A Time-based Pre-distribution Shared Key Pool Design

Linfeng Wei
Department of Computer Science, Jinan University, Guangzhou 510632, P.R. China
Email: twei@jnu.edu.cn

Caiting Huang 1,Guoxiang Yao 1, Meixiu Zhou 2, Qingliang Chen 1
1Department of Computer Science, Jinan University, Guangzhou 510632, P.R. China
2China Mobile Guangdong Branch, Guangzhou 510600, P.R. China
Email: xiaok068@163.com

***Abstract*—Based on several core factors of the wireless sensor network security model,a novel group-based key management scheme which contains a pre-shared key pool in order to confirm the network security is proposed. A new architecture of pre-distribution shared key pool filled with time-based hash functions is introduced in this scheme. A time-parameter to the key negotiation and the node's updating process is added by the time-based hash functions. By updating the key with time and detection mechanisms, this shceme produces much better nodes for both forward scecurity and backward security.For the captured problem of the local node,a security event-based rekeying mechanism is designed in this paper. The new one-way hash key chain which is from the density key pool performs robustly in meeting the needs of key generation and update security. The theoretical analysis and simulation are done. The result justifies that the node connectivity, the node storage consumption and the anti-trapping capabilities are improved significantly.**

***Index Terms*—Shared key pool; Time variable; Random pre-distribution;**

I. INTRODUCTION

Wireless Sensor Network (WSN) which is a multi-hop ad hoc network is different from the traditional network. It means the WSN security schemes including the algorithms and the models are also different. In order to build up a secure WSN, these schemes should be addressed well and the efficient key management scheme is part of them, whose design is the core to fulfill the increasing demands of practical applications of WSN [1][2].

Due to the differences of key distribution method, there're 2 wireless sensor key management schemes. One is random key pre-distribution schemes (RKPS) and the other is determined key pre-distribution schemes (DKPS). In the traditional random key pre-distribution scheme, all nodes have to select their pre-shared keys from the key pool before deployment. In determined key pre-distribution schemes, any two nodes get key ring in certain ways.

Eschenauer and Gligor [10] initially proposed the basic random key pre-distribution management scheme which widely impacted other researchers' further study. In the random key pre-distribution scheme, it is easy to distribute keys and there's no restriction on the deployment of nodes. But the keys are calculated by determined probability in the determined key pre-distribution scheme. Because the key generation method is highly targeted, nodes require less storage space to create directly communicate keys. Unfortunately this solution involves huge computing and communication consumption in the key negotiation process, and is inflexible in applications.

Numerous studies have shown [11][12] that the random key pre-distribution scheme is the most suitable for wireless sensor networks, but it also has shortcomings, such as key distribution with blindness and the nodes may store some useless keys which waste storage space. When the deployed nodes are out of battery or are removed from the legitimate node queue, the new nodes are selected from the same key pool [3][9]. This configuration brings the following new problems: Firstly, when captured nodes leak their pre-distribution key and these keys are still in the key pool, the new deployed nodes may select the leaked key into their key groups, which may be cracked by hackers. Secondly, if the leaked keys are still in the key pool, new nodes may select them as their keys. This may let the adversary take the advantage of the leakage to forge new nodes [4].

To solve those problems, a new pre-shared key pool design based on one-way hash key chain by using a group-based key management scheme is proposed in this paper. This scheme not only takes the rekeying security requirements into account, but also prevents the adversary from impersonating new node to make more security attacks when the old nodes establish communication keys with the new ones.

Manuscript received October 30, 2013; revised May 12, 2014; accepted June 6, 2014.
Corresponding author : Caiting Huang
Email: xiaok068@163.com

doi:10.4304/jcp.9.11.2530-2537

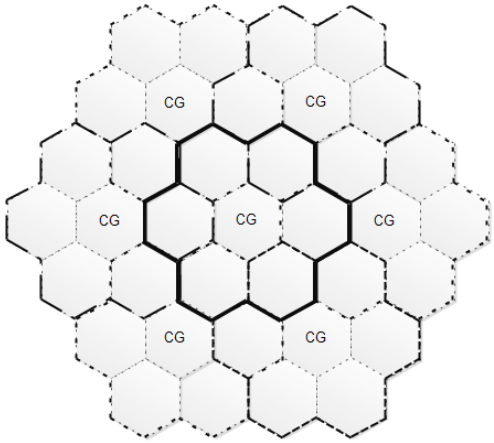

Figure1 The division of sub-group layers.

II. Our Design Of Improved Double-Groups Model

A. Model Assumes

It's assumed that sensor nodes are sown in the target X × Y size area in plane. If the sowing has a center deployment point, most of the sensor nodes are distributed on the deployment area of the center point with a probability distribution function [7]. To simplify the practical problems, two-dimensional Gaussian distribution and uniform deployment is used by most literature schemes. We declare that G_{ij} denotes the deployment area and $S_{G_{ij}}$ denotes the size of the area.

(1) The node's uniform probability model in G_{ij} is:

$$f(x, y \mid k \in G_{ij}) = 1 / S_{G_{ij}}$$

(2) The node's two-dimensional Gaussian distribution model is:

$$f(x, y \mid k \in G_{ij}) = 1/2\pi\sigma^2 \cdot e^{-[(x-X)^2+(y-Y)^2]/2\sigma^2}$$

The formula represents that the location of the k-th sensor nodes around the deployment point following the Gaussian distribution, in which the average is (X, Y), and the standard deviation is σ.

When putting nodes uniformly, the distances of the adjacent deployment points and the distances of the sowing nodes need to be considered carefully. Moreover, when putting nodes in the 2-dimensional Gaussian distribution, the distance d of adjacent nodes and the standard deviation σ of this model are very important parameters. In this scheme, the 2-dimensional Gaussian model is used.

B. Improved Double-Groups Model

In order to improve the node connectivity and to reduce key repeat probability, the deployment area is divided into two grouping layers, including the sub-group layer and the son-group layer. Each hexagonal region is a son group. Each sub-group corresponds to the same size of sub-key pool. All son groups are classified as the Core Group or the Sensor Group. In Figure 1, the marked hexagonal areas are Core Groups, and the unmarked hexagonal areas are Sensor Groups. Core Groups and Sensor Groups share keys in a certain percentage. In order to restrict the longest communication distance among nodes, the nodes can only communicate with the adjacent son groups in our design, which means that the sub-key pools between the neighboring sub-groups share some keys. This design ensures the node's connectivity rate.

Each sub-group includes a Core Group and its six surrounding Sensor Groups. Figure 1 shows that each sub-group overlaps one son group with its adjacent sub-groups. As each son group connects to a sub-key pool, the adjacent sub-key pools contain shared keys. It is designed that non-adjacent sub-key pools won't contain shared keys so that each key is present in only two adjacent sub-groups. If some nodes are captured, it can quickly locate the broken keys and remove them. This method shrinks the captured and affected area which can ensure the node's security in other sub-groups.

One Way Hash Function (OWHF) is the basic design of the pre-shared key pool. Hash function is a typical multi-to-one function whose inputs are the variable-length data (X for short) and a string (h for short) of fixed length (n for short) while h is called an input string of the hash value X [6].

One Way Hash Function can transform input message of arbitrary length into a fixed-length string, and the message is difficult to be obtained from the output string. The output string is called the hash value of the message. A good One Way Hash Function must have the following characteristics [5]:

(1) Speedability : With the known X, it is easy to calculate $Hash(X)$.

(2) One Way: X is difficult to calculate with the known $Hash(X)$.

(3) Collision Resistance: Given $h = Hash(X)$, it is hard to find another message Y to satisfy $h = Hash(Y)$. In addition, it is also hard to find two random messages X and Y to achieve $Hash(X) = Hash(Y)$.

(4) Avalanche effect: each bit of h is associated with every bit of the message X sensitively. That any change of X will have a significant impact on the result h.

Because of these characteristics of One Way Hash Function and its simple calculation, it often combines with key management schemes to improve the security.

III. Our Time-Based Pre-Distributed Shared Key Pool Scheme

A. Key Pool Generation

First, a two-dimensional model is created for the entire region. The horizontal direction is called the X-axis, and the vertical direction is called the Y-axis. Beginning with the origin, the horizontal coordinate of the first row of son groups is $(1, j)$ where $j = 1, 2, 3...$, and the coordinate of the i-th row of sub-groups is (i, j).

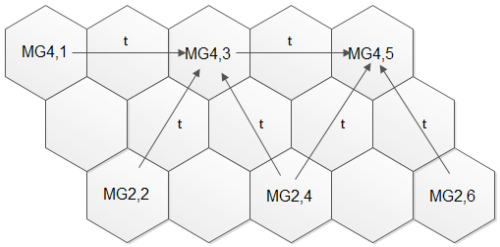

Figure 2. The key sharing relationship between the adjacent sub-groups

The pre-shared key pool is separated into sub-key pools and each sub-group refers to one Core Group. Here, the coordinate of the Core Group is used to represent the sub-group like $MG_{i,j}$. It is assumed that the size of the pre-shared key pool is $|S|$, the size of one sub-key pool is $|S_M|$, and the overlap factor between adjacent sub-key pools is t. The sharing relationship is shown in Figure 2.

Then how to generate the son-key pools in the Sensor Groups is discussed here. Each son group selects its keys both from the son-key pool of its corresponding sub-group and the shared keys of its two adjacent sub-key pools. Supposing that the size of the son-key pool is $|S_C|$ and the overlap factor is b, Figure 3 depicts the key sharing relationship between the Sensor Groups and their sub-groups.

Finally, how the Core Groups get their son-key pools is described here. The keys of each Core Group are selected both from the sub-key pool of the sub-group which contains the Core Group and the son-key pools which are composed of its six adjacent Sensor Groups. It is assumed that the overlap factor of one Core Group and its one adjacent Sensor Group is a.

Other Core Groups also follow this process to generate their key pools. So, it can be obtained that each Core Group shares at least $a|S_C|$ common keys with its adjacent Sensor Groups.

After every sensor node in each region has been deployed completely, nodes establish secure

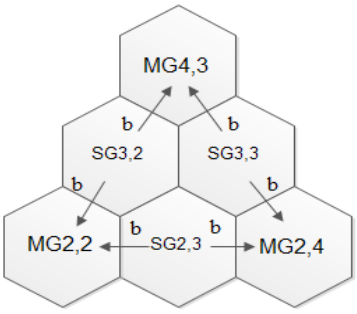

Figure3 The key sharing relationship of the Sensor Groups and their sub-groups

communication key pairs through mutual negotiation including the subsequent node updates, the node joining, and the node deletion.

TABLE I
THE TIME-BASED KEY PRE-DISTRIBUTION HASH CHAIN

$Hash^n(X_1)$	$Hash^{n-1}(X_1)$	$Hash^{n-2}(X_1)$	.....	$Hash^2(X_1)$	$Hash(X_1)$
$Hash^n(X_2)$	$Hash^{n-1}(X_2)$	$Hash^{n-2}(X_2)$	.....	$Hash^2(X_2)$	$Hash(X_2)$
$Hash^n(X_3)$	$Hash^{n-1}(X_3)$	$Hash^{n-2}(X_3)$	.....	$Hash^2(X_3)$	$Hash(X_3)$
⋮	⋮	⋮		⋮	⋮
$Hash^n(X_L)$	$Hash^{n-1}(X_L)$	$Hash^{n-2}(X_L)$	.....	$Hash^2(X_L)$	$Hash(X_L)$
T0	T1	T2		T(n-1)	Tn

B. Time-based Shared Key Pool Design

A new pre-shared key pool design is proposed in this section, which can enhance the security of the key management scheme to compensate for the risks resulting from reducing storage overhead. Supported by this new one-way hash key chain, key pool can not only meet the security requirements of key generation and key update, but also can prevent intruder's security attacks efficiently.

Here, One Way Hash Function is used to design the pre-shared key pool. The design of the key pool based on Time-based one-way hash key chain follows below steps:

(1) Base server randomly generates $|S_C|$ random digit named X_i, and selects the appropriate hash function, like MAC or SHA.

(2) A hash chain is generated. The recursive hash function equation is:

$$Hash^n(X_i) = Hash(Hash^{n-1}(X_i)) \quad (i = 1,2,3...k)$$

Hash (X) is one way hash function, X_i is the random digit of fixed length, L is the total number of keys shared key pool, and n is the number of power of the hash recursive computation.

Putting the random digit into the above formula can get the hash value of the sequence, such as the sequence $Hash(X_i), Hash^2(X_i).....Hash^{n-1}(X_i), Hash^n(X_i)$, while the reverse order is $Hash^n(X_i), Hash^{n-1}(X_i).....Hash^2(X_i), Hash(X_i)$. These constitute a reverse hash chains. And the timeline which is set in the reverse hash chain sequence resulting in pre-distribution key hash chain [8] is shown in Table 1.

(3) The time-based key pre-distribution method: different time points refer to different sets power of the hash recursive computation, like the longitudinal sequence in Table 1. For example, the key period selected by each node in the first deployment is called the T0 time period, and the corresponding key pool is the collection $\{Hash^n(X_1), Hash^n(X_2).....Hash^n(X_x)\}$. During the first deployment period, when a new node is deployed

in the network or the key is compromised and necessitates the update of the key pool, the current pool of shared keys will also change following with the change of the time period. For instance, at the T_k time point, the set of updated key pool is $\{Hash^{n-k}(X_1), Hash^{n-k}(X_2).....Hash^{n-k}(X_x)\}$

In this section, a key pools design of time-based one-way hash key chain is proposed to ensure the security of the pre-shared key in the latest period. On one hand, the affected scope and diffusion time when the nodes are captured is reduced by the design. The keys at different times from different key pools can be got by the nodes so that the node network which has achieved a secure link will not be influenced by the leaked keys. On the other hand, even if some leaked keys in a certain period of time from the hash key chain are mastered by the intruder, the following keys of the key chain cannot be computed because of the one-way property of One-way Hash Function. In addition, the new nodes cannot be impersonated by the intruder. The leaked keys will become invalid when the next update time point comes.

C. Key Negotiation

Direct Key Creation: After the key chain of the shared key pool is generated, each group of the sensor networks gets its corresponding son-key pool. Then each node determines its own regional grouping basing on the deployment information, and randomly selects m one-way hash key chains from its own son-key pool. Finally, it chooses the key element at the T0 time period of the key chain as its key ring. Before the node is deployed, the message is pre-stored in each node's memory including the node identifier, x randomly selected key elements and the key identifiers of the key elements.

Nodes establish secure communication key pairs by mutual negotiation after their deployment. In the process of creating a direct key, the time variable is introduced to solve the pre-shared key leakage.

(1) The Sensor Node i broadcasts all key identifiers in its key ring and the time point T_{i0} at time point T_{i0} to its surrounding neighbors in order to locate all the neighbors who share the common keys with itself.

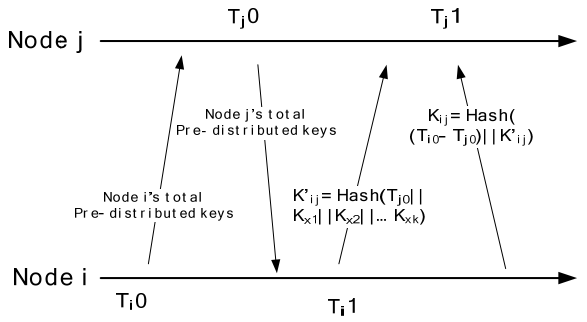

Figure 4. The generation process of a direct communication key

(2) The Sensor Node j notifies its surrounding neighbors including the Sensor Node i, all key identifiers in its key ring and the time point T_{j0} at time point T_{j0}. When Node j receives Node i's broadcast, it establishes its own pre-distributed key table of its neighbor nodes. Assuming Node j finds the same pre-shared keys $K_{x1}, K_{x2}, ..., K_{xk}$, it would calculate the same pre-shared keys and the time parameter $|T_{j0}|$ with one-way hash function that has been selected . The formula is $Hash(T_{j0} || K_{X1} || K_{X2} || \cdots\cdots K_{Xk})$.

(3) If Node i found the same keys $K_{x1}, K_{x2}, ..., K_{xk}$ from Node j's broadcast, it would encrypt the same pre-distributed keys and the time parameter $|T_{j0}|$ using the selected one-way hash function and send them to Node j at time point T_{i1}. The encrypted formula is $K_{ij}^{'} = Hash(T_{j0} || K_{x1} || K_{x2} || .....K_{xk})$.

(4) Node j compares its calculation results with those received from Node i . If they are the same, a secure link can be established. Then Node j computes hash operations using the parameters $|T_{i1} - T_{j0}|$ and K_{ij} . Here $K_{ij} = Hash((T_{i1} - T_{j0}) || K_{ij}^{'})$.

The hash calculation result of the above equation is a direct key between nodes in the communication link. Basic communication interaction is shown in Figure 4.

The time variable is used to generate direct keys in this mechanism, which makes the link keys between two nodes not directly use the same pre-shared keys from the broadcast. The improvement associates with the broadcast information closely and can effectively ensure the uniqueness of nodes' communication keys. Although the node may be captured during the deployment phase, the real communication key with the cracked pre-configured keys cannot be calculated by the intruder.

Indirect Key Creation: Depending on the network topology, those sensor nodes which don't share keys with some of their neighbor nodes cannot establish direct keys through broadcast of the key identifiers. In this case, the nodes must determine the safe paths to reach the neighbor nodes at first. Meanwhile, they negotiate communication keys via the secure paths.

(1) The Sensor Node i belongs to a sub-group. According to the topology, Node i can find the Node j's straight-line intersection along the horizontal direction of the sub-group inclined 60 degrees or 120 degrees. Because two adjacent sub-groups have some shared keys, the safe paths can be established theoretically through son groups.

(2) By the above method the Sensor Node i can connect the Sensor Node j via the path $i \to v_1 \to v_2 \to ..... \to j$, in which any two consecutive sensor nodes have public keys. It is assumed that there are k paths which do not intersect each other.

(3) For example, $i \to v_1 \to v_2 \to ..... \to j$ is one of the paths between Node i and Node j . Assuming that the direct communication key between Node i and Node v_1 is K_{iv1} , the direct communication key between Node v_1 and Node v_2 is K_{v1v2} , and the direct communication key between Node v_n and Node j is K_{vnj} , so the path key of this path $i \to v_1 \to v_2 \to ..... \to j$ is $K_{ij1} = K_{iv1} \oplus K_{v1v2} \oplus ..... \oplus K_{vnj}$.

(4) Then this k path keys are calculated by means of XOR to get final communication key from Nodes i to Node j . At last Node i puts K_{ij} into k parts $K_{ij1}, K_{ij2}, ...K_{ijk}$ and sends them to Node j through this k paths.

(5) After receiving the k fragments, Node j will synthesize $K_{ij1}, K_{ij2}, ...K_{ijk}$ into the communication key K_{ij} .

D. Key Update

The key update can be triggered by two cases: Detecting captured nodes update and Time-based update. If the captured nodes are detected by the intrusion detection system of the wireless sensor network, all of the shared keys stored in them are unreliable. It has to clear those keys timely and to notify other nodes which share the same keys to get rid of them. Then it would arouse key updating operations which include the updates of the shared key pools, the shared keys, and the security node list timely. Since a key pool of time-based One Way Hash Function is designed in our scheme, the key updating mechanism will start automatically when the next time point comes. It would replace each key node to the next key of its reverse hash function chain. Thus, even if some discoverable keys and the hash function in a certain period of time from the hash chain are mastered by the attacker, the subsequent keys after the key update process cannot be derived. Hence, the attacked nodes cannot be disguised and the leaked keys become invalid.

E. New Node Joining

The new node joining uses the same time-based rekeying mechanism. According to the previous section, the key pool based on the node's real-time security detection removes the leaked pre-shared key chain timely. This mechanism can ensure the key pool only contains secure key chains and the newly added node can get the key ring from key chains never leaked. In addition, in accordance with Section 3.3, the timeline is set on wireless sensor network status in this mechanism. With the new nodes added into sensor network in different time periods in succession, the deployment nodes of different time periods will correspond to different node pools to ensure the security and the reliability.

As the newly added nodes have identified its deployment group, a random pre-distributed key from its own son-key pool corresponding to its group is directly selected for them. In accordance with the description of how to establish communication keys in Section 3.4, the configured pre-shared key rings and the old nodes' key rings pass the verification of hash function calculation. Moreover, this process ensures the safety of the new node's identity and completes the shared key security upgrades.

IV. Efficiency Analysis

In this section, the performance of our scheme will be analyzed from four perspectives: the anti-trapping ability, the memory consumption, the communication consumption and the expandability.

A. Anti-capture Ability Analysis

Here, supposing that there are T captured nodes, the anti-capture ability is based on the probability of the communication link leakage resulting from the T captured nodes in the sensor network. In our pre-distribution key management scheme, the keys in the pre-shared key pools are generated by one-way hash function and will be updated in the interval of time after the key negotiation. Hence, the nodes' anti-capture ability will be discussed in two cases: the capture after the node deployment and the capture after the key negotiation.

(1) If the sensor node is captured in the initialization phase, the calculation process is similar to the original classic regional grouping of regular hexagon. The size of son key pool is $|S_C|$, the size of the key ring is m, and the common communication key between two sensor nodes is K . Name the probability breaking keys through the captured nodes be P_{key} . So when n nodes are captured, the unbroken probability of the keys is $(1-\frac{m}{|S_C|}P_{key})^n$. According to our scheme, while the adjacent nodes have k shared keys, all of these keys will be hashed to be the communication key.

Now, the probability to break the communication key is: $(1-(1-\frac{m}{|S_c|}P_{key})^n)^k$.

For the same group, the probability that two nodes have k common keys is: $Pk = \frac{C_{|S_C|}^k C_{|S_C|-k}^{m-k} C_{|S_C|-m}^{m-k}}{(C_{|S_C|}^m)^2}$

From the above equation, after n nodes are captured, the probability of leading the other nodes to be cracked is: $P = \sum_{k=1}^{m}(1-(1-\frac{m}{|S_c|}P_{key})^n)^k \times P_k$

Given the size of key pool of base station is 70000, the regional grouping is 20*21, and the probability P_{key} to break keys when nodes are captured is 1, the value of P according to the size m of the different key ring can be calculated. The relationship between the number of

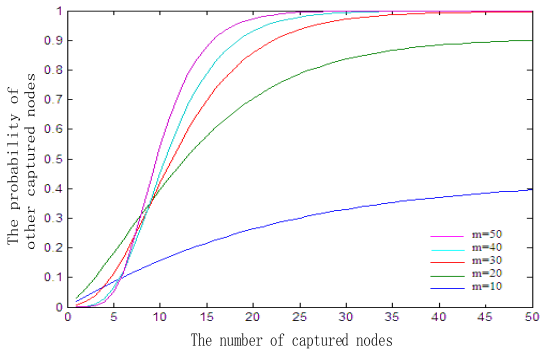

Figure 5. The node anti-capture ability according to different sizes of key pools

captured nodes and the node anti-capture ability is shown in Figure 5.

Figure 5 shows that with the increases of the key ring, the overall nodes' anti-capture ability is becoming weaker. But due to the better connectivity of our scheme, the nodes needn't to store too many keys, and the size of the key ring can be narrowed. And because the common keys exist only in two adjacent sub-groups, the impact can be reduced in two sub-groups even if nodes are captured. So the leaked keys can be quickly located and removed.

(2) If the sensor nodes are captured after the communication key negotiation, what the nodes hold is those keys calculated by the hash function rather than the original ones in the pre-shared key pool. From the previous sections, we know that the advantage of one-way hash function is fast, one-way, collision resistance and so on. Even if the sensor nodes are captured, the attacker can only get the current key information from trapped nodes but not any original keys. Moreover, the time variable is used in the direct key establishment phase, so the direct keys are not the original hash keys, which can protect other communication links which are established with the same direct keys even if the attackers get the captured nodes' keys.

Meanwhile, this time-based pre-shared key pool management can dynamically manage key information to confirm the network security further when the individual node is damaged. It also can timely exclude the trapped nodes' keys out of the network to protect others. After the keys are updated, the current captured nodes' keys have been replaced, and the leaked keys become invalid. So the security of network can still be maintained.

TABLE II
THE NODE CONNECTIVITY RATE

	Node Connectivity Rate			
The relationship of communication radius and regional radius	**0.3**	**0.5**	**0.7**	**0.9**
r=0.5D	9	13	17	22
r=D	12	16	21	29
r=2D	21	30	40	54

Therefore, if the sensor nodes are captured after the communication key negotiation, it will affect nothing and the probability of other nodes to be captured is always 0.

B. Storage Consumption Analysis

Storage consumption analysis is to compare the size m of the node key rings in the same connection probability. The size of the key pool of the base station is 70000 and the regional grouping is 20*21. According to the son-key pool allocation algorithm and the analysis introduced in Section 4.1 and Section 4.2, the necessary size of the key ring in the same connection probability P_{local} is shown in Table 2.

The result from Table 2 shows that sensor nodes only need to store a few keys. Therefore the scheme can save the storage space significantly.

C. Communication Consumption Analysis

For the communication consumption, only a collection of key identifiers of the key rings and the data packets of the node identifiers needs to be sent by a single node A in the key negotiation phase. After the neighbor nodes receive the packets, the set of key identifier from the packets is compared with its own set. If A's neighbor nodes find the same key identifiers, their own set of key identifiers of the key rings and the data packets of the node identifiers will be sent back to A. Finally, one of these common elements is taken by A as the communication key. In this phase, the keys do not need to be computed. And only a small number of data packets are transmitted between the nodes and their neighbors.

However, in the key negotiation phase, though the same keys located on the key chain, the hash function $H(x)$ computation needs to be passed. The hash average is $L/2$ (where L is the length of the key chain). The node key ring in our scheme is the same as those in the basic random pre-distribution key scheme, but a single node requires $L/2$ times hash computation on average. When the key negotiation phase completed, A needs to change the original keys stored in the key rings, and each key K would be calculated via hash function, such as $H(K)$. The computational complexity is m. Therefore, the whole scheme has to calculate $(m+L/2)$ times and the computational complexity is $O(n)$ while the original scheme does not have this calculation consumption.

Since the time variable is involved in our scheme, the communication consumption still includes a communication process of adding time variable and a hash function calculation. Although the energy cost of nodes increases, nodes only need to store a small number of key rings. The computational complexity grows linearly with the number of the key rings. Due to fewer key rings in out scheme, the computational complexity is also reduced. Because the nodes' communication consumption is mainly in the data transmission after the network initialization, in comparison to it, the initialization overhead of this scheme can be fully

accepted. And the security of communications between keys is significantly enhanced by just increasing slight overhead in the initialization.

D. Expandability Analysis

Expandability means that the key management scheme can adapt to different wireless sensor networks. Although the network expansion of original scheme is simple and quick with less overhead, its flaws are also obvious: (i) The key pool cannot remove and update the leaked keys from the captured nodes; (ii) New nodes select their key rings from the same pre-shared key pool with old nodes; (iii) Those threatening nodes are not updated timely, which may make the network unsafe. If new nodes select the leaked keys and then join the network, they can be potential danger of further attack when communicating with other nodes.

Our scheme is distributed and no longer relies on base station after initialization is completed. Furthermore, it is able to adapt to different sizes of wireless sensor networks, and even if the network size increases, the storage and communication overhead of each node's will not change.

In addition, the analysis of the new node joining algorithm shows that although the improvement adds new nodes through the rekeying mechanism which needs to increase certain computational overhead, this pre-shared key pool design based on one-way hash key chain ensures the reliability to configure pre-shared keys and guarantees forward security. If deployed nodes are captured and the leakage information of the pre-distributed keys are not detected in time, the introduction of the time-based key update mechanism can ensure that the newly added nodes get their keys from the corresponding pre-shared key pool at a new time point. So the newly added nodes will not select the leaked pre-shared keys. Nodes can easily join the network securely at any time. Hence, our scheme fully supports the network expandability.

E. Comparison

Different key management schemes have their own characteristics to adapt specific application needs. Each of them seeks compromise and balance among efficiency, security and energy consumption. The comparison of our scheme and others is show in the Table III.

In terms of security, both the Scheme [10] and the Scheme [11] are good. However, their storage consumptions are large. For Scheme [3], the auxiliary storage consumption is reduced at the expense of the connectivity security risks. Kindly, The Scheme [13] achieves high connectivity and scalability while increasing the computation and communication overhead. In our scheme, a time variable is added to ensure the security which just increases the computation overhead in the nodes' deployment slightly. Theory and research indicate that to design a perfect solution with high efficiency, strong security and low power consumption is not realistic. The proposed scheme with the double-groups model is more dominant than the Scheme [3] in terms of connectivity, scalability and storage consumption.

TABLE III
THE COMPARISON OF SCHEMES

	Advantage	Disadvantage
Scheme[10]	Low key storage overhead	Weak connectivity, communication island exists.
Scheme[11]	Enhances anti-destroying ability to some extent	Needs more storage space and anti-destroying ability decreases rapidly with the number of damaged nodes.
Scheme[3]	The auxiliary storage consumption gets lower and the security grows higher when the size of key pool is larger.	Ordinary nodes will not be able to get keys if all the auxiliary nodes are damaged.
Scheme[13]	Low key storage overhead, high connectivity and high scalability	High computation and communication consumption
Our scheme	Low communication consumption, high connectivity and high expandability	Computation consumption slightly increases when the nodes are deployed.

V. CONCLUSION

The core issues of wireless sensor network key management are studied in this paper. A time-based design of pre-distributed shared key pool is proposed, and the practicality, security, scalability, energy and storage consumption, and other communication aspects of the scheme are analyzed with testing demonstration. A one-way hash function is used by the scheme to protect the security of the key chains. Because of the one-way property of one-way hash function, the latter keys of the key chain cannot be computed even if the adversary masters some leaked keys in a certain period of time from the hash key chain. So the scheme can prevent any attacker from disguising the new deployed node. All the other properties of one-way hash function also ensure the forward security of the pre-shared keys in the wireless sensor network.

In addition, the scheme is based on time-based key chain and the key generation, key negotiation, and key update are added into the key pool application. These approaches not only effectively prevent attackers from masquerading new deployed nodes when communication keys are established, but also promptly remove the unreliable keys to avoid them disguising as the old nodes, which ensures the backward security of the wireless sensor network. The experimental result tells that although the consumption of computing and communication by using the key chain is increased slightly in this design, the security and scalability of the link are enhanced remarkably.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (Grant No. 61272415, 61272413, 61133014), the Science Program of Guangdong Province, China (Grant No. 2012A080102007, 2012B091000136, 2012B091000038)

and the Engineering Research Center Program of Guangdong Province, China (Grant No. GCZX-A1103).

CONFLICT OF INTEREST

The authors do not have any conflict of interest with the content of the manuscript.

REFERENCES

[1] LI Min, YIN Jianping,WU Yongan, CHENG Jieren. A Survey on the Key Management Schemes for Wireless Sensor Networks [J]. Computer Engineering & Science. 2008, 12.

[2] MI Bo, CAO Jianqiu, DUAN Shukai et al.Survey on key management of wireless sensor networks.Computer Engineering and Applications, 2011, 47(13):77-82..

[3] SU Zhong LIN Chuang REN Feng-Yuan. Hash Chain Based Random Keys Pre-Distribution Scheme in Wireless Sensor Networks [J]. Chinese Journal of Computers, 2009, 32(1): 30-41.

[4] WANG Chao, HU Guang-yue, ZHANG Huan-guo. Lightweight security architecture design for wireless sensor network [J]. Journal on Communications. 2012, 33(2): 30-35.

[5] ZHANG Ju-wei, SUN Yu-geng .KPSBM: pairwise key pr e - distr ibution scheme for wir eless sensor networks based on deployment knowledge.Computer Engineer ing and Applications, 2008, 44(20) : 4- 6.

[6] FANG Wang-sheng, ZHANG Tao, CHEN Kang. Key Distribution Scheme for WSN Based on Hash Function [J]. Computer Engineering, 2010, 36(11): 161-163.

[7] HAN Xinhui , LONG Qin , SI Duanfeng et al. A Practical Hierarchical Key Management Scheme Based on One-Way Hash Function [J]. Acta Scientiarum Naturalium Universitatis Pekinensis, 2008, 44(4): 527-536.

[8] Yuan T, Ma JQ, Zhong YP, Zhang SY. Key management scheme using time-based deployment for wireless sensor networks. Journal of Software, 2010,21(3):516−527.

[9] Guoxiang YAO, Saizhi YE, Caiting HUANG. A Key Management Algorithm Based on Two-tier Architecture in Wireless Ad Hoc Network[J]. Journal of Networks, 2013, 8(6): 1241-1247.

[10] Eschenauer L, Gligor V D. A key-management scheme for distributed sensor networks[C]//Proceedings of the 9th ACM conference on Computer and communications security. ACM, 2002: 41-47.

[11] Chan H, Perrig A, Song D. Random key predistribution schemes for sensor networks[C]//Security and Privacy, 2003. Proceedings. 2003 Symposium on. IEEE, 2003: 197-213.

[12] Du W, Deng J, Han Y S, et al. A key management scheme for wireless sensor networks using deployment knowledge[C]//INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies. IEEE, 2004, 1.

[13] Bechkit W, Challal Y, Bouabdallah A, et al. A highly scalable key pre-distribution scheme for wireless sensor networks[J]. IEEE Transactions on Wireless Communications, 2013, 12(2): 948-959.

[14] Yao G, Guan Q, Ni K. Test Model for Security Vulnerability in Web Controls based on Fuzzing[J]. Journal of Software (1796217X), 2012, 7(4).

[15] Chen Z, Ling W, Zhao J, et al. Consonant recognition of dysarthria based on wavelet transform and fuzzy support vector machines[J]. Journal of Software, 2011, 6(5): 887-893.

[16] HE Z. Accelerometer based gesture recognition using fusion features and SVM[J]. Journal of Software, 2011, 6(6): 1042-1049