Key Distribution Strategy of Wireless Sensor Network Based on Multi-Hash Chain

Peng Xiong[1] and Qinggang Su[2,*]

[1]*School of Electronics and Information, Shang hai Dianji University, Shanghai China*
[2]*Chinesisch-Deutsche Kolleg für Intelligente Produktion, Shanghai Dianji University, Shanghai China*
E-mail: xiongp@sdju.edu.cn; suqg@sdju.edu.cn
[*]*Corresponding Author*

Received 23 March 2021; Accepted 01 April 2021;
Publication 02 June 2021

Abstract

Key management is the basis of the security mechanism for wireless sensor networks and services, and random key pre-distribution is the most effective key management mechanism at present. However, there is a potential challenge to most current random key pre-distribution strategies: it is difficult to achieve both ideal network security connectivity and network survivability. In this paper, we present a novel random key pre-distribution scheme based on the hash chain. By adjusting certain system parameters, such as the hash chain length, the number of common auxiliary nodes and the number of hash chains, a sensor node only need to preload a few of keys, making it possible to establish the pairwise key with high probability among its neighboring nodes. The proposed scheme can still maintain strong network survivability even if there are many compromised nodes. The theoretical analysis and simulation experiments show that the proposed scheme is not only effective and secure, but also scalable.

Keywords: Wireless sensor network, key pre-distribution, pairwise key, hash chain.

Journal of Web Engineering, Vol. 20_3, 713–742.
doi: 10.13052/jwe1540-9589.2037

Introduction

Wireless Sensor Networks (WSN) can be deployed in many environments and areas, where conventional networks cannot generally reach, providing solutions to many important application fields, such as pollution monitoring, environment and traffic monitoring, and military [1–3]. When WSN is deployed in inaccessible or even hostile environments, the sensor nodes may face a variety of attacks. The communication may be intercepted, resulting in the leakage of sensitive and secrete information if the information were not well protected. Key management is an information measure for the security of WSN, and its main purpose is to establish the pairwise keys between the communication nodes in WSN. Efficient key management also provides a foundation to other security mechanisms or services, such as the routing [4–6], localization [7], and data fusion [8, 9]. As it is well known, WSN are usually composed of many nodes for which the resources are strictly constrained, and the asymmetric key management strategies [10, 11] are usually considered as unsuitable because of their storage complexity, computation and communication complexity as well as excessive energy consumption. As for symmetric key management strategies, the simplest solution is to use the same key for all the nodes. So, each node only needs to keep one key, and the storage complexity is the smallest, but the resilience against node compromising is the worst. Another scheme is to have any a pair of nodes with different pairwise keys. The survivability of this scheme is the best, and any compromised node will not expose the other node's pairwise keys. But its storage complexity is $O(n)$ (n is the total number of nodes in the networks) and its scalability is bad, making it unsuitable for the large-scale WSN.

Existing symmetric key management strategies for WSN can be roughly divided into two categories: determined and random. The determined key management strategies follow traditional network key management ideas to establish an independent pairwise key between each node and any other node. However, such strategies may impose special requirements for the deployment of the nodes [12–15], or the node storage complexity is usually higher [16]. Moreover, in WSN, a node communicates only with its neighbors, so there is no necessity to establish a pairwise key for any pair of nodes. In the random key management strategies, before deploying, the nodes randomly obtain some keys from a key pool to build the key rings. Once deployed, it generates common keys with a certain probability between the neighboring nodes. By using these common keys, the neighboring nodes can establish their own pairwise keys. Compared with the determined key management strategies, the random key management strategies cannot ensure that any two neighboring nodes can directly establish the pairwise keys, but it effectively reduces the storage complexity, computational complexity and communication complexity of the nodes. So, the random key management is most suitable for WSN [17–20].

However, existing random key management strategies cannot ensure the high network security connectivity and strong survivability simultaneously. The reason is that if the probability of secure connectivity is increased, the number of preloaded keys of the nodes must be increased or the size of the key pool should be reduced. However, to enhance the network survivability, it must reduce the number of preloaded keys of the nodes or increase the size of the key pool. The contradiction between the network security connectivity and the survivability brings a lot of technical challenges to the random key management research of WSN.

This paper proposes a random key pre-distribution scheme based on the hash chains, trying to achieve the ideal compromise between the network security connectivity and the survivability. The basic design idea is that the whole WSN consists of many sensor nodes (that is, the ordinary nodes) and a small number of auxiliary nodes. The key pool consists of a series of hash chains with equal length. The sensor nodes randomly preload a small number of special keys. And the auxiliary nodes randomly select the part of keys from the key pool. Once deployed, the sensor nodes generate new derived keys by the information broadcasted by the auxiliary nodes to establish the pairwise keys between the neighboring sensor nodes. By adjusting the length of hash chains, the number of hash chains and shared auxiliary nodes and so on, the neighboring sensor nodes can establish their pairwise keys with high probability if they preload a small number of special keys, while ensuring a strong survivability even if there are many compromised nodes. In addition, these adjustments are also very effective in reducing the storage complexity of nodes and make the storage complexity of nodes independent of the network size to be more suitable for the large-scale WSN. Theoretical analysis and simulation results show that the proposed scheme can achieve high network security connectivity while maintaining strong survivability.

1 Related Works

PIKE [12] is a determined key pre-distribution scheme. According to the number of nodes N in the network, a grid with $m * m$ sequence number is built, where $m = |\sqrt{N}|$. The nodes are numbered according to the row and column number of the grid. Before deployment, each node builds the pairwise key with the other nodes in same row or same column (a total of $2(\sqrt{N}-1)$ nodes), and then is deployed according to its sequence number. After deployment, the nodes in same rows or same columns can have directly their pairwise keys, and the pairwise keys between the nodes in different rows and different columns must be built with the aid of the nodes of common row or common column between them. Obviously, the node storage complexity of PIKE scheme is $O(\sqrt{n})$, and does not apply to the large-scale networks, and its node deployment also has the special requirements.

Camtepe [16] uses the combinatorial design theory to design the determined key pre-distribution scheme of WSN. Assuming that the total number of nodes in the network is *N*, a symmetrical balanced incomplete block design (BIBD) with the parameters $(n^2+n+1, n+1, 1)$ is generated by the *n*-order finite projective plane (*n* is a prime number meeting $n^2+n+1=N$). This scheme can support the network with n^2+n+1 nodes. Its size of key pool is n^2+n+1 and can generate n^2+n+1 key rings with the size of $n+1$. Any two key rings share at least one common key, and each key appears in $n+1$ key rings. Obviously, in Camtepe's scheme, the probability of security connectivity between any two nodes is 1. But the prime *n* cannot support a network of any arbitrary size. For example, when $N > n^2+n+1$, *n* must be a new larger prime, and an overlarge prime will cause the key ring to increase dramatically to beyond the node's storage space and be not suitable for WSN.

Eschenauer and Gilgor [17] first proposed a random key pre-distribution scheme (referred to as Eschenauer scheme) for WSN. Before deployment, each node randomly selects *k* keys from a key pool with *P* keys $(k \ll P)$, and then the nodes are randomly deployed in a given area. Once deployed, if two neighboring nodes share one key, they can directly build a pairwise key. Otherwise, it needs to build a pairwise key by the intermediate nodes. Thus, its security connectivity probability *p* can be expressed as follows [17]: $p = 1 - \frac{((P-k)!)^2}{(P-2k)!P!}$. From this equation, the nodes can build the pairwise keys with a specific probability if they preload the certain number of keys. The scheme can increase the *k* value (when the *P* value is fixed) or reduce the *P* value (when the *k* value is fixed) to improve the secure connectivity probability.

In Eschenauer scheme, if some of nodes are compromised, the keys held by them will also be compromised. When the other normal nodes use these compromised keys to build their pairwise keys, the corresponding links are known as compromised. When there are α compromised nodes, the compromised probability p_c of the links between the normal neighboring nodes can be expressed as follows: $p_c = 1 - (1 - \frac{k}{p})^{\alpha}$. From this, the p_c can be reduced by reducing the *k* value (when the *P* is fixed) or increasing the *P* value (when the *k* is fixed).

According to the above analysis, for the random key pre-distribution strategies, it is a serious challenge to achieve simultaneously the requirements of high network security connectivity and strong network survivability by adjusting the number of preloaded keys of the nodes or the key pool size.

Chan et al.[18] suggested that it can enhance the network survivability by increasing the shared key threshold, and proposed the q-composite scheme to modify Eschenauer scheme by increasing the shared key threshold from 1 to q. In this scheme, two neighboring nodes share at least q keys to build their pairwise key so that the attacker must capture more nodes in the case of achieving the same compromised probability of the communication links as Eschenauer scheme. However, when the number of compromised nodes is large, the survivability of the q-composite scheme is worse than Eschenauer scheme.

Traynor et al. [21] proposed a random key pre-distribution scheme for heterogeneous WSN. In this scheme, the nodes are divided into two types: strong ability and weak ability. The nodes with strong ability preload a large number of keys, while the nodes with weak ability preload a small number of keys. The pairwise keys are built between the neighboring nodes (isomorphic or isomeric). The scheme fully takes advantage of the role of strong ability nodes to reduce the communication overhead and can achieve the ideal network security connectivity. However, if many strong ability nodes are compromised, it will have a huge impact on the network security connectivity and strong survivability.

Some of researchers apply the node location information [22, 23] and deployment knowledge [24] to the random key pre-distribution scheme. These strategies can improve the targeted of key assignment and enhance network survivability. But the location information or deployment knowledge is known as more demanding requirements.

In recent years, some of researchers have applied lightweight asymmetric key mechanisms such as ECC [25, 26] to WSN, but their computational complexity is only the level of milliseconds, and still much higher than the level of microseconds of symmetric key mechanisms. So its practicality is poor.

2 Preliminaries

2.1 System Hypothesis

In the proposed scheme, the WSN consists of many sensor nodes (that is, the ordinary nodes) and a small number of auxiliary nodes.

The resources of sensor nodes are strictly limited, including their energy, storage capacity, computing power and communication capability and so on. All the sensor nodes are isomorphic. Once deployed, the sensor nodes can only communicate with other sensor nodes within their communication range by the omnidirectional antennas, so the communication links between them are symmetrical. By contrast, the auxiliary nodes are superior to the sensor nodes in terms of energy, storage capacity, computing power and communication capability. The primary role of the auxiliary nodes is to send the preloaded keys to the sensor nodes within their communication range after deployment. Each auxiliary node has a unique identifier, which is a random hash value.

All the sensor nodes and auxiliary nodes are randomly deployed in a given area. Figure 1 shows the WSN structure diagram for the scenario with many sensor nodes and a small number of auxiliary nodes in a given area. The dotted circle represents the communication range of the auxiliary nodes.

In the proposed scheme, it is assumed that the primary purpose of the attacker is to destroy the secure communication between the nodes. The attackers can destroy any number of sensor nodes or auxiliary nodes by stealing a communication channel or the physical capture. If the sensor nodes or auxiliary nodes are compromised, the confidential information carried by them, including the keys, the data, the code and so no, will be exposed to the attackers. The attackers can associate the compromised sensor nodes or auxiliary nodes to launch a collusion attack.

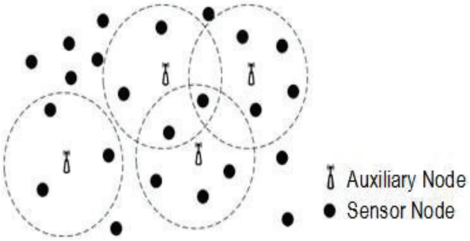


Figure 1 WSN structure of the proposed scheme.

It is assumed that making the auxiliary nodes compromised is more difficult than making the sensor nodes compromised. Moreover, the attackers cannot obtain the identifiers of the normal auxiliary nodes by the compromised auxiliary nodes.

2.2 One-way Hash Chain

A one-way hash chain is a sequence of hash values as follows: $\{x_1, \ldots, x_j, \ldots, x_n\}$, and meets $\{x_j | \forall_j := j = n,\ x_j = H(x_{j-1}, G)\}$. The hash function $H(\cdot)$ meets the following properties: (1) given x_{j-1} and *G*, it is easy to compute x_j; (2) not given *G*, it is difficult to compute x_j even if x_{j-1} is given; or not given x_{j-1}, even if given *G*, it is difficult to compute x_j.

As for the above defined hash chain, a given hash value x_j can be authenticated by repeatedly computing the hash chain and then comparing with the value of the last element x_n.

In the proposed scheme, the *G* is called the generating factor, and the last element x_n of hash chain is *truth*. Obviously, each hash chain has only one *truth*, and the other elements are called chain key.

2.3 Related Definitions

In order to facilitate the discussion and analysis of the proposed scheme, the following definitions are made.

Definition 1: Neighboring Sensor Nodes. For the sensor nodes *u* and *v*, if the physical distance *x* between them is less than their signal range *r*, that is, $x < r$, then they are the neighboring sensor nodes. As shown in Figure 2, the sensor node *u* and *v* are the neighboring sensor nodes.

Definition 2: Neighboring Auxiliary Nodes. If the sensor node *u* is located within the signal range *R* of the auxiliary node A_w, then A_w is called the neighboring auxiliary node of the *u*. Similarly, the *u* is also called the neighboring sensor node of the A_w. Note: A_w is not necessary to locate within the signal range of the *u*. As shown in Figure 2, the auxiliary node A_1, A_4 and A_5 are the neighboring auxiliary nodes of the *u*, but the A_5 is not within the signal range of the *u*.

Definition 3: Auxiliary Communication Area. For a sensor node, assuming that the communication radius of the auxiliary nodes is *R*, then the area in which the sensor node is the center, and the *R* is the radius is known as the auxiliary communication area of the sensor node. As shown in figure

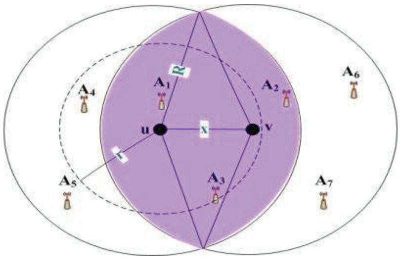


Figure 2 The neighboring relation between sensor node and auxiliary node.

2, the two solid circles are the auxiliary communication areas of the *u* and *v* respectively. These areas contain all the neighboring auxiliary nodes corresponding to the *u* and *v*.

Definition 4: Common Hash Chains. For the sensor node *u* and the auxiliary node A_w, if the *u* selects the $truth_i$ from the hash chain C_i and the A_w also selects one or more the chain keys from the C_i, then it says that the *u* and A_w share the common hash chain C_i. Similarly, if both sensor nodes select the $truth_i$ from the C_i, then it says that they share the common hash chain C_i.

Definition 5: Derived Keys. The keys that the sensor nodes use the received chain keys and the corresponding auxiliary node's identifier to generate are called the derived keys.

Definition 6: Compromised Hash Chains. If a hash chain's *truth* is selected by a compromised sensor node, it is said that the hash chain is compromised.

2.4 The Deployment Model of the Auxiliary Node

The density of the auxiliary nodes determines the number of common auxiliary nodes between the neighboring sensor nodes. For two neighboring sensor nodes, if they are located within their common auxiliary communication area, the corresponding auxiliary nodes are the common auxiliary nodes of them. As shown in Figure 2, the auxiliary node A_1, A_2 and A_3 are the common auxiliary nodes of the neighboring sensor node *u* and *v*.

The random deployment of the auxiliary nodes is described using a homogeneous Poisson Point Process model [27]. Namely, if the density of the auxiliary nodes after deployment is ρ_a, their random deployment can be described as an event sequence complying with the homogeneous Poisson Point Process with the rate ρ_a.

Let the distance between the sensor node *u* and *v* be *x* ($x \leq r$), then the shaded area $\mathrm{Z}_{shaded\ (x)}$ in Figure 2 can be described as follows:

$$Z_{shaded}\,(x) = 2R^2\cos^{-1}\left(\frac{x}{2R}\right) - x\sqrt{R^2 - \frac{x^2}{4}} \tag{1}$$

If the density of the auxiliary nodes is $\rho_a = \frac{N_a}{|Z|}$, where the $|Z|$is the size of the deployment area and the N_a is the total number of auxiliary nodes, then the probability that the node *u* has *s* neighboring auxiliary nodes is equal to the probability that the *s* auxiliary nodes are located in the area πR^2, that is, $p(|NA_u| = s)$. It can be described as follows.

$$p(|NA_u| = s) = \frac{\left(\rho_a \pi R^2\right)^s}{s!} e^{-\rho_a \pi R^2} \tag{2}$$

Thus, the average number of neighboring auxiliary nodes owned by a sensor node can be described as follows.

$$\lambda = E[s \times p(|NA_u| = s)] = \rho_a \pi R^2 \tag{3}$$

Further, the probability that two neighboring sensor nodes at least share *g* common auxiliary nodes is equal to the probability that the *g* auxiliary nodes are located in their common auxiliary communication area $Z_{shaded}(x)$, which can be described as follows:

$$\begin{aligned} p(|NA_{shaded}| \geq g) &= 1 - \sum_{i=0}^{g-1} p(|NA_{shaded}| = i) \\ &= 1 - \sum_{i=0}^{g-1} \frac{\left(\rho_a Z_{shaded(x)}\right)^i}{i!} e^{-\rho_a Z_{shaded(x)}} \end{aligned} \tag{4}$$

3 The Proposed Scheme

The proposed scheme (Key distribution scheme based on multi-hash chain) can be divided into three phases: (1) the key pre-distribution phase; (2) the direct pairwise key establishment phase. Mainly it is on how to build a pairwise key between two neighboring sensor nodes; (3) the path key establishment phase. Mainly it is on how to use the intermedia sensor nodes to help the neighboring sensor nodes to build their pairwise keys.

3.1 The Key Pre-distribution

An off-line trusted server generates a series of hash chains so as to build a key pool. All the hash chains share a seed. For the hash chain C_i, assuming that its generating factor is G_i, the *j*th chain key of the C_i can be generated as follows:

$$k_{i,j} = H^j(seed, G_i) \tag{5}$$

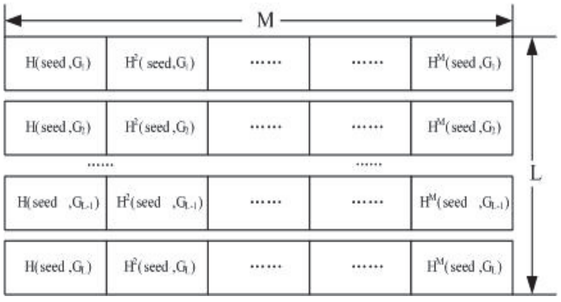
Where $H^j(seed, G_i) = H(H^{j-1}(seed, G_i))(1 = j = M)$. The last element $H^{M+1}(seed, G_i)$ of the hash chain is called the *truth* of the C_i. This element is not an element of the key pool.

In order to generate a key pool, the trusted server selects *L* different generating factors, and repeatedly runs the formula (5) to generate *L* hash chains. The final key pool will consist of *L* hash chains, where each hash chain contains *M* chain keys. As shown in Figure 3.

The key allocation scheme for the sensor nodes and the auxiliary nodes is as follows.

Each sensor node randomly selects q_n different *truths* from the *L* hash chains. In addition, the sensor nodes preload the hash function $H(\cdot)$ and the pseudo-random function $F(\cdot)$ to generate the derived keys. Unlike other random key pre-distribution strategies, in the proposed scheme, the sensor nodes do not need to preload any chain key from the key pool.

Each auxiliary node must preload the following confidential information: 1. q_a different chain keys are randomly selected from the key pool, where there is no restriction on the number of selected chain keys from each hash chain; 2. if one or more chain keys of the C_i are selected by an auxiliary node, this auxiliary node must preload the corresponding hash mapping $F(truth_i)$ and generating factor G_i. For example, if the chain key $k_{3,4}$ is selected by an auxiliary node, the auxiliary node simultaneously preloads $F(truth_3)$ and generating factor G_3. The $F(truth)$ is the broadcasting key of the auxiliary node.



$H(seed, G_1)$	$H^2(seed, G_1)$	......	......	$H^M(seed, G_1)$
$H(seed, G_2)$	$H^2(seed, G_2)$	......	......	$H^M(seed, G_2)$
$H(seed\ , G_{L-1})$	$H^2(seed\ , G_{L-1})$	......	......	$H^M(seed\ , G_{L-1})$
$H(seed, G_L)$	$H^2(seed, G_L)$	......	......	$H^M(seed, G_L)$

Figure 3 The composition of the key pool.

All the sensor nodes and auxiliary nodes can be randomly deployed in the designated area after preloading the required confidential information.

3.2 Direct Pairwise Key Establishment

Once deployed, the auxiliary nodes broadcast their preloaded chain keys and identifiers. If the auxiliary nodes contain multiple chain keys from the different hash chains, they use different packets to broadcast these chain keys respectively. Each packet contains three parts: (1) the chain keys from the same hash chain and their indexes; (2) the generating factor of the corresponding hash chain; (3) the auxiliary node's identifier. The data packets are encrypted with the corresponding *F*(*truth*)s. For example, if the auxiliary node A_w preloads τ chain keys from the hash chain C_i, the packet format for broadcasting is as follows:

$$i, E(F(truth_i), ID_{A_w}|k_{i,j_1}|\dots|k_{i,j_\tau}|j_1|\dots|j_\tau|G_i)$$

The ID_{A_w} is the identifier of the A_w. The *i* is the identifier of the hash chain. The parameter τ must be greater than or equal to 1 and less than or equal to *M*. The *E*(*K*,*m*) denotes that the information *m* is encrypted with the key *k*.

If the sensor node shares common hash chains with its neighboring auxiliary nodes, this sensor node can decrypt the received broadcast packets and use the corresponding *truths* and generating factors to authenticate the received chain keys by the formula (5). If the authentication of the chain key $k_{i,\ j}$ is passed, the corresponding sensor node can use this chain key and the identifier of the corresponding neighboring auxiliary node to generate a derived key $k_{i,j,w}$ as follows:

$$k_{i,j,w} = F(k_{i,j}||ID_{A_w}) \tag{6}$$

Note: if the chain key $k_{i,j}$ cannot pass his authentication, or the range of the parameter *j* is not between 1 and *M*, the $k_{i,j}$ will be discarded because the $k_{i,j}$ may be forged by the attackers.

Once the authentication process is finished, the sensor nodes will delete all the received identifiers of the auxiliary nodes and the generating factors.

Although the sensor nodes may share the same hash chains with the different neighboring auxiliary nodes, the derived keys are not the same because of the difference of the auxiliary node identifiers. The total number of derived keys generated by each sensor node is related to the number of neighboring auxiliary nodes. Thus, the number of derived keys of different sensor nodes may be different.

Once generated all the derived keys, the sensor nodes will broadcast the packets containing the indexes of the derived keys so as to indicate their respective derived keys. For example, the index <*i*, *j*, *w*> represents the derived key $k_{i,j,w}$ is generated by the chain key $k_{i,j}$ and the identifier of the auxiliary node A_w. Assuming that two sensor nodes have same derived key indexes and the number of them is *t*, that is, $\{k_1, k_2, \ldots, k_t\}$, and is greater than the threshold *q*, that is, $q = t$, they can use this *t* derived keys to generate the pairwise key as follows:

$$k_{uv} = k_1 \oplus k_2 \oplus \cdots \oplus k_t \tag{7}$$

In the formula (7), the symbol $\oplus$ means "exclusive OR (XOR)" operation.

Once the pairwise keys is generated, the sensor nodes will delete all the derived keys.

3.3 The Path Key Establishment

In the above mentioned, if $q > t$, where the q is the threshold which is the smallest number of derived keys generating a pairwise key, it means that the pairwise key cannot be built directly between the neighboring sensor nodes. In this case, there are two options: the first is to use the method similar to Eschenauer scheme to build the pairwise key; the second is not to build the pairwise key. If the node density in the network is large enough to ensure a very high probability of information transmission, some neighboring sensor nodes do not build a secure communication link is acceptable.

4 Performance Analysis

This section mainly analyzes the performance of the proposed scheme and compares it with two typical random key pre-distribution schemes [10, 11].

4.1 Analysis of Network Security Connectivity

The network security connectivity is defined as the probability that WSN can build a secure communication link. If two neighboring sensor nodes share a sufficient number of derived keys, they can build a secure communication link.

For the neighboring sensor node *u* and *v*, if one of the following two conditions is met, then they cannot share the derived keys: (1) they do not share any hash chain; (2) the common auxiliary nodes and them do not share any hash chain. If these two conditions are all not true, then the neighboring sensor node *u* and *v* may share the derived keys.

Let m_i be the number of common derived keys between the *u* and *v* generated by one of their common neighboring auxiliary nodes, and *m* is the total number of common derived keys generated by all *g* common neighboring auxiliary nodes, that is, $m = m_1 + m_2 + \cdots + m_g$. A lemma and a theorem are proved as follow.

Lemma 1: assuming that the neighboring sensor node *u* and *v* have only one common neighboring auxiliary node. If the number of *truths* shared by the *u* and *v* is *l*, the number of common derived keys between them is not more than *l***M*.

Proof. The *truth* of and generating factor from same hash chain can authenticate all the chain keys of this hash chain, so each *truth* can authenticate *M* chain keys. Since the chain keys generating the shared derived keys between the *u* and *v* must be authenticated by the corresponding *truths* shared by them, and the number of *truths* shared by the *u* and *v* is *l*, even if all of the chain keys of the hash chains corresponding to the *l truths* are selected by the common neighboring auxiliary nodes of the *u* and *v*, and the number of common derived keys between the *u* and *v* is just *l***M*. Once one or more of the chain keys are not selected, the number of common derived keys between the u and v does not exceed *l***M*. Proof finished.

Theorem 1: assuming that there are *g* common neighboring auxiliary nodes between the neighboring sensor node *u* and *v*, the number of derived keys generated by each of them is $m_1, m_2, \ldots, m_g$ ($\forall i, m_i \ll q_n$), respectively. Then the number of shared *truths* between the *u* and *v* should be $\lceil \frac{\max(m_1, m_2, \ldots, m_g)}{M} \rceil$ at least.

Proof. If the number of shared *truths* between the neighboring sensor node *u* and *v* is *l*, according to Lemma 1, the number of derived keys generated by any of their common neighboring auxiliary nodes is between 0 and *l***M*. Conversely, if they generate m_i derived keys based on the common auxiliary node A_i, the number of shared *truths* between them must be $\lceil \frac{m_i}{M} \rceil$ at least. For all m_i $(i = 1, \ldots, g)$, if the *u* and *v* share the enough number of *truths* so as to generate the maximum number of common derived keys, then they definitely also be able to generate the other number of common derived keys by the other common auxiliary nodes, so that the number of *truths* shared by the *u* and *v* is $\lceil \frac{\max(m_1,m_2,\ldots,m_g)}{M} \rceil$ at least. Proof finished.

Assuming that the number of shared *truths* between the *u* and *v* is *l*, for any common auxiliary node between them, such as the A_w, if the *u* and *v* can generate *s* derived keys by the A_w, the method that the A_w preloads q_a chain keys is as follows: first the A_w selects randomly *s* chain keys from the hash chains corresponding to the *l truths* shared by the *u* and *v*, a total of $\binom{l\times M}{s}$ kinds of selection methods. Then $(q_a - s)$ chain keys are randomly selected from the remaining $(L-l)$ hash chains, a total of $\binom{(L-l)\times M}{q_a-s}$ methods. Thus, the method that the A_w preloads q_a chain keys can be described as follows:

$$\Omega(l,s) = \binom{l \times M}{s}\binom{(L-l) \times M}{q_a - s} \tag{8}$$

If there are *g* common neighboring auxiliary nodes between the *u* and *v*, and the number of derived keys generated by each auxiliary node is m_1, m_2,..., m_g respectively, then the probability that they share *m* derived keys can be described as follows: firstly, the *u* can select randomly q_n *truths* from the *L* hash chains, and a total of $\binom{L}{q_n}$ kinds of selection methods. Secondly, a total of $\sum_{m1+m2+\cdots+mg=m}$ methods permit the common auxiliary nodes to provide the shared chain keys. According to Theorem 1, the number of shared *truths* between the *u* and *v* is between $\lceil \frac{\max(m_1,m_2,\ldots,m_g)}{M} \rceil$ and q_n. Assuming that the number of shared *truths* between them is *l*, then the *v* can select randomly *l truths* from the q_n hash chains selected by the *u*, and randomly selects (q_n-*l*) *truths* from the remaining $(L - q_n)$ hash chains. Thus, the *v* has a total of $\binom{q_n}{l}\binom{L-q_n}{q_n-l}$ kinds of selection methods for its *truths*. And the methods that each common auxiliary node selects chain keys are shown in the formula (8).

Thus, the probability generating *m* common derived keys between the *u* and *v* by *g* common neighboring auxiliary nodes can be described as follows:

$$p(m) = \frac{\sum_{m_1+m_2+\cdots+m_g=m} \sum_{i=\left\lceil \frac{\max(m_1,m_2,\ldots,m_g)}{M} \right\rceil}^{q_n} \binom{q_n}{i}\binom{L-q_n}{q_n-i}\Omega(i,m_1)\ldots\Omega(i,m_g)}{\binom{L}{q_n}\binom{L\times M}{q_a}^g} \tag{9}$$

Obviously, the probability that the number of derived keys shared by two neighboring sensor nodes is less than the threshold *q* is $\sum_{i=0}^{q-1} p(i)$, where the *p*(*i*) is defined as the formula (9). Thus, the probability that the number of derived keys shared by two neighboring sensor nodes is at least *q* can be described as the formula (10):

$$
\begin{aligned}
p_{connect} &= 1 - (p(0) + p(1) + \cdots + p(q-1)) \\
&= 1 - \frac{\sum_{m_1=0}^{q-1} \sum_{m_2=0}^{q-m_1-1} \cdots \sum_{m_g=0}^{q-m_1-\cdots-m_{g-1}-1} \sum_{i=\left[\frac{\max(m_1,m_2,\ldots,m_g)}{M}\right]}^{q_n} \binom{q_n}{i}\binom{L-q_n}{q_n-i}\Omega(i,m_1)\ldots\Omega(i,m_g)}{\binom{L}{q_n}^2 \binom{L\times M}{q_a}^g}
\end{aligned}
\tag{10}
$$

4.2 Parameter Analysis

It can be seen from the formula (10) that some system parameters, such as the length *M* of hash chain, the number *L* of hash chains, the number *g* of common neighboring auxiliary nodes, etc., will affect the network security connectivity. These parameters will be discussed in detail below, and the proposed scheme is compared with two typical random key pre-distribution schemes. All the following analyzes are taken place in the case of 99.99% network security connectivity.

4.2.1 The effect of the common neighboring auxiliary nodes

The number of common derived keys between two neighboring sensor nodes depends on the number of common neighboring auxiliary nodes between them. So, the larger the number of common neighboring auxiliary nodes, the greater the probability that the two neighboring sensor nodes can share the derived key. Figure 4 shows the ratio between the number of preloaded *truths* required by each sensor node and the number of preloaded chain keys required by each auxiliary node when the number of common neighboring auxiliary nodes is 1, 2 and 3, respectively.

Figure 4 provides the results consistent with the above observations, that is, the larger the number of common auxiliary nodes, the smaller the number of preloaded *truths* required by the sensor nodes. Moreover, it also reveals a very interesting phenomenon that the increase in the number of common auxiliary nodes cannot obviously reduce the number of preloaded *truths* required by the sensor nodes. For example, when the number of preloaded chain keys required by the auxiliary nodes is 3000 and the number of common

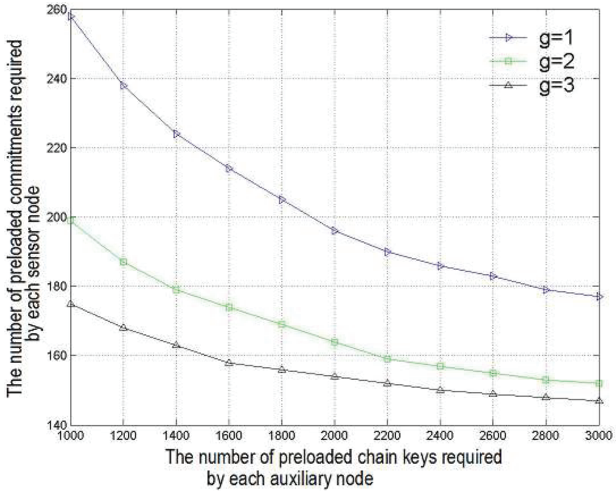


Figure 4 The ratio of the number of preloaded *truths* required by each sensor node and the number of chain keys of pre-distribution for each auxiliary node, where: M = 8, L = 2500, q = 1.

auxiliary nodes increases from 2 to 3, the number of preloaded *truths* required by the sensor nodes is reduced by only 3.29%, from 152 to 147.

4.2.2 The effect of the hash chain length

Each chain key in the same hash chain can be authenticated by the corresponding *truth* and generating factor. Thus, the longer the length of hash chains, the more the chain keys authenticated by the same *truth* and generation factor, that is, the smaller the number of preloaded *truths* required by the sensor nodes. Figure 5 shows how the length of hash chains affects the number of preloaded *truths* required by the sensor nodes.

It is to be noted that in Figure 5, the longer the length of hash chains, the smoother the curves will become. This means that the attenuation rate of the number of preloaded *truths* required by the sensor nodes will slow. For example, when the number of preloaded chain keys required by the auxiliary nodes increases from 1000 to 3000 and the length of hash chains is 4, the number of preloaded *truths* required by the sensor nodes will decrease by 32.8% (from 368 to 247). And when the length of hash chains is 16, the required number of preloaded *truths* is reduced by only 13.8% (from 116 to 100). Therefore, if the length of hash chains is long enough, increasing the number of preloaded chain keys required by the auxiliary nodes do not

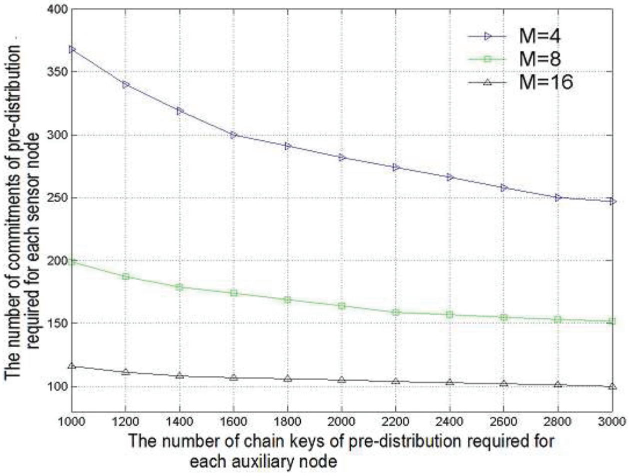


Figure 5 The effect of different hash chain length on the number of preloaded *truths* required by each auxiliary node. Where: $L * M = 20000$, g = 2, q = 1.

obviously affect the security connectivity of the network. Figure 5 also shows that the length of hash chains is an important factor affecting the network connectivity and can obviously affect the number of preloaded *truths* required by the sensor nodes.

4.3 Comparison with Two Typical Random Key Management Schemes

4.3.1 Comparison with Eschenauer scheme

In Eschenauer scheme, when the total number of keys in the key pool increases, the number of preloaded keys required by the nodes must be larger to achieve the high security connectivity. However, due to the resource constraint of the nodes, the number of preloaded keys required by the nodes should not be too much. But from a security point, the larger the number of keys in the key pool, in order to get more keys, the lager the number of the nodes that the attackers need to capture. So, the larger the total number of keys in the key pool or the smaller the number of preloaded keys, the stronger the survivability of the network.

Figure 6 shows the number of preloaded *truths* or keys required by the sensor nodes under the condition that the number of keys in the key pool and the number of hash chains are different in Eschenauer scheme and the proposed scheme. In the proposed scheme, the number of preloaded chain

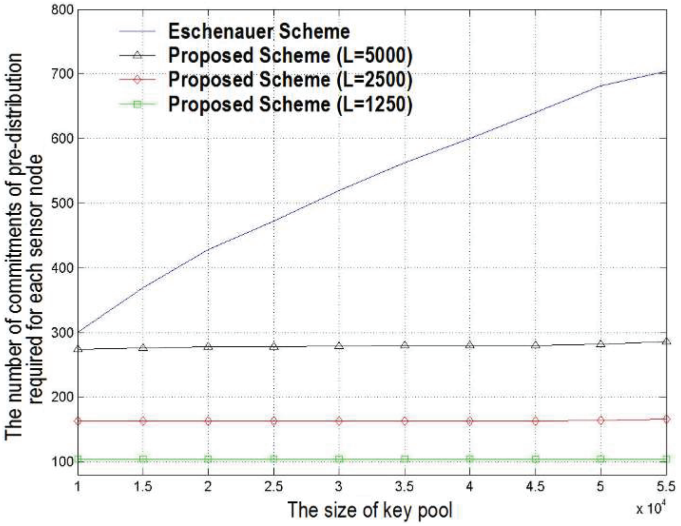


Figure 6 The comparison between the proposed scheme and Eschenauer scheme.

keys required by the auxiliary node is 2000 and the number of common auxiliary nodes is 1.

It can be seen from Figure 6 that the number of preloaded *truths* required by the sensor nodes in the proposed scheme is smaller than the number of preloaded keys required by the nodes in Eschenauer scheme, and in the case of the different number of hash chains, regardless of the total number of keys, if the number of hash chains is fixed, the number of preloaded *truths* required by the sensor nodes is nearly unchanged. Therefore, the proposed scheme has a striking feature, that is, regardless of the size of the key pool, if the appropriate adjustment of the number of hash chains, the number of preloaded *truths* required by the sensor nodes can remain very low, and nearly unchanged. This means that the proposed scheme is very effective when the key pool is large.

4.3.2 Comparison with the q-composite scheme

The q-composite scheme improves the survivability of the network by increasing the threshold of the shared keys. However, it also means that the number of preloaded keys required by the nodes must be larger or the number of keys in the key pool needs to be reduced. Figure 7 shows the comparison of the proposed scheme with the q-composite scheme in the case of the different length of hash chains. In the proposed scheme, the number of preloaded chain

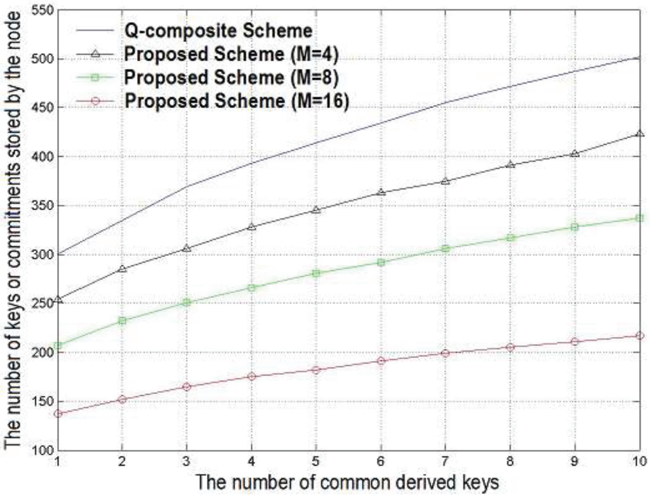


Figure 7 The comparison between the proposed scheme and the q-composite scheme.

keys for each auxiliary node is 2000 and the number of common auxiliary nodes is 1.

Figure 7 shows that the number of preloaded *truths* for each sensor node in the proposed scheme is smaller than the number of preloaded keys for each node in the *q*-composite scheme. Moreover, when the length of hash chains grows slightly, the number of preloaded *truths* for each sensor node will be obviously reduced, which means that for the proposed scheme, if the length of hash chains is large enough, it can just greatly improve the security of the network connectivity if it slightly increases the number of preloaded *truths* for each sensor node.

4.4 Comparison Between Theoretical Analysis and Simulation Experiments

In order to prove the formula (10), this paper runs the simulation experiments. In the experiments, the total number of sensor nodes and auxiliary nodes is 1000, and four scenarios are set up to verify the difference between the ideal link and the actual link in the case of different number of hash chains, different length of hash chains, and different threshold of derived keys. For two neighboring sensor nodes: if there is at least one common auxiliary node, then there is an ideal link between them; If there is at least one common auxiliary node and there is at least one shared hash chain between the common auxiliary node and them, then there is a real link.

Table 1 The parameters setting in different simulation circumstance

Scenarios	Parameters
Scenario 1	M = 4, L = 5000, g = 2, curkeys = 1000, m = 1, truths = 368
Scenario 2	M = 8, L = 2500, g = 3, curkeys = 3000, m = 1, truths = 164
Scenario 3	M = 4, L = 5000, g = 1, curkeys = 2000, m = 3, truths = 437
Scenario 4	M = 8, L = 2500, g = 1, curkeys = 2000, m = 8, truths = 293

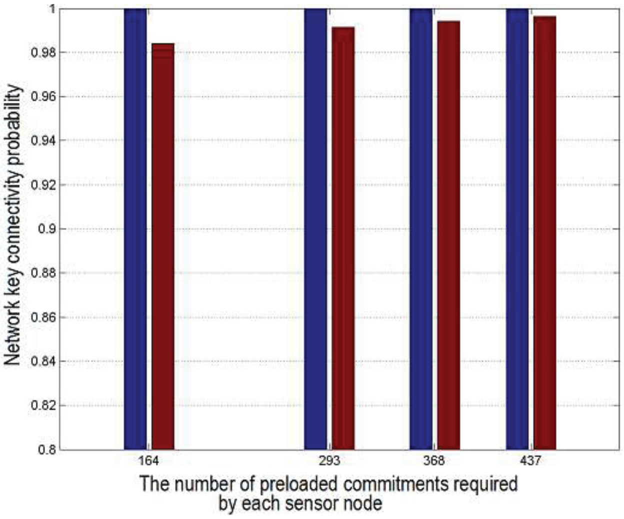


Figure 8 The comparison between the simulation results and the theoretical results.

Table 1 shows the experiment conditions including the values of preloaded truths for each sensor node in each scenario in the case of security connectivity probability of 99.99% according to formula (10), where the curkeys means the number of preloaded chain keys for the auxiliary nodes.

Figure 8 shows the ratio of the actual link to the ideal link from the simulation experiment. It can be seen from the results that when the number of preloaded *truths* for each sensor node is 164,293,368 and 437 respectively, the ratio is 98.16%, 99.13%, 99.41% and 99.64% respectively, which means that the difference between the experiment results and the theoretical analysis results is at most 1.84%, so it proves the correctness of formula (10).

As can be seen from the above analysis, the proposed scheme greatly improves the network security connectivity compared with other typical random key pre-distribution schemes. The reason is the correlation between the chain keys of the hash chains as well as the effect of the common auxiliary nodes. The hash chains can improve the correlation between the chain keys so that the sensor nodes can generate a large number of derived keys with the aid of the auxiliary nodes as long as the sensor nodes save a small amount of special confidential information (i.e. the *truths*). Thus, it enables higher network security connectivity with smaller storage overhead.

5 The Security Analysis of the Proposed Scheme

The proposed scheme can effectively defend against the passive attacks. On the one hand, each broadcast packet from the auxiliary nodes is encrypted with the key F(*truth*) so that the attackers cannot get any data of the packet; on the other hand, the sent packets between the sensor nodes only contain the indexes of the derived keys, even if the attackers can intercept these packets, and cannot get any the derived key.

The proposed scheme can also effectively defend against the active attacks. For example, if the attackers forge the broadcast packets and launch a DoS attack on a sensor node, because it cannot pass the authentication, these packets will be also discarded by the sensor node. Even if the attackers got the F(*truth*) by capturing the auxiliary nodes, they cannot forge the chain keys because the sensor nodes only accept the authenticated chain keys. The most common active attack is that the attackers take advantage of the compromised sensor nodes or auxiliary nodes to launch an attack on the network. Here, it proves a theorem.

The security of the pairwise keys between the normal sensor nodes can be described as follows.

Theorem 2: for two normal sensor nodes, even if all the hash chains shared by them are completely compromised, as long as there is a secure common auxiliary node and there is at least one derived key generated by this auxiliary node, the attackers cannot get their pairwise key.

Proof. If all the hash chains shared by two normal sensor nodes are completely compromised, it means that the attackers can get all the chain keys of the corresponding hash chains. However, it can be seen from the formula (6) that for the derived key $k_{i,j,w}$, if the auxiliary node's identifier ID_{A_w} is secure, and the corresponding derived key is certainly secure; therefore, as long as there is a secure common auxiliary node, it can generate the corresponding derived key, and the derived key is certainly secure. From the formula (7) can also draw the following conclusions: as long as there is a secure one among *t* shared derivative keys, then the generated pairwise key is certainly secure. Proof finished.

The following analysis shows the effect of the compromise of the sensor nodes or auxiliary nodes on the network survivability. Namely, how the compromise of some sensor nodes or auxiliary nodes affects the compromise probability of the communication links between two normal neighboring sensor nodes. Obviously, the smaller this probability, the stronger the survivability of the network.

5.1 The Sensor Nodes Compromise Analysis

First it only considers the scenario in which a small part of the sensor nodes is compromised, and all the auxiliary nodes are secure. Obviously, if a sensor node is compromised, all the confidential information which it holds will be exposed. Moreover, the compromised sensor nodes can decrypt some of the received broadcast packets from its neighboring auxiliary nodes and get the corresponding chain keys and the generating factors.

When a sensor node is compromised, the probability that a chain key belongs to the compromised hash chain is $\frac{q_n}{L}$. Since each sensor node can have an average of λ neighboring auxiliary nodes (λ is defined in the formula (3)), the probability that a chain key is selected by the λ neighboring auxiliary nodes is $\frac{\lambda q_a}{L \times M}$ (assuming that the chain keys selected by the λ neighboring auxiliary nodes are different, then the number of compromised chain keys is the largest). Thus, the probability that a derived key is secure is $1 - \frac{q_n}{L} \times \frac{\lambda q_a}{L \times M}$. Assuming that there are α compromised sensor nodes, under this scenario, the probability that a derived key is still secure is $(1 - \frac{q_n}{L} \times \frac{\lambda q_n}{L \times M})^{\alpha}$. Thus, the probability that a derived key is compromised can be described as follow:

$$p_{c1} = 1 - (1 - \frac{q_n}{L} \times \frac{\lambda q_n}{L \times M})^{\alpha} \tag{11}$$

If the pairwise key between two neighboring sensor nodes is generated by *t* corresponding derived keys, the probability that a communication link is compromised is $(p_{c1})^t$. Therefore, when there are α compromised sensor nodes, the probability of compromised communication link between two normal neighboring sensor nodes can be described as follows:

$$p_{compromised\_1} = \sum_{t=q}^{g \times q_a} (p_{c1})^t \frac{p(t)}{p_{connect}} \tag{12}$$

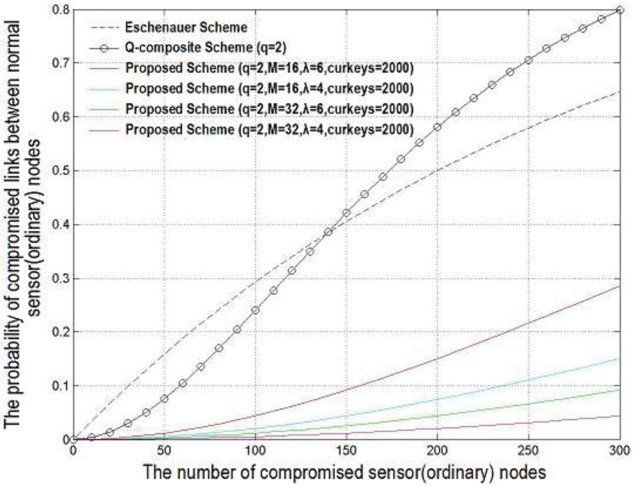


Figure 9 The probability of compromised communication links between the normal sensor node is compromised.

Where the $p\ (t)$ and the $p_{connect}$ are defined in the formula (9) and the formula (10) respectively, the g is the number of common auxiliary nodes.

Figure 9 shows the effect of the different number of compromised sensor nodes on the communication links of the normal sensor nodes when the security connectivity probability is 0.5, the number of common auxiliary nodes is 1, the number of preloaded chain keys or *truths* for each (sensor) node is 200.

Figure 9 compares the network survivability of the proposed scheme, Eschenauer scheme, and the q-composite scheme. Obviously, the proposed scheme provides the better survivability than that of the other two schemes. For example, Eschenauer scheme and the q-composite scheme have respectively 59% and 71% communication links between the normal sensor nodes to be compromised when there are 250 compromised sensor nodes. However, in the proposed scheme, in the case of the same number of compromised nodes, when the length of hash chain is 16, there being only 21% ($\lambda = 6$) or 11% ($\lambda = 4$) communication links between the normal sensor nodes are compromised. For the proposed scheme, the probability of compromised communication links can be reduced by reducing appropriately the average number of common auxiliary nodes or increasing the length of hash chains. As can be seen from Figure 9, when the length of hash chains is doubled from 16 to 32, the probability of compromised communication links is obviously reduced by more than two times.

5.2 The Auxiliary Nodes Compromise Analysis

This section considers the scenario in which only some of the auxiliary nodes are compromised and all the sensor nodes are normal. The attackers can use the compromised auxiliary nodes to broadcast the forged identifiers, which can cause the sensor nodes to generate the derived keys by this information. However, in the proposed scheme, the attackers cannot generate or forge any derived key because they cannot get any knowledge about the hash function $H(\cdot)$ and the pseudorandom function $F(\cdot)$. From the point of derived key generation, the forged identifiers have no effect on the derived keys, and the attackers cannot get any derived key.

However, there is a constraint in the proposed scheme, that is, if all the auxiliary nodes are compromised in the phase of derived key generation, the sensor nodes cannot build the pairwise keys. In order to solve this problem, it is possible to make the same appearance of auxiliary nodes and sensor nodes as far as possible, because the attackers cannot distinguish them, they can only randomly choice the attacked targets. In addition, the auxiliary nodes are just to broadcast their own identifiers and chain keys after deployment, it is impossible to damage all of the auxiliary nodes in a very short period by the attackers. Thus, in the proposed scheme, it is also nearly impossible that the pairwise keys cannot be built between the neighboring sensor nodes because all of the auxiliary nodes are compromised in a short period.

5.3 The Analysis that the Sensor Nodes and Auxiliary Nodes are all Compromised

Obviously, unlike the scenarios of the 5.1 and 5.2 sections, the attackers are more likely to attack a mix number of nodes in a certain area including the sensor nodes and auxiliary nodes, so that some of auxiliary nodes and sensor nodes in this area are all compromised. In the proposed scheme, the number of deployed sensor nodes is generally far more than the auxiliary nodes. Therefore, if some of the sensor nodes and auxiliary nodes are compromised, under normal circumstances, the number of compromised sensor nodes should be more than the compromised auxiliary nodes.

Assuming that there are α compromised sensor nodes and β compromised auxiliary nodes and it can be further assumed that $\alpha > \beta$. For the β compromised auxiliary nodes, it can be divided into two parts: one part is neighboring to the α compromised sensor nodes and called as the neighboring compromised auxiliary nodes; the other part is not neighboring to any

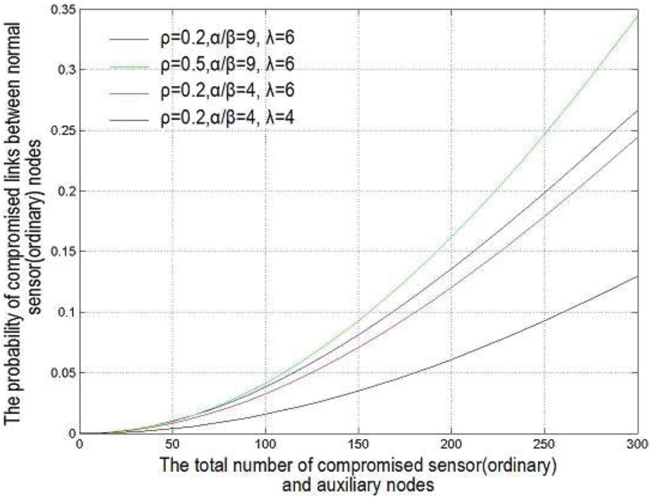


Figure 10 The probability of the compromised communication link between the normal nodes when there are the compromised sensor nodes and compromised auxiliary nodes.

compromised sensor node and called as the separated compromised auxiliary nodes. For the neighboring compromised auxiliary nodes, under there being α compromised sensor nodes, regardless of the number of neighboring auxiliary nodes, the compromised probability of a derived key is p_{c1}. On the other hand, assuming that the number of separated compromised auxiliary nodes is $\rho\beta$ $(0 < \rho < 1)$, under there being α compromised sensor nodes, the probability that a chain key belongs to a compromised hash chain is $1 - (1 - \frac{q_n}{L})^{\alpha}$. Thus, under there being $\rho\beta$ separated compromised auxiliary nodes, the probability that a derived key is compromised should be formula (13).

$$p_{c2} = 1 - \left(1 - \left(1 - \left(1 - \frac{q_n}{L}\right)^{\alpha}\right)\left(\frac{\rho\beta \times q_a}{L \times M}\right)\right)^{\rho\beta} \tag{13}$$

Therefore, under there being α compromised sensor nodes and β compromised auxiliary nodes, the probability of the compromised communication links between the normal sensor nodes can be described as follows:

$$p_{insecure\_2} = \sum_{t=q}^{g \times q_a} (p_{c1} + p_{c2})^t \frac{p(t)}{p_{connect}} \tag{14}$$

Figure 10 shows the comparison results of the compromised communication links between the normal sensor nodes when the relevant parameters are changed, where the secure connectivity probability is 0.5, the number of preloaded *truths* for each sensor node is 200, the number of preloaded chain keys for each auxiliary node is 2000 and the hash chain length is 16.

As can be seen from Figure 10, in the proposed scheme, the compromise of the sensor nodes are still the main factor affecting the compromise of the communication links. However, even if there are many separated compromised auxiliary nodes, the proposed scheme can still remain high survivability. For example, when the total number of compromised sensor nodes and compromised auxiliary nodes is 300, even if the half of the compromised auxiliary nodes are separated, the probability of the compromised communication links is only 34%.

6 Summary and Future Work

The key management is an important mechanism to ensure the secure communication in WSN. From the current research results and development trends, the random key management schemes are suitable for WSN with the strict limit of node resources. In this paper, the random key pre-distribution scheme based on hash chain is proposed to effectively solve the problem simultaneously achieving the high network security connectivity and strong survivability by increasing the key's correlation, which cannot be achieved in the other similar schemes for WSN. The theoretical analysis and simulation experiments show that the proposed scheme can provide the ideal security connectivity and survivability, which provides a feasible and important solution for the key management research of large-scale WSN.

For the proposed scheme, there are some problems which are still worthy of further study, such as, how to optimize the system parameters to achieve the best network security connectivity and network survivability; Also, in the case of other types of attacks, such as Wormhole attack, Sinkhole attack, etc., the survivability of networks researche.

Acknowledgement

This work sponsored by the National Natural Science Foundation of China (No.11261037) and the Nation Science Foundation of Shanghai (No.19ZR1420000).

References

[1] Akyildiz F, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor network: A survey. Computer Networks, 38(4): 393–422, 2002.

[2] K.M. Martin and M. Paterson. An application-oriented framework for wireless sensor network key establishment. Electronic Notes in Theoretical Computer Science, vol. 192, no. 2, pp. 31–41, 2008.

[3] O. Boyinbode, H. Le, M. Takizawa. A survey on clustering algorithms for wire-less sensor networks. Int. J. Space-Based Situated Comput. 1 (2) (2011) 130–136.

[4] S.K. Gupta, P. Kuila, P.K. Jana. GAR: an energy efficient GA-based routing for wireless sensor networks. In: International Conference on Distributed Computing and Internet Technology 2013, LNCS, vol. 7753, Springer, 2013, pp. 267–277.

[5] P. Kuila, P.K. Jana. An energy balanced distributed clustering and routing algorithm for wireless sensor networks. In: 2012 2nd IEEE International Conference on Parallel Distributed and Grid Computing (PDGC), IEEE, 2012, pp. 220–225.

[6] Deng J, Han R, Mishra S. INSENS: Intrusion-tolerant routing in wireless Sensor Networks. Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 32–39, 2003.

[7] Lazos L, Poovendran R. SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks. Proceedings of the 2004 ACM workshop on Wireless Security (WISE), pp. 21–30, 2004.

[8] Przydatek B, Song D, Perrig A. SIA: Secure Information Aggregation in Sensor Networks. Proceedings of the 1st international Conference on Embedded Networked Sensor Systems, pp. 255–265, 2003.

[9] Crossbow Technology. MICA2: Wireless measurement system. http://www.xbow.com/Products/Product_pdf_files/Wirelesspdf/6020-0042-04_A_MICA2.pdf

[10] Koc KC. High-Speed RSA implementation. RSA Laboratories, Technical Report, TR201, 1994.

[11] Neuman B.C, Tso T. Kerberos: An authentication service for computer networks. IEEE Communications, 1994, 32(9):33–38.

[12] Chan H, Perrig A, PIKE: Peer Intermediaries for Key Establishment in Sensor Networks. Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), pp. 524–535, March 2005.

[13] J.-P. Sheu and J.-C. Cheng. Pair-wise path key establishment in wireless sensor networks. Computer Communications, vol. 30, no. 11–12, pp. 2365–2374, 2007.

[14] C.-L. Chen, Y.-T. Tsai, and T.-F. Shih. A novel key management of two-tier dissemination for wireless sensor network. In Proceedings of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS'12), pp. 576–579, Palermo, Italy, July 2012.

[15] K.-A. Shim, Y.-R. Lee, and C.-M. Park. EIBAS: an efficient identity-based broadcast authentication scheme in wireless sensor networks. Ad Hoc Networks, vol. 11, no. 1, pp. 182–189, 2013.

[16] Camtepe S.A, Yener B. Combinatorial design of key distribution mechanisms for wireless sensor networks. Proceedings of the Computer Security—ESORICS. Berlin: Springer-Verlag, 2004. 293–308.

[17] Eschenauer L, Gligor V. A Key Management Scheme for Distributed Sensor Networks. Proceedings of 9th ACM Conference on Computer and Communications security, pp. 41–47, 2002.

[18] Chan H, Perrig A, Song D. Random Key Pre-Distribution Schemes for Sensor Networks. Proceedings of IEEE Symposium on Security and Privacy, pp. 197–213, 2003.

[19] C.-T. Li, C.-Y. Weng, C.-C. Lee, C.-W. Lee. Towards secure and dynamic password-based user authentication scheme in hierarchical wireless sensor networks. Int. J. Secur. Appl. 7(3) (2013) 249–258.

[20] J. Kim, D. Lee, W. Jeon, Y. Lee, D. Won. Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks. Sensors 14(4) 6443–6462, 2014.

[21] Traynor P, Choi H, Cao G, Zhu S, Porta T. L. Establishing Pairwise Keys in Heterogeneous Sensor Networks. Proceedings of 25th IEEE Conference on Computer Communications (INFOCOM), pp. 52–91, April 2006.

[22] Liu D, Ning P. Location-based Pairwise Key Establishments for Static Sensor Networks. Proceedings of the 1st ACM workshop on Security of Ad hoc and Sensor Networks, pp. 72–82, 2003.

[23] R. Stoleru, H. Wu, and H. Chenji. Secure neighbor discovery and wormhole localization in mobile ad hoc networks. Ad Hoc Networks, vol. 10, no. 7, pp. 1179–1190, 2012.

[24] Du W., Deng J., Han Y.S., Chen S. Varshney P.K. A key management scheme for wireless sensor networks using deployment knowledge,

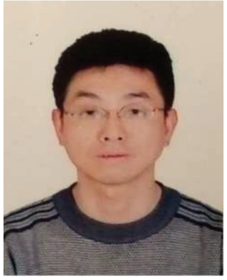
Proceedings of the IEEE INFOCOM. Piscataway: IEEE Press, 2004. 586–597.

[25] Malan DJ, Welsh M, Smith MD. A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography. Proceedings of IEEE Sensor and Ad Hoc Communications and Networks (SECON), pp. 71–80, October. 2004.

[26] Uhsadel L, Poschmann A, Paar C. Enabling Full-Size Public-Key Algorithms on 8-bit Sensor Nodes. Proceedings of Fourth European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS), pp. 73–86, July, 2007.

[27] Cressie N. Statistics for Spatial Data, John Wiley & Sons, 1993.

Biographies



Peng Xiong, received the B.Sc. degree and M.Sc. degree in Electrical Engineering from Nanchang University China in 1998 and 2004, respectively, and Ph.D. degree in Computer science and technology from East China Normal University China in 2009. From 2010 on, he is a faculty member in the school of electronic information, Shanghai Dianji University China. He is a member of China Computer Federation (CCF), and his research is currently focused on network secure, wireless networks, cloud computing and big data etc.

Qinggang Su, received the B.Sc. degree in Computer Science from Anhui University of Technology in 2002, and got the M.Sc. degree in Communication Engineering in Shanghai Jiao Tong University, and is studying for Ph.D. degree at East China Normal University. He became a faculty member in the school of electronic information, Shanghai Dianji University China from 2002, and he is rhe vice dean of Chinesisch-Deutsche Kolleg für Intelligente Produktion of Shanghai Dianji University now. He is a member of China Computer Federation (CCF), and his research is currently focused on wireless networks, 5G application and smart manufacturing.