Analysis of Key Management Schemes in Dynamic Wireless Sensor Networks

Seyed Hossein Erfani1, Hamid H. S. Javadi2, and Amir Masoud Rahmani1

Email: h.erfani@srbiau.ac.ir

1Department of Computer Engineering, Islamic Azad University, Science and Research branch, Tehran, Iran
2Department of Mathematics and Computer Science, Shahed University, Tehran, Iran

Abstract

With the increasing use of sensor nodes in several stationary and mobile devices in today's Internet-connected life, providing security is a challenging issue. Ensuring security in wireless sensor networks is a popular subject which has been studied for almost a decade on static networks. However, due to the new trend to use sensor nodes in mobile applications, considering dynamicity is important in providing secure communication between sensor nodes. In particular, providing efficient key management schemes for mobile networks, due to its potential limitations is essential. In this paper, we provide a survey on existing key management schemes proposed for dynamic wireless sensor networks. We also provide a comparison between these schemes in terms of different security and efficiency metrics.

Keywords: Key Management Schemes, Dynamic Wireless Sensor Networks

1. INTRODUCTION

Dynamic wireless sensor network (DWSN) composed of lightweight mobile sensor devices with considerable limitations in computational power, energy, and memory space. Sensor nodes collected environmental data and transmit to a sink node or a base station (BS). The use of mobile nodes in mobile applications, considering dynamicity is important in providing secure communication between sensor nodes. In particular, providing efficient key management schemes for mobile networks, due to its potential limitations is essential. In this paper, we provide a survey on existing key management schemes proposed for dynamic wireless sensor networks. We also provide a comparison between these schemes in terms of different security and efficiency metrics.

A. Evaluation metrics

The evaluation metrics of key management schemes can be categorized as security, efficiency and flexibility metrics. Due to the special characteristics of DWSNs, and DKM schemes’ applications, we have to consider specific metrics and requirements in order to use such schemes. The metrics which should be considered in DKM schemes are: node revocation, forward secrecy, backward secrecy, collusion resistance.

a) Security metrics: DKM schemes provide secure cryptographic keys. Upon detecting a node capture attack, the keys assigned to the captured node should be revoked and some new generated keys should be distributed between other non-captured sensor nodes. Furthermore, a convenient scheme should also ensure forward and backward secrecy, as well as...
prevention of collusion between captured and newly added nodes. Moreover, it is desirable to have sufficient resilience against node capture attack and node replication.

- Node revocation. A desirable approach should revoke captured node’s keys upon detection and remove it from the network. This way, the captured node will not be able to inject false data and manipulate the correct data.
- Forward and backward secrecy. Forward secrecy prevents a new message to be decrypted by an expired key. In contrast, backward secrecy refers to prevention of decrypting old messages by a node using its new keys. These two concepts are used to mitigate the node capture attack.
- Collusion attack. An attacker might compromise some nodes in the network and force them to collude and disclose the whole network keys. An efficient key management scheme should resist against collusion between compromised nodes and new nodes.
- Resilience. Resilience refers to resistance against node capture attack in the presence of physical attacker which recovers the stored secrets in the nodes. To measure the resilience we have to consider the effect of capturing one node in the network. If the attacker cannot extract more information than the captured node, the resilience of the scheme is high. In contrast, if by capturing a single node an attacker can compromise whole the network, the resilience is low.

b) Efficiency metrics: The number of exchanged messages for re-keying, number of required cryptographic keys, and required operations should be low. Therefore, we do not need to consider limitations for network size, and it prevents fast depletion of nodes energy and wasting the memory space. In DKM schemes, the following metrics should be taken into account:

- Memory. The required memory space to store cryptographic keys (such as public, private and symmetric keys), user certificate (such as ID), and trust certificate (such as neighbors reputation).
- Band width. The number and size of the exchanged messages in key generation, re-keying and key revocation procedures.
- Energy. The energy consumption during the key agreement and data transmission procedures and computational functions required to generate and distribute the new generated keys.

c) Flexibility metrics: Key generation methods should be flexible in order to support various scenarios in WSNs. The major flexibility metrics are:

- Mobility. In general, the sensor nodes supposed to be static in most of the network scenarios. However, we may need mobile BS, mobile sensor nodes or both in specific applications [6]. In such scenarios, key establishment procedure should be able to assign new keys to mobile nodes through their new neighboring nodes. It is more difficult to generate and distribute new keys for mobile nodes which is strongly dependent to the mobility pattern, energy and bandwidth.
- Scalability. A WSN might be composed of hundreds or thousands sensor nodes. Furthermore, some new nodes may join the network or some current nodes may leave the network as well. Therefore, a convenient key management scheme should be able to support various network sizes, i.e., should be scalable. Moreover, the security and efficiency metrics for small networks should also satisfied for large networks.
- Key connectivity. This term refers to the probability that two (or more) arbitrary nodes be able to establish a common key among themselves after the re-keying process. The connectivity between each pair of neighboring nodes is called local connectivity. In contrast, global connectivity is considered as the connectivity of whole network. Providing high key connectivity after each rekeying procedure is needed in order to provide continuous security.

B. Organization

The remainder of the paper is structured as follows: In Section II we survey the existing key management schemes, and we provide an analysis in Section III. Finally, we conclude the paper and highlight the future work in Section IV.

2. KEY MANAGEMENT SCHEMES

In this section, we review the existing key management schemes for DWSNs. Due to the fact that in DWSNs the network topology is dynamic, the traditional key management schemes for WSNs are not suitable for such networks. Therefore, recently researchers proposed some new effective key management schemes in which the key generation overhead is lower than static key management schemes.

Qiu et al. [7] proposed a hybrid approach in which both key pre-distribution and post-deployment key establishment methods have been used. The authors adopted the random key pre-distribution scheme [2] and pair-wise post deployment key generation scheme. In the proposed scheme, each key has been assigned a time-stamp and a lifetime value. The timestamp value denotes the key generation time and the lifetime value indicates the key removal time. Scalability is one of the major advantages of this scheme; however, the memory consumption of this scheme is high due to the amount of memory required to store keys’ lifetime and time stamp in each sensor node. Key lifetime plays an important role in performing trade-offs between resilience of the scheme and energy consumption; considering large values for the key lifetime leads to a reduction in network resilience whereas choosing small values results in more energy consumption due to the need for more
new key generation processes. Furthermore, choosing small values for key lifetime leads to increasing the number of key generation process by each sensor node whereas considering large values for key lifetime results in more memory consumption. Therefore, considering the limitations of sensor nodes, this scheme is not very efficient and flexible.

In another research study, Han et al. [8] proposed a ticket-based approach which decreases the mobile nodes' reauthentication overhead and offers an efficient authentication and key exchange procedure. In this scheme, each sink node authenticates other neighboring sink and sensor nodes. Once a node leaves a sink node's radio range and connects to a new sink node, the new sink node will be able to re-authenticate the sensor node without imposing a high communication and computation overhead to the network. After the authentication process, the sensor node generates a common key with the sink node (i.e., the sink node that is located in its communication range) for secure communication. This scheme consists of 5 phases: (i) neighbor discovery, (ii) key establishment between sink nodes and BS, (iii) authentication between sink nodes, (iv) key establishment between sensor nodes and corresponding sink nodes, (v) re-authentication between mobile sensor nodes and new sink nodes. The major advantage of this scheme is its low computation overhead in re-authentication phase. However, the most important shortcoming of this scheme is that there should exist enough sink nodes in the network to cover all the area, otherwise, if there exist a sensor node which does not reside in any sink node's radio range, it will not be able to communicate with other sensor nodes and BS.

EDDK scheme is a distributed deterministic key management approach proposed by Zhang et al. [9]. In this scheme, each sensor node is pre-loaded by an initial key and a network wide shared pseudo-random function. Each sensor node is able to compute its individual key using its initial key and function. Moreover, each sensor node stores a table which maintains the information of neighboring nodes such as neighbor ID, pairwise key, sequence number. It also shares a local cluster key with its neighboring nodes and stores this key in the table as well. EDDK scheme consists of key establishment, data transfer and key maintenance phases. The authors illustrated that this scheme has high resilience due to the fact that pairwise keys are decentralized and compromising a sensor node does not threaten other communication links. Furthermore, they proved that this scheme is resilient against Sybil and node replication attacks. However, the major shortcoming of this scheme is that it is not suitable for large scale and dense networks.

In [10], Erfani et al. proposed a DKM scheme which provides perfect key connectivity. The authors adopted both random key pre-distribution scheme and post-deployment key management methods. In this scheme, the memory space of each sensor node is divided into two parts to store $\alpha$ predistributed keys and $\beta$ post-deployment keys. If a pair of sensor nodes, which reside within each other’s radio range, have a common key (either pre-distributed or post-deployment key), they can communicate securely. Otherwise, if the two neighboring nodes do not share any common key, they execute a procedure to generate a shared post-deployment key. The authors indicate that their proposed scheme outperforms the key pre-distribution schemes in terms of resilience and scalability. The major shortcoming of this scheme is that it is not efficient for highly dynamic networks.

3. ANALYSIS

In this section, we provide a comparison analysis between the schemes explained in the previous section, considering the evaluation metrics introduced in Section I. The security comparison results have been illustrated in Table I. As it can be seen, all the schemes provide forward and backward secrecy and collusion resistance. In the schemes proposed by Qiu et al. [7] and Erfani et al. [10], the resilience is medium as they adopted pre-distributed keys. In general, the key predistribution schemes have low resilience against node capture attack, while the schemes which use only post-deployment keys, have high resilience. However, the hybrid schemes (which use both pre-distribution and post-deployment) have medium resilience.

Table II shows the efficiency and performance evaluation of the aforementioned schemes. In terms of memory consumption by sensor nodes (without considering the memory space required by sink nodes and BS), as it can be seen, in the scheme proposed by Qiu et al. [7] each sensor node keeps $\alpha$ pre-distributed keys, $\beta$ post-deployment keys and one master key which is used to generate post-deployment keys. In the scheme proposed by Han et al. [8], each sensor nodes stores one master key which is used for node authentication and key generation, and one ticket used in re-authentication process when a sensor node leaves the radio range of one sink node and connects to a new sink node. In EDDK [9], each sensor node has a key $K_e$ and a sequence number $SN_e$, and stores a key $K_g$ and a sequence number $SN_g$ for each neighbor to be able to communicate with its neighboring nodes. Using these keys and sequence numbers, each sensor node generates a unique key $K_{eg}$ to securely communicate with one of the neighboring nodes. In the scheme proposed by Erfani et al. [10], each sensor node stores $\alpha$ pre-distributed and $\beta$ post deployment keys. It uses the pre-distributed keys to generate the new post-deployment keys.

In terms of computation overhead, the three most CPU consuming functions are: encryption, decryption and hash function. In the Processing column of the Table II, we demonstrated the computation overhead imposed by the aforementioned functions for each of the considered key management schemes. As it can be seen, all the schemes have almost the same computation overhead.
To compare the scalability of the mentioned schemes, we considered the efficiency of the schemes in various network sizes. As it has been illustrated in Table II, the EDDK scheme has the worst scalability due to the fact that each sensor node needs to store a neighbor table and each node have to generate a new common key with the newly added nodes. The scheme proposed by Han et al. [8] is highly scalable, since each sensor node only communicates with sink nodes and only requires a shared key with the sink node, not the neighboring sensor nodes. Therefore, the changes in the network size do not affect the number of required key generation processes. The schemes proposed by Qiu et al. [7] and Erfani et al. [10] have medium scalability. As in these schemes, each sensor node stores some pre-distributed keys, the key generation process is not necessary for all the newly added nodes (i.e., they may have common pre-distributed key in order to communicate).

In terms of connectivity, Table II shows that all the specified schemes provide full connectivity between all the sensor nodes in the network, except the scheme proposed by Han et al. [8].

As it can be seen in Figure 1, as network size increases, the approach proposed by Erfani et al. [10] outperforms the Han and EDDK schemes and it is almost the same as Qiu scheme. This is due to the fact that, the size of the exchanged messages to generate a shared key in the scheme proposed by Erfani et. al. is smaller than the size of the messages in Qui et. al. scheme. Moreover, in Han scheme, the sensor node generates the shared key with the sink node in initial and reauthentication phases. Therefore, each pair of sensor nodes can communicate only through a sink node. While, in the scheme proposed by Erfani et. al. and Qui et. al., the sensor nodes use their pre-distributed keys to communicate with sink nodes and other neighboring nodes.

### 4. CONCLUSION

In this paper we reviewed the state of the art of the dynamic key management schemes in wireless sensor networks, as well as their advantages and shortcomings. The major advantage of adopting such schemes is their ability in prolonging the network lifetime. As we discussed in the paper, there is not any perfect scheme which satisfy all the evaluation metrics. Each of the explained schemes has specific characteristics which make them suitable for particular applications. The scheme proposed by Han et al. [8] is suitable for networks in which there exist enough sink nodes in the network to cover all the network area. EDDK [9] scheme is applicable for small and non-dense networks. The schemes proposed by Qiu et al. [7] and Erfani et al. [10] are suitable for networks in which sensor nodes are less dynamic, the network is not dense and the sensor addition rate is low.
We believe that his paper can be used as a guide for researches to design key management schemes for highly dynamic wireless sensor networks. A convenient solution should consume small memory, and impose small computation and communication overhead to the network. Furthermore, it should be flexible in terms of high mobility of sensor nodes and support for joining new nodes to the network.

REFERENCES


