ROLL Working Group Y. Qiu
Internet-Draft J. Zhou
Expires: September 13, 2012 F. Bao
Institute for Infocomm Research
March 12, 2012

# Lightweight Key Establishment and Management Protocol in Dynamic Sensor Networks (KEMP)

draft-qiu-roll-kemp-01

## Abstract

When a sensor node roams within a very large and distributed wireless sensor network, which consists of numerous sensor nodes, its routing path and neighborhood keep changing. In order to provide a high level of security in this environment, the moving sensor node needs to be authenticated to new neighboring nodes as well as to establish a key for secure communication. The document proposes an efficient and scalable protocol to establish and update the secure key in a dynamic wireless sensor network environment. The protocol guarantees that two sensor nodes share at least one key with probability 1 (100%) with less memory and energy cost, while not causing considerable communication overhead.

## Status of this Memo

This Internet-Draft will expire on September 13, 2012.

## Copyright Notice

## Table of Contents

1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . 3
2. Network Assumptions . . . . . . . . . . . . . . . . . . . . . 5
3. Shared-Key Discovery . . . . . . . . . . . . . . . . . . . . 6
4. Dynamic Authentication and Key Establishment Protocol . . . . 7
4.1. Basic Protocol . . . . . . . . . . . . . . . . . . . . . 7
4.2. Key Management . . . . . . . . . . . . . . . . . . . . . 8
4.3. Distribution Mode . . . . . . . . . . . . . . . . . . . . 10
4.4. Node Bootstraps . . . . . . . . . . . . . . . . . . . . . 11
4.5. Working with Multiple Trust Domains . . . . . . . . . . . 11
5. Security Consideration . . . . . . . . . . . . . . . . . . . 13
6. IANA Consideration . . . . . . . . . . . . . . . . . . . . . 15
7. Conclusions . . . . . . . . . . . . . . . . . . . . . . . . 16
8. Normative References . . . . . . . . . . . . . . . . . . . . 17
Appendix A. Version History . . . . . . . . . . . . . . . . . . 18
Authors' Addresses . . . . . . . . . . . . . . . . . . . . . . . 19

## 1. Introduction

The demand of wireless sensor networks (WSNs) is growing exponentially. It has turned out that the sensor networks can be widely applied in the areas of healthcare, environment monitoring, and the military. One of the surveys on WSNs points out that, in the near future, wireless sensor networks will be an integral part of our lives, more so than the present-day personal computer [1].

A sensor node has low capability in terms of power, computation, storage and communication. A wireless sensor network is composed of a large number of wireless sensor nodes and multi-hop communication is desired in WSNs. As a result, security in wireless sensor networks has six challenges to overcome: (a) the wireless nature of communication, (b) resource limitations of sensor nodes, (c) very large and dense WSNs, (d) lack of fixed infrastructure, (e) unknown network topology prior to deployment, (f) high risk of physical attacks on unattended sensors [2][3].

The capabilities in term of Scalability, Mobility/Dynamicity Network, Latency, etc. are also listed in the RFC documents, i.e. Routing Requirements for Urban Low-Power and Lossy Networks (RFC 5548)[6], Routing Requirements for Urban Low-Power and Lossy Networks (RFC 5673)[7], Home Automation Routing Requirements in Low-Power and Lossy Networks (RFC 5826)[8], and Building Automation Routing Requirements in Low-Power and Lossy Networks (RFC 5867)[9].

RFC 5548 required local network dynamics SHOULD NOT impact the entire network to be reorganized or re-reconfigured; a viable routing security approach SHOULD be sufficiently lightweight that it may be implemented across all nodes in a U-LLN; the U-LLN MUST deny any node that has not been authenticated to the U-LLN and authorized to participate to the routing decision process.

RFC 5673 addressed the handover speed; a compromised field device does not destroy the security of the whole network; because nodes are usually expected to be capable of routing, the end-node security requirements are usually a superset of the router requirements.

RFC 5826 needed a node MUST authenticate itself to a trusted node that is already associated with the LLN before the former can take part in self-configuration or self-organization. A node that has already authenticated and associated with the LLN MUST deny, to the maximum extent possible, the allocation of resources to any unauthenticated peer. The routing protocol(s) MUST deny service to any node that has not clearly established trust with the HC-LLN.

RFC 5867 listed the possible security keys below: a) a key obtained

from a trust center already operable on the LLN; b) a pre-shared static key as defined by the general contractor or its designee; or c) a well-known default static key.

With the aforementioned limitations of the existing solutions in mind, we now propose a secure protocol in dynamic WSN, addressing all of the following issues:

- A moving sensor node needs to change its attached routers (or cluster heads) frequently.

- A router (or cluster head) needs to ensure a joining node is not a malicious sensor.

- A moving node needs to establish a secure tunnel with the new router (or cluster head).

- The energy consumption for establishing the secure tunnel must be minimal.

One of the important novel features of the proposed protocol is that the router or cluster head is employed as sub-base-stations to execute key establishment.  This way, the total dependency on the base station for key establishment can be avoided.  Also, this approach reduces the hops between two communicating ends and hence results in reduction of the communication cost.

## 2. Network Assumptions

In this document, we consider a scenario in which a sensor node roams within a very large and distributed wireless sensor networks (WSN), consisting of a large number of sensor nodes and base stations. It is a typical scenario that is widely adopted in hospital environments as the patients or doctors equipped with sensors roam across each department in the hospital. A patient who carries the sensor nodes can move freely within the range of a hospital. When a wireless sensor node is moving, its routing path and neighborhood keep changing. The moving node needs to be authenticated to the new neighbors and to establish a key for secure communication.

This scenario reflects the problems described in Section 1: (a) composition by a large number of sensor nodes; (b) communication based on wireless multi-hop mechanism; (c) no fixed infrastructure; (d) the possible location change of sensor node (patient). Therefore, the challenges of this network assumption are how to establish a secure channel with these routers.

## 3. Shared-Key Discovery

In the WSN environment, as data transmission consumes much more energy than computation, the probabilistic solution is widely accepted in order to reduce the storage and communication overhead during key establishment.

So far in the literature, numerous random key pre-distribution schemes have been proposed. For example, in Chan et al.'s scheme[4], each sensor node stores a random set of Np dedicated pair-wise keys to achieve the probability p that two nodes share a key. At the key setup phase, each node ID is matched with Np other randomly selected node IDs with probability p. A distinct pair-wise key is generated for each ID pair, and is stored in both nodes' key-chain along with the ID of the other party. During the shared-key discovery phase, each node broadcasts its ID so that neighboring nodes can tell if they share a common pair-wise key. Note that Chan et al.'s scheme reduces the storage overhead by sacrificing key connectivity, but it still provides perfect key resilience.

In this protocol, it is assumed that a sensor node (carried by a patient) can move within a special range (e.g. hospital). As each sensor's memory is severely constrained, each sensor may only store a small set of keys randomly selected from a key pool at the deployment. Two nodes may use any existing key discovery protocol (e.g., the solution proposed in [4]) to find a common key from their own sets. If the common key is not found, the key establishment scheme will be initiated. The reason why binding a general pre-shared key discovery phase to the protocol is to reduce the energy cost as much as possible.

## 4. Dynamic Authentication and Key Establishment Protocol

### 4.1. Basic Protocol

Due to the limited storage of sensor nodes, the pre-shared key-pair is not always available between the roaming node and its new neighbors in the circumstance of a dynamic node roaming within large WSNs (e.g., in hospitals and nuclear power plants). Therefore it requires an efficient and scalable protocol to establish and update the keys among nodes for secure communications.

Figure 1 shows the basic architecture and message flow of our protocol for authentication and key establishment in dynamic WSNs. When a dynamic sensor node moves to a new area and wants to attach to a router or a cluster head in this area, it first sends a request message to the base station (refer to Figure 1).

```
                        +--------+
                        | Router |
                        +--------+
                       /         |\
               notice /            \ appv
                     /              \
                   |/                \
             +--------+           +---------+
             | Sensor |    req    |  Base   |
             |  Node  |---------->| Station |
             +--------+           +---------+
```

Figure 1. The basic architecture and message flow of KEMP protocol

req={Src=SN, Dst= BS, RT||R0||MAC(K_BN, SN||RT||R0)} (1)

where Src and Dst denote the source and destination address of a message respectively. SN, BS and RT are identifiers for sensor node, base station and router, respectively. R0 denotes a random number generated by the sensor node. MAC indicates the message authentication code algorithm with a key and K_BN is the shared secret key between the base station and the sensor node.

After receiving the req message, the base station will check its revocation list whether the sensor node has been revoked. If the sensor node is acceptable, then the base station verifies the MAC message. If the result is positive, the base station will generate a session key K_NR for the roaming sensor node and the router (or cluster head).

K_NR = H(K_BN, SN||R0||R1) (2)

where H is a keyed one-way hash function, and R1 is the random number selected by the base station. The base station then sends an approval message appv with the session key to the router:

appv = {Src=BS, Dst=RT, E(K_BR, SN||R0||R1||K_NR)} (3)

where E is an encryption algorithm, and K_BR is the shared secret key between the base station and the router.

After receiving the appv message, the router decrypts the payload and extracts the session key KNR, and then sends a notice to the sensor node.

notice = {Src=RT, Dst=SN, R0||R1|| MAC(K_NR, RT||SN|| R0||R1)} (4)

Upon getting the notice message, the sensor node extracts the random numbers R0 and R1. After checking if the received random number R0 is equal to the original R0, the sensor node recalculates the session key K_NR = H(K_BN, SN||R0||R1) and then verifies the MAC value. If the result is positive, the sensor node will use the session key for the communication with this router afterwards. The R0 cannot be ignored because the sensor node might send out many request messages with various R0 if it cannot receive the notice message in time. Hence, the sensor node must know which R0 is used in the notice message.

In practice, the router could be any sensor node that the dynamic sensor node wants to connect to.

### 4.2. Key Management

In order to manage the keys, every sensor node maintains a table, called "Key Cache". Table 1 shows the structure of the Key Cache.

Table 1. The structure of Key Cache

| Key Cache in Sensor Node N | | |
|---|---|---|
| Correspondence Node ID | Key | Key Lifetime |
| BS | K_BN | T_BN |
| Node_i | K_Ni | T_Ni |
| ... ... | ... | ... ... |
| Node_j | K_Nj | T_Nj |
| PreSharedKey_x | K_x | T_x |
| ... ... | ... | ... ... |
| PreSharedKey_y | K_y | T_y |

When a sensor node, say node N, wants to connect to other sensor node, say node R, it executes the following procedure:

(1) Checks first if there is an existing key pair between them.

(2) Otherwise, processes the subroutine of shared-key discovery to find a common key between node N and node R based on those "PreSharedKeys" in their key caches.

(3) If there is still no common key between them, the sensor node allocates an entry in the key cache, and assigns Node ID as nodeR, Key Stuff as the random number R0 and Key Lifetime as 0, as shown in Table 2.

Table 2. The initial key entry.

| Correspondence Node ID | Key | Key Lifetime |
|---|---|---|
| Node_R | R0 | 0 |

(4) Then the sensor node initiates the procedure of key establishment described in the above section. After receiving the notice message, and recalculating the session key KNR, the sensor node updates the entry's key stuff and key lifetime accordingly.

(5) When the key lifetime is expired, the dynamic sensor node should re-initiate the procedure of key establishment described in the above section.

(6) When the sensor node leaves the range of the connected router, the sensor node deletes the related entry from its cache table in order to save the storage. In case there is no space for adding a new entry, it may first delete the oldest key which has expired or will expire soon.

The base station also maintains a key table (Table 3) that includes the secret keys shared with all of the sensor nodes in the network.

Table 3. The structure of Key Table in basestation

| Key Table in Base Station | | |
|---|---|---|
| Node ID | Key | Key Lifetime |
| Node_i | K_Bi | T_Bi |
| ... ... | ... | ... ... |
| Node_j | K_Bj | T_Bj |

If a node is compromised and revoked, its field of key lifetime would be marked as negative.

### 4.3. Distribution Mode

In WSNs, the more hops between two communicating ends exist, the poorer the traffic performance becomes and the more energy consumption is required. To overcome these problems, we introduce the distribution mode.

The major idea of distribution mode is to deploy the cluster heads as the sub-base-stations because a cluster head is more powerful than normal sensor nodes. The distribution mode includes the following steps:

(1) Each cluster head manages to establish the shared key with its neighboring cluster heads after deployment. There are several ways to do this. One could embed those keys in advance if the topology is known at deployment, or use the basic protocol described in the above sections, via the base station. (As this is a one-time operation, the overheads may be acceptable.)

(2) Each sensor node keeps two base station identifiers (IDs): one is a real base station ID; the other is a sub-base-station (the cluster head) ID. Initially, the ID of sub-base-station is a real base station.

(3) After deployment, the first round for a mobile node to establish the shared key with the nearest cluster head uses the basic protocol, too.

(4) When the mobile node moves, use the basic protocol to establish the shared key with the new cluster head, via the sub-base-station (old cluster head) rather than the real base station.

(5) After successfully establishing the keys, the sensor node updates the ID of sub-base-station with the current cluster head.

(6) For security reasons, each sensor node must reset its sub-base-station ID to the real base station at a specified interval (say a few hours or days, depending on the various applications) and re-establish keys with its near cluster heads via the real base station. If the base station does not receive any request from a sensor node, it considers the sensor node has been compromised.

The distribution mode could provide an efficient and low energy-cost solution for the shared-key establishment. The basic protocol can provide the stronger protection since it can immediately block and revoke compromised nodes.

## 4.4. Node Bootstraps

The description in this paragraph is how to establish the secure session between sensor node SN and its first router RT_first when the node wake up in a new sensor networks.

(1) After bootstrapping, the sensor node SN sends its first request to Base Station BS via RT_first itself (in generic, via the reviopuse router RT_previous) as below:

```
req={Src=SN, Dst= BS, RT_first||R0||MAC(K_BN, SN||RT_first||R0)}        (5)
```

(2) Hence, the BS will return appv message to RT_first. Upon receiving the notice from RT_first, the sensor node SN could establish the secure session with RT_first by normal processing descibed in previous sections 4.1 and 4.2

## 4.5. Working with Multiple Trust Domains

This paragraph describes the operation in scenarios of Multiple Trust Domains.

(1) If these multiple domains are managed by one base station (key centre), each node address should include the prefix of the domain. With the prefix, a base station / key centre could distinguish each node and avoid any confliction.

(2) If these multiple domains have their own base station, extend the node's cache table to store the pre-shared secrets between the node and these base stations.

(3) If the sensor node cannot decide which base station is its destination, let the req message carry a set of all of MACs with generated by the secret between the node and the basestation, respectively.

## 5. Security Consideration

In this proposed protocol, the session key K_NR between the sensor node and the router is generated by the base station and sensor node respectively, and the session key is directly sent to the router from the base station by an encrypted packet. Hence, the session key K_NR is never disclosed during transmission. The session key K_NR is only known by the related peers, i.e., the sensor node, the base station and the router.

Referring to equation (2), the session key K_NR is generated by a keyed hash function with the shared key K_BN between sensor node and base station as well as two random numbers, R0 and R1, which are generated by the sensor node and base station respectively. As both R0 and R1 are used only one time, there are not the same session keys K_NR. This property is useful to against the replication attacks.

Since the session key K_NR is generated by a keyed hash function with the secret key K_BN between the sensor node and the base station, the different sensor nodes will have different session keys. This feature is useful to protect sensor node privacy.

Even though an eavesdropper at the edge of the sensor node can monitor and capture the random numbers R0 and R1 as well as the identity of the sensor node, it is still not able to regenerate the session key K_NR due to lack of the secret key K_BN. Without a proper session key, the routers will not forward the packets to next nodes. This attribute could prevent camouflage and traffic attacks.

Due to the fact that no trusted connection is established between sensor node and new router before the connection between them, the proposed protocol employs a random number R1 issued by the base station. The sensor node needs to recalculate the K_NR first based on the R1 together with K_BN and R0. Then using the calculated session key K_NR to verify the received session key K_NR and the random number R1. If the result is positive, then the sensor node will trust that the router is authorized by the base station.

Besides the function of informing the sensor node that the new session key K_NR is ready to use in the router, the notice message also plays an important role to check if the sensor node!_s address is reachable. Without this reachability check, the sensor node may claim that it is at any location rather than its real location. It could launch redirecting attacks.

The path between the base station and the router is secure because the packet between them is encrypted with a pre-shared key K_BR.

The messages from the sensor node to the base station and from the router to the sensor node are authenticated by a keyed hash function. Before accepting the inward message and making further processing, the receivers must verify the authentication. Since the cost of a hash algorithm is very small, the base station and sensor node could avoid the attacks of denial of service.

In order to achieve high efficiency and low energy cost, the protocol deploys a distribution mode which uses the cluster headers as the sub-base-stations. Due to the capability of cluster header, it is not able to recognize any compromised sensor nodes in time; the protocol requires each sensor node to reset its sub-base-station ID to the real base station regularly, and to re-establish keys with its near cluster heads via the real base station. This step is also useful to avoid a sensor node binding a compromised cluster head for long time.

According to the above analysis, this proposed protocol, which is simple and easy to implement, can provide relatively strong protection for sensor node networks.

The solutions based on public keys are not suitable in sensor networks. As sensor nodes are very easy to be compromised and lost, the public-key revocation is a very hard work in low power and lossy networks due to the 2 reasons below:

(1) How and how often to update the revocation list?

(2) How to store the revocation list in sensor node?

Moreover, before deploy the negoation of public keys, the new attached node must be ensured it is reachable in the sensor networks. Our proposal lets the previous router and basestation to endorse the new attaching node.

## 6. IANA Consideration

This version does not need new values to be assigned by IANA.

## 7. Conclusions

In this document, we have proposed an efficient and scalable protocol to establish and update the authentication key between any pair of sensor nodes in a dynamic wireless sensor network. Our protocol has the following features:

- It is suitable for both static and dynamic WSNs. Any pair of nodes can establish a key for secure communication.

- A roaming node only deals with its closest router for security. There is no need to change the rest of routing path to the base station.

- The base station can manage a revocation list for lost or compromised roaming nodes.

- The system is scalable and resilient against node compromise.

- The protocol is efficient due to the small number and size of signalling messages.

- The size of each signalling message is smaller than the IEEE 802.15.4 frame size so that it can to avoid packet fragmentation and the overhead for reassembly.

- The distribution mode can considerably reduce the latency.

- Any pair of nodes can establish a key. The protocol guarantees that two sensor nodes share at least one key with probability 1 (100%).

Thanks to above features, the protocol can satisfy the requirements for IPv6 over Low power WPAN Routing [5] and could be the security solution deployed in Routing Requirements for Urban Low-Power and Lossy Networks (RFC 5548)[6], Routing Requirements for Urban Low-Power and Lossy Networks (RFC 5673)[7], Home Automation Routing Requirements in Low-Power and Lossy Networks (RFC 5826)[8], and Building Automation Routing Requirements in Low-Power and Lossy Networks (RFC 5867)[9].

After comparing with some of the popular and latest protocols used in WSNs, our protocol could save about 30% in communication energy, and has the higher probability (100%) of sharing a key between two sensor nodes with less memory cost than those pre-distribution schemes, without incurring in a considerable amount of communication.

## 8. Normative References

[1] Akyildiz, I., Sankarasubramaniam, Y., and E. Cayirci, "Wireless sensor networks: a survey", Comput. Netw 38, 393-422, 2002.

[2] Camtepe, S. and B. Yener,, "Key Distribution Mechanisms for Wireless Sensor Networks: a Survey", Technical Report TR-05-07; Department of Computer Science, Rensselaer Polytechnic Institute: Troy, NY, USA , Mar. 2005.

[3] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, August 2007.

[4] Chan, H., Perrig, A., and D. Song, "Random key predistribution schemes for sensor networks", IEEE Symposium on Research in Security and Privacy Oakland, California, USA, May 2003.

[5] Kim, E., Kaspar, D., Gomez, C., and C. Bormann, "Problem Statement and Requirements for 6LoWPAN Routing", Work in Progress, Aug. 2010.

[6] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, May 2009.

[7] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, October 2009.

[8] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, April 2010.

[9] Martocci, J., De Mil, P., Riou, N., and W. Vermeylen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, June 2010.

## Appendix A. Version History

- v00 to v01

  * Add a new section about the processing of node bootstraps.

  * Add a new section about the multiple trust domains.

  * The modification is based on the feedback from Rene Struik, Steve Childress, Shoichi Sakane, Greg Zaverucha, Matthew Campagna.

## Authors' Addresses

Ying Qiu
Institute for Infocomm Research, Singapore
1 Fusionopolis Way
#21-01 Connexis (South Tower)
Singapore  138632

Phone: +65-6408 2053
Email: qiuying@i2r.a-star.edu.sg

Jianying Zhou
Institute for Infocomm Research, Singapore
1 Fusionopolis Way
#21-01 Connexis (South Tower)
Singapore  138632

Phone: +65-6408 2075
Email: jyzhou@i2r.a-star.edu.sg

Feng Bao
Institute for Infocomm Research, Singapore
1 Fusionopolis Way
#21-01 Connexis (South Tower)
Singapore  138632

Phone: +65-6408 2073
Email: baofeng@i2r.a-star.edu.sg