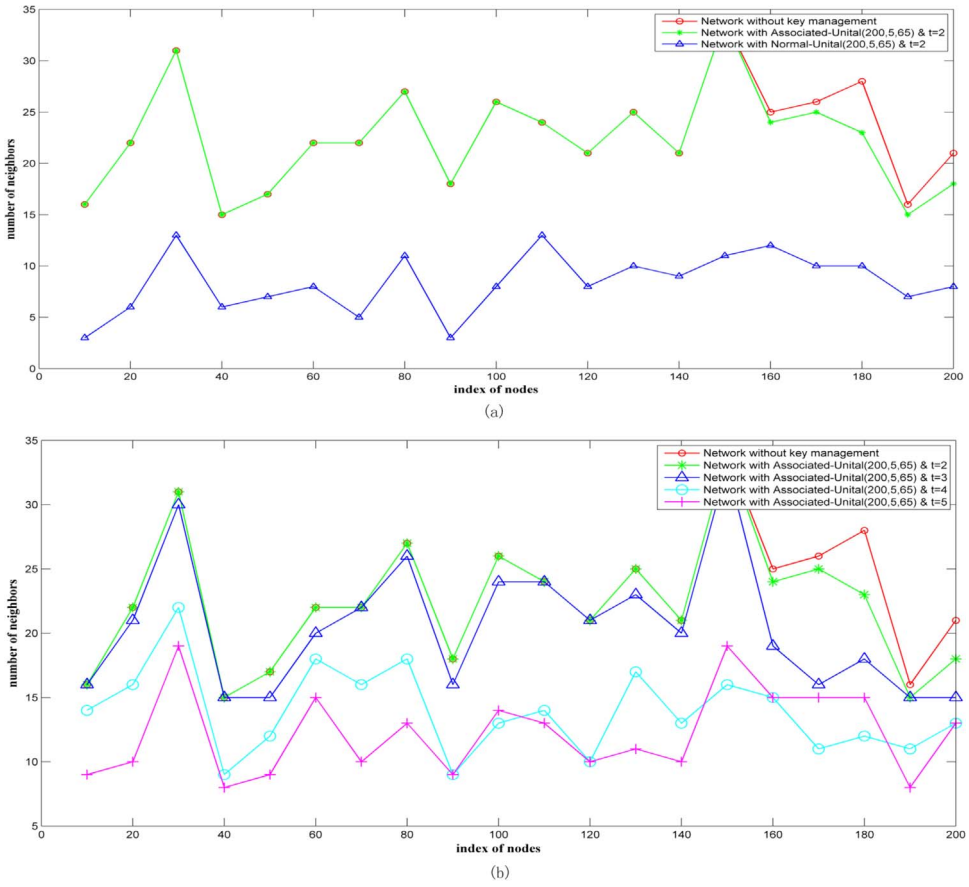

Fig. 12. Number of neighbors of each node in different Unital-based cases.

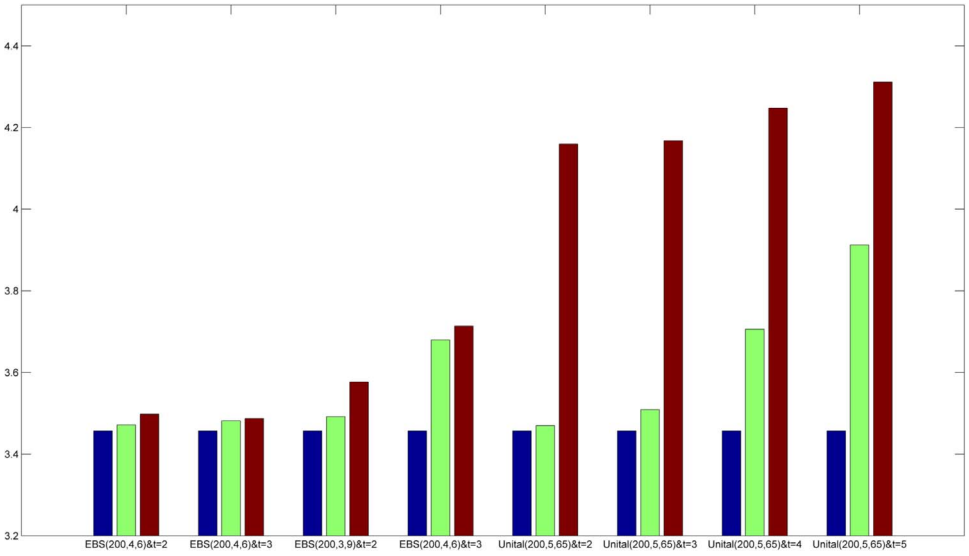

Fig. 13. The average length of key-paths in different cases.

Average length of key-path In Fig. 13, the average length of key-paths (ALKPs) in different cases are illustrated. The ALKP is a robust measure of network topology, which is defined as the average number of steps along the shortest paths for all possible pairs of network nodes. The blue bar indicates the ALKP when key management is not applied, which is the smallest in all cases. In contrast, the green bars and red bars reflect ALKPs of cases where different Associated-schemes and Normal-schemes are applied, respectively. It can be found that, when various t=2 schemes are applied, ALKPs are very close to the smallest value. In addition, the ALKP of each Associated-scheme is smaller than the corresponding Normal-scheme. The results also demonstrate that key connectivity is improved when different Associated-schemes are used to protect the network.

6.2. *Resilience against node capture*

Fig. 14 illustrates the resilience against node capture of different key management schemes. In this case, we analyze the resilience when

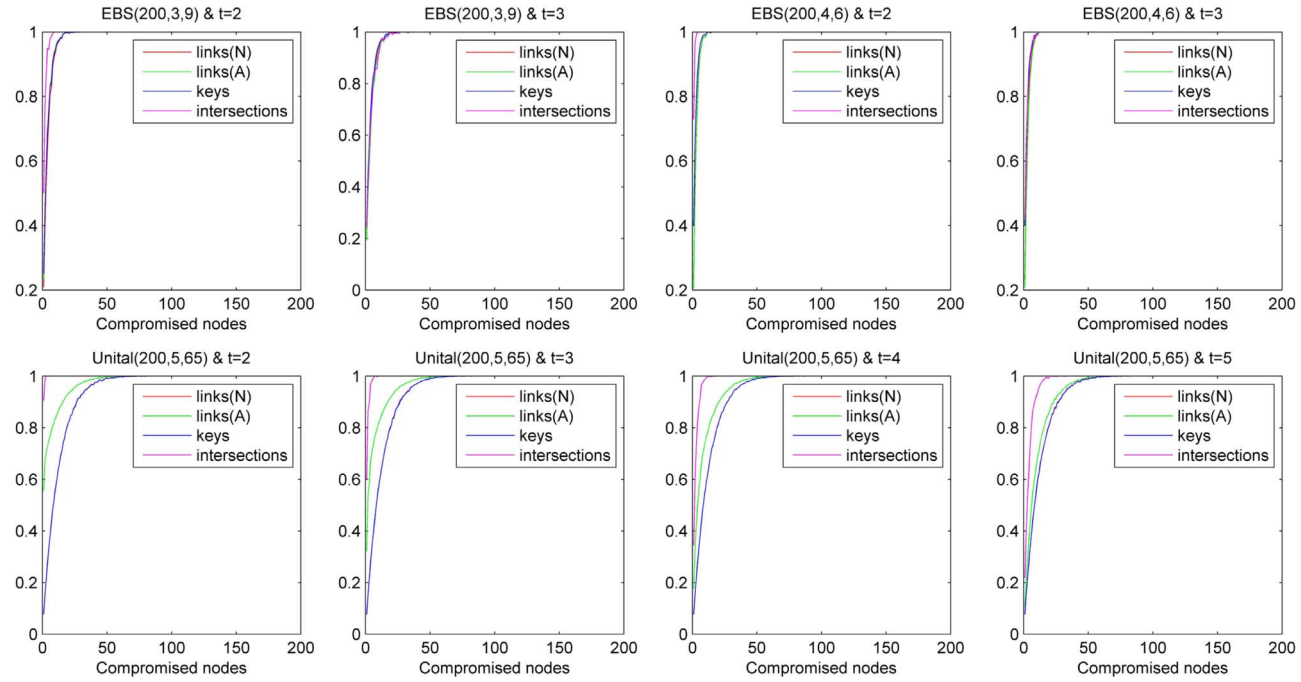

Fig. 14. Resilience of different schemes. In this figure, links(N) is the fractions of compromised links in different Normal-schemes, while links(A) denotes the fractions of compromised links in different Associated-schemes.

1 – 100 nodes are captured in different schemes. Moreover, 100 iterations are performed for each case where x ($1 \le x \le 100$) nodes are randomly captured. The fractions of compromised links, keys and intersections are used to evaluate the resilience. According to Fig. 14, we can find that

1) The fraction of compromised keys in EBS-based schemes is larger than Unital-based schemes when x ($1 \le x \le 100$) nodes are captured, since the ratio of key ring to key pool is larger when EBS-based schemes are applied. Moreover, The fractions of compromised keys in each Associated-scheme and its corresponding Normal-scheme are the same, since the secrecy of associated-keys is same as normal keys;
2) The fraction of compromised intersections is reduced when more linear systems are used to establish secret keys;
3) The fraction of compromised links in each Associated-scheme is larger than it in the corresponding Normal-scheme. However, we can find that the additional broken links are the links established by the compromised intersections. Such links are unavailable in Normal-schemes all the time. What's more, according to the analyses in Section 4.4, although the fractions of compromised links in Associated-schemes are larger than Normal-schemes, the number of available links in each Associated-scheme is still larger than the corresponding Normal-scheme since other uncompromised intersections can still provide secure links between uncompromised nodes.

6.3. Other metrics

Besides key connectivity and resilience against node capture, other metrics are also analyzed, such as scalability, storage overhead, computation complexity and communication complexity. Table 2 illustrates the comparisons of these metrics between Associated-schemes and their corresponding Normal-schemes. According to Table 2, we can find that

1) Scalability. Obviously, we can find that the scalability is not changed when different types of keys are used for key management, since scalability is determined by key assignment;
2) Storage overheads. For the illustrated schemes, storage overheads of various Associated-schemes are $2k$ or $2q$, since each key in these schemes is established by two points in the 2-dimensional space;
3) Communication complexity. When a node wants to communicate to its neighbor, this node broadcasts the IDs of its keys. Then, the other can know whether they have common keys or intersections by searching the list of keys. For key update phase, the key server broadcasts the key update message encrypted by uncompromised keys and each node searches the list of keys to determine whether replace keys. Therefore, the communication overheads are the same for Associated-schemes and the corresponding Normal-schemes;
4) Computation complexity. Comparing with Normal-schemes, the additional computation overheads in Associated-schemes are produced when intersections are required to establish secure links. As analyzed in Section 4.4, intersection computation only involves several elementary arithmetics. What's more, intersection computation means that neighbors can directly communicate with each

Table 2
Comparisons of Associated-schemes and the corresponding Normal-schemes.

	Scalability	Storage	Communication complexity	Computation complexity
Normal-EBS	$\binom{k+m}{k}$	k keys	1: broadcast k IDs; 2: m keys	Search
Associated-EBS	$\binom{k+m}{k}$	$2k$ points	1: broadcast k IDs; 2: m keys	Search & intersection computation
Normal-Unital	$q^2(q^2-q+1)$	q keys	1: broadcast $q+1$ IDs; 2: $\frac{q^5-2q-1}{q+1}$ keys	Search
Associated-Unital	$q^2(q^2-q+1)$	$2(q+1)$ points	1: broadcast $q+1$ IDs; 2: $\frac{q^5-2q-1}{q+1}$ keys	Search & intersection computation

Note: for communication complexity, case 1 denotes the shared key discovery phase, while case 2 represents the key update phase.

other without establishing key-path. In summary, Associated-schemes can reduce many other overheads by intersection computation which produces few computation overheads.

7. Conclusions

In this work, the impact of key connectivity on WSNs is analyzed. When a key management scheme is applied, the efficiency and security of the network might be significantly affected if its key connectivity is lower than 1. To enhance the key connectivity without sacrificing other metrics, we exploit system of equations to generate secret keys and use these keys (associated-keys) to implement key management. According to the theoretical analyses, associated-keys can establish a shared hidden key with the underlying association among the involved equations. Then, nodes can establish secure links by such shared hidden key even when they do not have common keys. Consequently, the probability of direct communication is increased and key connectivity can be improved. Furthermore, linear system of two variables is proved to be sufficient to implement the proposed scheme and generate keys for large scale of networks. Simulations results show that key management schemes that utilize associated-keys have better connectivity than the corresponding schemes with normal keys and other metrics are unaffected. Therefore, the proposed scheme can be used to generate keys to provide efficient protection for WSNs.

References

Bechkit, W., Challal, Y., Bouabdallah, A., Tarokh, V., 2013. A highly scalable key pre-distribution scheme for wireless sensor networks. IEEE Trans. Wirel. Commun. 12, 2 .

Blom R., 1985. An optimal class of symmetric key generation systems. In: Proceedings of the EUROCRYPT 84 Workshop on Advances in Cryptology: Theory and Application of Cryptographic Techniques. pp. 335–338.

Camtepe, S.A., Yener, B., 2007. Combinatorial design of key distribution mechanisms for wireless sensor networks. IEEE/ACM Trans. Netw. 15 (2), 346–358.

Chan H., Perrig A., Song D., 2003. Random key predistribution schemes for sensor networks. In: Proceedings - IEEE Symposium on Security and Privacy. volume 2003-January; pp.197–213.

Das, A.K., 2012a. Improving identity-based random key establishment scheme for large-scale hierarchical wireless sensor networks. Int. J. Netw. Secur. 14 (1), 1–21.

Das, A.K., 2012b. A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks. Int. J. Inf. Secur. 11 (3), 189–211.

Das A.K., Sengupta I., 2007. A key establishment scheme for large-scale mobile wireless sensor networks. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). volume 4882 LNCS; pp. 79–88.

Dong, Q., Liu, D., 2012. Using auxiliary sensors for pairwise key establishment in wsn. Trans. Embed. Comput. Syst. 11, 3.

Du W., Han Y.S., Deng J., Varshney P.K., 2003. A pairwise key pre-distribution scheme for wireless sensor networks. In: Proceedings of the ACM Conference on Computer and Communications Security. pp. 42–51.

Eltoweissy, M., Heydari, M.H., Morales, L., Sudborough, I.H., 2004. Combinatorial optimization of group key management. J. Netw. Syst. Manag. 12 (1), 33–50.

Eltoweissy, M., Moharrum, M., Mukkamala, R., 2006. Dynamic key management in sensor networks. IEEE Commun. Mag. 44 (4), 122–130.

Eschenauer L., Gligor V.D., 2002. A key-management scheme for distributed sensor networks. In: Proceedings of the ACM Conference on Computer and Communications Security. 2002. pp. 41–47.

Habgood, K., Arel, I., 2012. A condensation-based application of cramer's rule for solving large-scale linear systems. J. Discret. Algorithms 10 (1), 98–109.

He, X., Niedermeier, M., De Meer, H., 2013. Dynamic key management in wireless sensor networks: a survey. J. Netw. Comput. Appl. 36 (2), 611–622.

Liu, D., Ning, P., 2005. Improving key predistribution with deployment knowledge in static sensor networks. ACM Trans. Sen. Netw. 1 (2), 204–239.

Liu, D., Ning, P., Du, W., 2008. Group-based key predistribution for wireless sensor networks. ACM Trans. Sens. Netw. 4, 2.

Liu, D., Ning, P., Rongfang, L., 2005. Establishing pairwise keys in distributed sensor networks. ACM Trans. Inf. Syst. Secur. 8 (1), 41–77.

Lo C.C., Huang C.C., Chen S.W., 2009. An efficient and scalable ebs-based batch rekeying scheme for secure group communications. In: Proceedings - IEEE Military Communications Conference MILCOM. pp. 1343–1349.

Malan D.J., Welsh M., Smith M.D., 2004. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In: 2004 Proceedings of the First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON 2004. pp. 71–80.

Mi Q., Stankovic J.A., Stoleru R., 2010. Secure walking gps: A secure localization and key distribution scheme for wireless sensor networks. In: WiSec'10 - Proceedings of the 3rd ACM Conference on Wireless Network Security. p.163–168.

Moharrum, M., Eltoweissy, M., Mukkamala, R., 2006. Dynamic combinatorial key management scheme for sensor networks. Wirel. Commun. Mob. Comput. 6 (7), 1017–1035.

Nam, J., Kim, M., Paik, J., Lee, Y., Won, D., 2014. A provably-secure ecc-based authentication scheme for wireless sensor networks. Sens. (Switz.) 14 (11), 21023–21044.

Rajendiran, K., Sankararajan, R., Palaniappan, R., 2011. A secure key predistribution scheme for wsn using elliptic curve cryptography. ETRI J. 33 (5), 791–801.

Rashid, B., Rehmani, M.H., 2015. Applications of wireless sensor networks for urban areas: a survey. J. Netw. Comput. Appl..

Ruj S., Nayak A., Stojmenovic I., 2011. Fully secure pairwise and triple key distribution in wireless sensor networks using combinatorial designs. In: Proceedings - IEEE INFOCOM. pp. 326–330.

Ruj, S., Nayak, A., Stojmenovic, I., 2013. Pairwise and triple key distribution in wireless sensor networks with applications. IEEE Trans. Comput. 62, 11.

Syed, M.K.U.R.R., Lee, H., Lee, S., Lee, Y.K., 2010. Muqami+: a scalable and locally distributed key management scheme for clustered sensor networks. Ann. Des. Telecommun./Ann. Telecommun. 65 (1–2), 101–116.

Ying B., Makrakis D., Mouftah H.T., Lu W., 2011. Dynamic combinatorial key pre-distribution scheme for heterogeneous sensor networks. In: Communications in Computer and Information Science. volume 186 CCIS; pp.88–95.

Younis, M.F., Ghumman, K., Eltoweissy, M., 2006. Location-aware combinatorial key management scheme for clustered sensor networks. IEEE Trans. Parallel Distrib. Syst. 17 (8), 865–882.

Zhang, J., Varadharajan, V., 2010. Wireless sensor network key management survey and taxonomy. J. Netw. Comput. Appl. 33 (2), 63–75.

Zhu, S., Setia, S., Jajodia, S., 2006. Leap+: efficient security mechanisms for large-scale distributed sensor networks. ACM Trans. Sens. Netw. 2 (4), 500–528.