2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing

A Key Management Protocol for Multiphase Hierarchical Wireless Sensor Networks

Biming Tian, Song Han, Sazia Parvin, Tharam S. Dillon
DEBI Institute, Curtin University
Enterprise Unit 4, De Laeter Way, Technology Park
Bentley, Perth, Australia 6102
Email: {biming.tian, sazia.parvin }@postgrad.curtin.edu.au
{song.han, tharam.dillon}@cbs.curtin.edu.au

***Abstract*—The security of Wireless Sensor Networks (WSNs) has a direct reliance on secure and efficient key management. This leaves key management as a fundamental research topic in the field of WSNs security. Among the proposed key management schemes for WSNs security, LEAP (Localized Encryption and Authentication Protocol) has been regarded as an efficient protocol over the last years. LEAP supports the establishment of four types of keys. The security of these keys is under the assumption that the initial deployment phase is secure and the initial key is erased from sensor nodes after the initialization phase. However, the initial key is used again for node addition after the initialization phase whereas the new node can be compromised before erasing the key. A time-based key management scheme rethought the security of LEAP. We show the deficiency of the time-based key management scheme and proposed a key management scheme for multi-phase WSNs in this paper. The proposed scheme disperses the damage resulting from the disclosure of the initial key. We show it has better resilience and higher key connectivity probability through the analysis.**

***Keywords*-Key management, key predistribution, Wireless Sensor Networks(WSNs);**

I. Introduction

Wireless Sensor Networks (WSNs) have gained wide applications ranging from civilian to military use. A typical WSN is composed of a great number of sensor nodes. These sensor nodes have limited battery power, weak data processing capability and short radio range. Most importantly, sensor nodes are often randomly spread out over specific regions and work in unattended environment. They are prone to all kinds of attacks thus security becomes the first concern. In order to keep communication secure, sensitive data should be encrypted and authenticated. Therefore, key management, which is a prerequisite of encryption and authentication, should be addressed carefully.

Among all the key management mechanisms for WSNs, key pre-distribution mechanism provides a nice tradeoff between storage overhead and processing power, and is considered as the most suitable mechanism for WSNs. However, most of key predistribution schemes consider a homogeneous topology and only support the establishment of pairwise keys. Even though homogeneous networks are simple and efficient for small network scale, such networks lack scalability due to "one-affect-n" effect in node addition and revocation. In a hierarchical WSN, the effect of node addition and revocation can be localized into a cluster thus scalability is achieved.

In a hierarchical WSN (HWSN), various types of communication may happen. The base station broadcasts control commands to the whole network. Control node multicasts messages within the cluster. A node communicates with its neighboring nodes by unicasting. Therefore, network-wide key, cluster key, and pairwise key are required to satisfy different types of secure communication. Zhu et al. [1] devised a scheme called localized encryption and authentication protocol (LEAP) for hierarchical WSNs. LEAP supports establishment of individual keys, pairwise keys, cluster keys, and a global key. Different keys are used to handle the different types of packets. The security of all types of keys relys on that of the initial key. As many existing key management protocols, LEAP assumes that the initial key is secure during the initialization phase and is erased from the memory of sensor nodes when the initialization phase finishes. The authors regarded the scheme is secure under such an assumption. However, the same key should be used again for node addition and replacement. According to the assumption, some new nodes may be captured at any time after the initialization phase. That is, the new deployed nodes could be captured before removing the initialization key. The security of the scheme is threatened by the attacks launched after the initialization phase.

Jang et al. [2] improved LEAP by introducing a time-based key management protocol. The scheme strengthens the security with a new notion of probabilistic time intervals. However, the scheme does not guarantee the perfect key connectivity. In addition, the pairwise key does not exclusively belong to the two end nodes. Those nodes which have the same initial key or master key can calculate the other nodes' pairwise keys as the ID of each node is public. To address these security and performance issues, we present an elegant key management scheme in this paper.

This paper is organized as follows: We describe the LEAP protocol [1] and the time-based key management scheme in [2] and then talk about the security problems of them in Section II. We propose a key management protocol for multiphase hierarchical WSNs in Section III. We analyze its

978-0-7695-4322-2/10 $26.00 © 2010 IEEE
DOI 10.1109/EUC.2010.99

performance and security in Section IV and conclude this paper in Section V.

II. Related Works

A. Key Predistribution Schemes

Eschenauer et al. proposed the pioneering work [3]. It is a random key predistribution scheme. Initially a large key pool of P symmetric keys and their identities are generated. Each sensor randomly draws $k(k \ll P)$ keys from the key pool without replacement. These k keys and their identities form the key-chain for a sensor node. In the shared-key discovery phase, two neighbor nodes exchange and compare the list of identities of keys in their key-chains. If two sensors have at least one key in common, they can setup a secure link directly. Otherwise, the path-key establishment procedure is triggered to setup a link between two neighbors. The nodes still can establish a secure channel under the help of one or more intermediate nodes. The advantage of the random key predistribution scheme is that there is no computational overhead to generate pairwise keys between sensor nodes. The main shortcoming of the scheme in [3] is that a large number of keys could be disclosed by compromising a few nodes. The basic scheme in [3] was further improved by Chan et al. in [4] from two different aspects. Two variations are proposed: the $q(q \geq 2)$-composite scheme and the multi-path key reinforcement scheme. The variations make it more difficult to compromise a node.

The schemes [3] and [4] share a common shortcoming. That is, they only explore the way to establish the pairwise keys between nodes but not other types of keys. Obviously, it is not enough for different communication manners in HWSNs. In addition, both of [3] and [4] allow dynamic addition, however, their key pools do not evolve with time. As a result, if the network encounters a long-term attack, the newly deployed nodes may be preloaded with some already compromised keys. It is possible to discover all the keys in the key pool if an attacker continues his/her attack. One naive countermeasure is to refresh key pool when some nodes are added to the network. The attacker cannot deduce the key ring of the newly deployed nodes with the knowledge of the key materials of the compromised nodes. However, this renewal introduces unexpected consequences. That is, sensor nodes deployed at different time slots cannot establish pairwise keys because they do not have the common initial key. How to achieve connectivity between nodes deployed at different time slots in a dynamically renewed key pool is an open problem to be answered.

To the best of our knowledge, RoK [8] is the first key predistribution scheme adapted to multiphase WSNs. In this scheme, sensor nodes which run out of power will be removed from the network and new sensor nodes need to be periodically deployed to assure network connectivity. Correspondingly, the predistributed keys have limited lifetimes and the key pool should be refreshed periodically. This scheme overcomes the drawback of the general key predistribution schemes. [3] [4]. The security of the network does not degrade with time. Zo-RoK [9] takes advantage of prior deployment knowledge in order to reduce the size of key ring. In this way, the resiliency of the network against node capture attacks increases with a smaller key ring of each node. However, deployment knowledge is not always available in WSNs. Lately, a random generation material (RGM) key predistribution scheme was proposed in [5]. In this scheme, the lifetime of the whole network is divided into generations. Each generation has its own random keying material and pairwise keys are established by two nodes is only known by the nodes in two generations which the two nodes belong to. Nodes deployed in other generations other than the two generations which the nodes belong to have no access to the pairwise key. We lend this idea to improve the performance of the LEAP scheme in this paper.

B. LEAP

Zhu et al. [1] devised a key management scheme which is abbreviated as LEAP for hierarchical WSNs. LEAP offers establishment methods of individual keys, pairwise keys, cluster keys, and a global key. Different keys are used to handle the different types of packets. The establishment of cluster keys and the global key mainly depends on the the established pairwise keys, so we omit the establishment of them here and focus on the description of the establishment of pairwise keys.

- Individual key: This is an unique key that is shared between the base station and each sensor node [1]. The key is preloaded into each node's memory before being deployed. The individual key is calculated as $K_u^m = f_{K^m}(u)$ where f is a pseudo-random function and K^m is the system key known only to the base station and u represents the ID of the node u.
- Pairwise key: Each node shares a pairwise key with each of its immediate neighbors. Similar to the scheme in [3], there are four stages of pairwise key establishment: key predistribution, neighbor discovery, pairwise key establishment, and key erasure. During the initial stage of key predistribution, node u is loaded with a initial key IK by the controller and drives the master key $K_u = f_{IK}(u)$. For neighbor discovery, node u first initialize a timer to activate during a time slot of T_{min}, then it broadcasts a HELLO message containing its ID to discover its neighbors. The neighboring node v responds to node u with an acknowledgement (ACK) message containing its ID if it receives node u's HELLO message. The ACK message of v is authenticated using its master key K_v which is derived from IK. Node u verifies the Message Authentication Code (MAC) of v by generating the master key K_v with IK. The neighbor discovery stage can be denoted as:

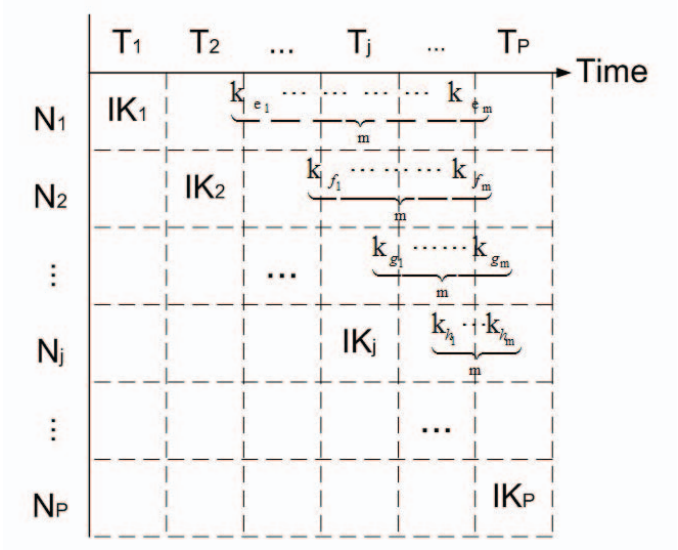

Figure 1. Key materials preloaded to nodes at different time slots in scheme [2].

$$u \rightarrow * : u;$$

$$v \rightarrow u : v, MAC(K_v, u \mid v).$$

In the stage of pairwise key establishment, node u calculates the pairwise key K_{uv} shared with node v, as $K_{uv} = f_{K_v}(u)$. Node v can also derive K_{uv} in the same way. K_{uv} serves as their pairwise key. In the final stage, when its timer expires after T_{min}, node u erases IK and all the masters keys of its neighbors, which it computed in the neighbor discovery stage. Even though an adversary captures a node, the communication between the captured node and another node cannot be decrypted without the key IK.

C. The Time-based Key Management Scheme

With the motivation of minimizing the portion of compromised network when the initial key IK is disclosed, Jang et al. split the lifetime of a sensor network into P time slots and each time slot is assigned with an initial key. As depicted in Figure 1, T_j and N_j represent a time slot and a group of node deployed during that time slot T_j, respectively. If a node will be deployed at time slot T_j, the sensor node is preloaded with the initial key IK_j and m master keys of randomly-chosen time slots. Then the newly deployed node can establish pairwise keys with nodes which are deployed at same or different time slots. Three situations exist for the establishment of pairwise keys.

1) All nodes in the same group $N_j (1 \leq j \leq P)$ are able to establish pairwise keys with each other using the initial key IK_j during the time slot T_j.
2) Then, they are able to establish pairwise keys with other nodes which are deployed at different time slots, but have the master key derived from the current initial key. Suppose u is a node deployed at time slot T_j and v is a node deployed before T_j. If the node v has the master key K_{vj} which is derived from the initial key IK_j for time slot T_j, the node v can compute a pairwise key $K_{uv} = f_{K_{vj}}(u)$. The node u is also able to generate a master key of v, $K_{vj} = f_{IK_j}(v)$.
3) Finally, a pair of sensor nodes that do not share any keying material but are in wireless communication range can establish pairwise keys via proxy nodes.

D. The Security Problems of LEAP and The Time-based Key Management Scheme

As many existing key management protocols, LEAP assumes that sensor nodes are secure during the initialization phase and can be compromised after the phase. However, such an assumption could be incorrect. Security of LEAP mainly depends upon the initial key which is erased from sensor nodes after the initialization phase. However, the same initial key IK should be used again for node addition after that phase while the new node can be captured before removing the initial key. Therefore, the initial key IK should never be used for node addition in LEAP after the initial time T_{min}. Different initial keys are used for different time slots in the time-based key management scheme [2]. The threat caused by the disclosure of the initial key is eliminated. However, the key connectivity is constrained by the number of preloaded master keys m and the order of the current time slot. If m is far less than the lifetime P of the network, the key connectivity $\frac{m}{P-i}$ is far less than 1 at the time slots $j (j \leq P/2)$. On the contrary, if m is close to P, higher key connectivity can be achieved with heavy burden on storage. We consider the security problem of the established pairwise key between two nodes. The pairwise key does not exclusively belong to the two end nodes. As shown in Figure 2 in [2], Nodes of group N_1, N_2, and N_6 are preloaded with master key K_{u7}, the pairwise keys between any two groups of them are known by the other group. In addition, m master keys of randomly-chosen time slots are preloaded to the nodes when they are deployed to the network without taking the lifetime of nodes into consideration. Suppose a node who can survive at most G_w time slots is deployed at the j-th time slot with m master keys of randomly-chosen time slots. Those master keys of the time slots from $(j+G_w)$-th to P-th would never be used. They waste scarce memory of sensor nodes.

III. Our Construction

Motivated by the random key predistribution scheme in [5], we propose a novel key management for multi-phase hierarchical WSN. In this scheme, the time domain of the network is split into many time slots. In this paper, we

call the time slot generation as well. It is assumed that there are totally P generations. Sensor nodes are usually powered by battery. It is assumed that a node may live at most for G_w generations. Each generation has its initial key which is constructed by a key distribution center. There is no relation between the initial keys for different time slots. This property prevent the attackers from concluding the previous and future initial keys. Wireless sensor network are set up for longer lifetime as compared to that of sensor nodes. Therefore, new nodes need to be replenished in some generations in the case of node capture attack or depletion of battery to provide continuity of network. It is supposed that an attacker can get all the key materials stored in the captured node. In order to achieve connectivity between nodes belonging to different generations and resiliency, the keys that are used to establish pairwise keys should evolve in a different way, independent of evolution of the initial keys. Table I presents the symbols used in our proposed key management scheme.

Table I
SYMBOLS USED IN THE PROPOSED KEY MANAGEMENT SCHEME

P	The initial key pool size and the whole life time of the network
KR^j	The key ring of nodes deployed at generation j
K_{uv}^j	Pairwise key between nodes u and v which deployed at generation j
K_{uv}^{gh}	Pairwise key between nodes u which deployed at generation g and v which deployed at generation v where $1 \leq g < h \leq g + G_w - 1$
$H(\cdot)$	Secure hash function

A. Predistribution of Key Materials

In the proposed scheme, as depicted in Figure 2, the group G_j deployed at generation T_j are assigned with an initial key IK_j and $G_w - 1$ master keys in order to establish links with nodes deployed at the same or different generations. The initial key is reserved for the establishment of secure communication with nodes deployed in the same generation. The other $G_w - 1$ master keys are used to establish secure links with the groups deployed at subsequent generations. Different from the time-based key management scheme in [2], the master keys in our scheme are constructed in an ingenious way. These master keys are transformed from the initial keys for the generations within the the node's generation window G_w. For a group deployed at generation j, their sub-keyring $S - KR^j$ containing the master keys for the subsequent generations $j + 1 \leq i < j + G_w - 1$, is given as follows:

$$S - KR^j = \{K_{j,i} | K_{j,i} = H(IK_i \parallel j)\}, \qquad (1)$$

where IK_i is the initial key for the i-th generation.

"This process " can be detailed as follows. In order to form the sub-keyring $S-KR^j$, the key distributor first picks up G_w-1 initial keys for the subsequent G_w-1 generations. Each of these keys is appended with the generation number of the group, which is j, and hashed using a secure hash function like SHA-1 [6] or SHA-256 [7], depending on the key size. These hashed values are stored in the sub-keyring. In this way, we customize the keys belonging to a subsequent generation to be used in another generation without storing the actual initial keys, owing to the one-way property of the secure hash functions.

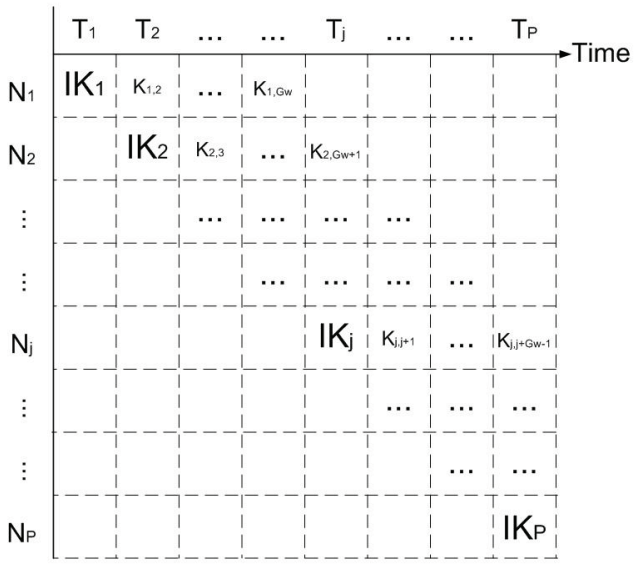

Figure 2. The key rings preloaded to nodes at different generations.

To sum up, the keyring of a sensor node contains (1) the initial key assigned to the current generation; (2) the transformed master keys for up to G_w-1 subsequent generations. More formally, for a node deployed at generation j, its key ring KR^j is shown as follows:

$$KR^j = \{IK_j, K_{j,j+1}, \ldots, K_{j,j+G_w-1}\}. \qquad (2)$$

That is, each sensor node stores one initial key and G_w-1 master keys for each upcoming generation. Because a sensor node may communicate with nodes at most $G_w - 1$ next generation, the maximum number of keys in the key ring of a particular node is G_w.

B. Key Establishment

After the keyring is created, the nodes are deployed over the sensor field. Two situations exist for the establishment of pairwise keys.

1) Since all sensor nodes deployed at generation j contain the initial key IK_j, they can establish pairwise keys using IK_j. Suppose two nodes u and v belong to the same generation j, they compute their pairwise keys as follows:

- After a node u computes a master key $K_u^j = f_{IK_j}(u)$, node u broadcasts a HELLO message with its ID and generation j and then waits for a response from the neighboring node v which is deployed in the same generation. Node v sends node u a response message including its ID and MAC.

$$u \rightarrow * : u, j, nonce;$$

$$v \rightarrow u : v, MAC(K_v^j, u \mid v).$$

- Both u and v can compute a pairwise key $K_{uv}^j = f_{K_v^j}(u) = f_{K_u^j}(v)$.

2) If the two nodes belong to different generations, the pairwise key creation process is different. Let us suppose that the node u deployed at generation g and another node v deployed at generation $h(1 \geq g < h \leq g + G_w - 1)$. They computer their pairwise keys as follows:

$$K_{uv}^{gh} = f_{K_v^{gh}}(u) = f_{K_u^{gh}}(v)$$

where $K_v^{gh} = f_{K_{gh}}(v)$ and $K_u^{gh} = f_{K_{gh}}(u)$. Node u is already preloaded with the key K_{gh} before deployment. By using this key, node u can calculate K_u^{gh}. However, from the point of view of node v, the master key K_{gh} is the master key for the previous generation g. Therefore, it is not in node v's keyring. Fortunately, node v can calculate K_{gh}. As discussed in the previous subsection, $K_{gh} = H(IK_h \parallel g)$, where H is a secure one-way hash function. Node v is deployed at generation h, and therefore it stores the initial key IK_h. By using IK_h, it calculates the key K_v^{gh} and then calculates K_{uv}^{gh}.

IV. Security and Performance Analysis

A. Security Analysis

The goal of this section is to evaluate the security of our proposal and compare it with that of the time-based key management scheme proposed in [2].

We suppose that when a node is compromised, the key material stored in the node will be extracted by the adversary. The key material will be utilized to attack the rest of network. In [2], the resilience of schemes is described as that the additional portion of network that an adversary can compromise using the key material obtained from x compromised nodes. We still use this definition in this section. The security of the LEAP scheme depends on the security of the initial key IK. The whole network can be compromised once the initial key K_I is disclosed. The damage resulting from a disclosure of an initial key IK is localized by the time-based key management scheme [2]. In order to provide connectivity between nodes deployed at different generations, the preloaded master key for different generation is the same. An compromised initial key IK_a at generation T_a only affect the nodes deployed at generation T_a rather than the whole network. However, as we mentioned in Section II, the pairwise key does not exclusively belong to the two end nodes. If three nodes are preloaded with the same master key, the pairwise keys between any two groups of them are known by the other group. Once a node is captured, the pairwise keys shared by other two nodes will be compromised as well.

In our scheme, the pairwise key K_{uv}^{gh} exclusively belongs to the two end nodes. For example, the pairwise key used between node u deployed at generation g and v deployed at generation h is confined to the two end nodes deployed at these two generations. Nodes deployed at generations other than g and h have no access to this key. This is because the master key K_{gh} can be computed by a node if and only if this node has been deployed at generation h and has an initial key IK_h in its key ring. As a result, a sensor node w, which is deployed at any other generation l cannot compute a master key K_{gh}. Three conditions exist according to the value of l

- $l < h$. The node w needs IK_h to compute K_{gh}. Even though the node w has the master key $K_{lh} = H(IK_h \parallel l)$, it cannot derive IK_h from K_{lh} due to the one-way property of the secure hash function H.
- $h < l \leq g + G_w - 1$. The node u is preloaded with the master key $K_{gl} = H(IK_l \parallel g)$ and the node v is preloaded with the master key $K_{hl} = H(IK_l \parallel l)$. Even the node w can calculate K_{gl} and K_{hl}, it cannot derive IK_h or K_{gh} due to the one-way property of the secure hash function H.
- $l > g + G_w - 1$. The node u is power off at the generation l.

It is clear that a master key K_{gh} is known only by the nodes of the generation g and h. No node deployed at any other generation can compute the key that is unique to the generation g and h. Hence an attacker has to spend extra effort if s/he wants to acquire the pairwise key between the nodes u and v that are deployed at the generations g and h, respectively. This has an advantage over the scheme in [2] in restricting the information that an attack acquires if s/he captures a node.

B. Performance Analysis

Key Connectivity. The key connectivity of a group N_t which is deployed at generation t of the time-based key management scheme [2] is assessed from two aspects. One is N_t's probability of sharing keying materials with prospective sensor nodes, $p_{pros}(t)$ and the other is the probability of sharing keying materials with predeployed sensor nodes and nodes deployed at the current generation G_t, $p_{pre}(t)$. According to the scheme, when a sensor node is deployed at generation t, only the predeployed nodes which have the master key, which is derived from the initial key IK_t of the generation t, can establish pairwise key with it. Because each node is preloaded with the initial key of the

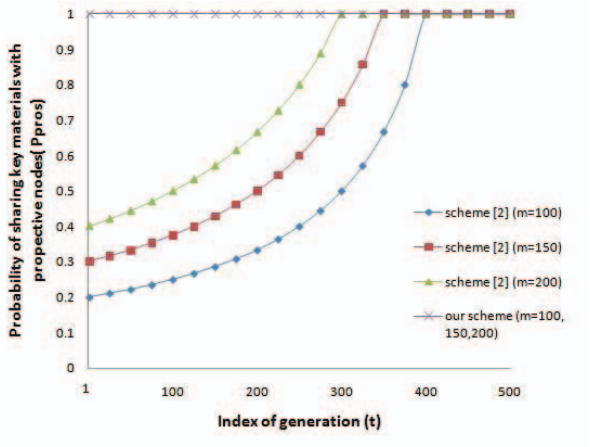

Figure 3. The comparison of the probabilities that N_t establishes pairwise key with prospective groups in our scheme and the scheme in [2].

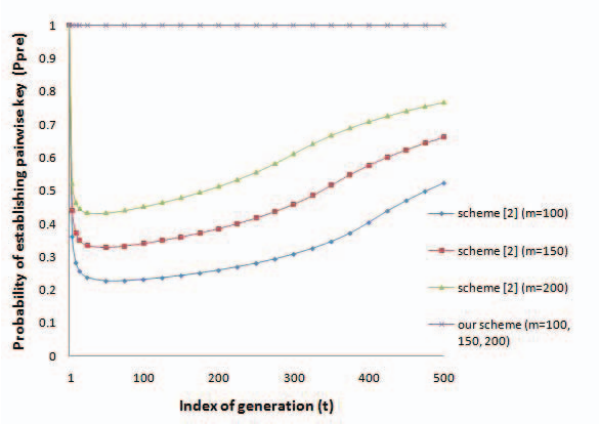

Figure 4. The comparison of the probabilities that N_i establishes pairwise key with pre-deployed sensors and the sensors being deployed in the same generation in our scheme and the scheme in [2].

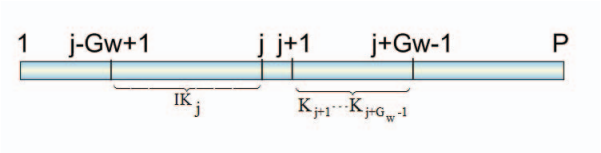

Figure 5. Nodes deployed at generation j can establish pairwise nodes with at most $2G_w - 1$ generations with 100% probability.

current generation and m master keys of randomly chosen generations after deployment, the probability of sharing key material with other prospective nodes is $\frac{m}{P-t}$ where P is the number of total generations (Please refer to [2] for the process of calculation). The variation $1 \leq t < P$. When $P - t \leq m$, the master keys of all remaining generations will be preloaded to the nodes deployed at generation t so that the key connectivity probability $p_{pros}(t)$ is 1 and keeps 1 for the subsequent generations. We make a comparison of the probabilities that N_t establishes pairwise key with prospective groups in our scheme and the scheme in [2] in Figure 3. In order to facilitate the comparison, we keep the size of key pool (the same as the number of generations of a network) 500. The three curves in Figure 3 demonstrate the conditions of [2] when m equals 100, 150, and 200, respectively. As shown in the figure, the probability p_{pros} increases as the index the generation t increases. In our scheme, the group N_t can always share keying material with groups in its generation window $[t, t + G_w - 1]$ with 100% probability. The variation $1 \leq m \leq P-1$, it is easy to reach the conclusion that the more master keys a sensor node has in [2], the higher probability of key connectivity becomes. In our scheme, more than $G_w - 1$ preloaded master keys do not increase the key connectivity probability as $G_w - 1$ master keys are enough for reaching 100% probability.

In terms of $p_{pre}(t)$, it can be calculated by using p_{pros}. If we suppose that sensor nodes are uniformly distributed at each generation, p_{pre} can be calculated as:

$$p_{pre}(t) = (\sum_{i=1}^{t-1} p_{pros}(i) + 1)/t,$$

where 1 means that a sensor node can establish pairwise keys with nodes deployed at the same generation with 100% probability. In fact, it is the average value of the probabilities of key connectivity with each preloaded and the current group of sensor nodes. Figure 4 describes the change tendency of probability p_{pre} for various m when the number of generation P is fixed to 500.

Different from the scheme in [2], when a sensor node is deployed at generation t, it is preloaded with the initial key IK_t and $G_w - 1$ master keys for the subsequent $G_w - 1$ generations. The node can establish pairwise keys with nodes in at most $2G_w - 1$ generations with 100% probability as long as both of them are still viable. As described in Figure 5, a node deployed at generation j can establish pairwise keys with nodes deployed from generation $j - G_w + 1$ to generation j by using its initial key IK_j and establish pairwise keys with nodes deployed from generation $j + 1$ to generation $j + G_w - 1$ by using the proloaded master keys for each generation. In time-based scheme [2], a node is preloaded with m master keys of randomly- chosen generations. Some of these master keys might not be used before the node is power off. Suppose a node is deployed at generation j and is viable at at most G_w generations, those master keys of the generations from $j+G_w$ to P are useless.

Storage Overhead. In terms of storage overhead, it is determined by the memory of a node and generations that a node can survive. In a practical application of network deployment model, the beginning of new generation means the occurrence of node additions Therefore, the number of generations is approximately equal to the number of node addition. Therefore, the higher frequency of node additions,

the larger number of generations a network has. According to our scheme, a sensor node surviving 100 generations has to store 100 keys including one initial key and 99 master keys. These node can establish pairwise keys with nodes in 199 generations with 100% probability. The modern sensor nodes such as MICA-Z have 128KB program memory, 4KB runtime memory, and 512KB external memory [10]. Suppose the size of a key is 128bits, our scheme requires only 1.6KB memory. Our scheme has reasonable storage requirement for modern sensor nodes.

V. Conclusion

The LEAP key management mechanism of LEAP is welcomed due to its multiple keying mechanism. However, the security of all types of keys is mainly depends on that of an initial key. It is assumed that the initial deployment phase is secure and the key is erased from sensor nodes after the initialization phase. However, the same key should be used again for node addition after that phase while the new node can be captured before removing the initial key. A time-based key management scheme was proposed to eliminate the effect of disclosure of the initial keys. The time-based scheme split the time domain of network into many time slots. Each time slot has its own initial key. This scheme does disperse the damage resulting from the disclosure of the initial key. However, the scheme reduces the probability of key connectivity. In contrast, the proposed scheme in this paper keeps 100% key connectivity within the node's lifetime time without degrade security. The established pairwise key is exclusively known only by the nodes of the generations which the two end nodes belong to. No node deployed at other generations can compute the pairwise key.

References

[1] S. Zhu, S. Sanjeev, and J. Sushil. *LEAP: efficient security mechanisms for large-scale distributed sensor networks*, Proceedings of the 10th ACM conference on Computer and communications security. Washington D.C., USA. ACM. 2003.

[2] J. Jang, T. Kwon, and J. Song. *A Time-Based Key Management Protocol for Wireless Sensor Networks*, Information Security Practice and Experience, pp.314-328, 2007.

[3] L. Eschenauer and V. Gligor, *A Key Management Scheme for Distributed Sensor Networks*, Proceedings of the 9th ACM Conference on Computer and Communications Security, pp.4147, 2002.

[4] H. Chan, A. Perring, and D. Song, *Random Key Predistribution Scheme for Sensor Networks*, Proceedings of the IEEE Symposium on Security and Privacy, 2003.

[5] M. Ergun, A. Levi, and E. Savas, *Increasing Resiliency in Multi-phase Wireless Sensor Networks: Generationwise Key Predistribution Approach*, The Computer Journal Advance Access, Published by Oxford Univerisity Press on behalf of The British Computer Society, May 11, 2010.

[6] FIPSPUB180-1. (1995) http://www.itl.nist.gov/fipspubs/fip180-1.htm (accessed July 12, 2010).

[7] FIPS PUB 180-2. (2002) http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf (accessed July 12, 2010).

[8] C. Castelluccia, and A. Spognardi, *RoK: A Robust Key Predistribution Protocol for Multi-phase Wireless Sensor Networks*, Proceedings of SecureComm 20073rd International Conference on Security and Privacy in Communications Networks, pp.351-360, 2007.

[9] K. Kalkan, S. Yilmaz, O. Z. Yilmaz, A. Levi, *A Highly Resilient and Zone-based Key Predistribution Protocol for Multi-phase Wireless Sensor Networks*, Proceedings Q2SWinet095th ACM International Symposium on QoS and Security for Wireless and Mobile Networks, Tenerife, Spain, pp.2936. 2009.

[10] Crossbow Technology (http://www.xbow.com)(Accessed July 20,2010)