An Authentication and Key Management Scheme for Heterogeneous Sensor Networks

Sarmadullah Khan[1(✉)], Rafiullah Khan[2], Inam Bari[3], and Naveed Jan[1]

[1] Electrical Department, CECOS University, Peshawar, Pakistan
{sarmad,naveed}@cecos.edu.pk
[2] Communication and Networks Lab, University of Genova, Genoa, Italy
Rafiuk7@gmail.com
[3] Electrical Department, FAST NUCES Peshawar, Peshawar, Pakistan
inam.bari@nu.edu.pk

Abstract. Recently, wireless sensor networks have attracted the attention of research comunity due to its numerous applications especially in mobility scenarios. However it also increases the security threats against confidentiality, integrity and privacy of the information as well as against their connectivity. Hence a proper key management scheme needs to be proposed to secure both information and connectivity as well as provide better authentication in mobility enabled applications. In this paper, we present an authentication and key management scheme supporting node mobility in a heterogeneous sensor networks that consists of several low capabilities sensor nodes and few high capabilities sensor nodes. We analyze our proposed solution agaist a well know attacks (sybil attacks) to show that it has good resilience against attacks compared to some existing schemes. We also propose two levels of secure authentication methods for the mobile sensor nodes for secure authentication and key establishment.

Keywords: Key management · Authentication · Sybil attacks

1 Introduction

The Wireless Sensor Network (WSN) are usually deployed in possibly remote and unattended locations they are definitely prone to security attacks. Hence to secure the network operation and securely gather and forward the information, security threats and its counter measures should be considered at design time in terms of both requirements and implementation techniques. The design of security algorithms considering the homogeneous sensor networks was the first step to secure sensor networks. However, some research work [1,2] have shown that homogeneous sensor networks have high communication and computation overheads, high storage requirements and suffer from severe performance bottlenecks. Hence, recent research work [3,4] introduced heterogeneous sensor networks, which consists of High-end sensors nodes (H-sensors) and Low-end sensors nodes (L-sensors). To achieve better performance and scalability, H-sensors have

J. Tian et al. (Eds.): SecureComm 2014, Part I, LNICST 152, pp. 123–131, 2015.
DOI: 10.1007/978-3-319-23829-6_9

more resources compared to L-sensors. However, both H-Sensors and L-sensors are still highly vulnerable in nature and are exposed to several security threats and particularly prone to physical attacks. Thus, proper security mechanisms should be applied to protect these nodes against attacks. Hence, a novel key management scheme for heterogeneous sensor networks suitable for scenarios with partial mobility is presented. The proposed solution relies on two types of keys: authentication keys and secret communication codes used to generate secret keys whenever needed. The remaining of the paper is organized as follows. Section 2 presents existing work. Section 3 describes the proposed key management scheme, while in Sect. 4 describe the security analysis of the proposed scheme, and finally conclusions are provided in Sect. 5.

2 Related Work

To secure wireless sensor networks, Perrig [5] proposed SPINS, in which there a secure central entity called server which is responsible for establishing a key among the sensor nodes. Since it is based on centralized base station approach, the failure of base station severely affects the performance of network. To overcome the above mentioned issue, a randomly key distributed approach is proposed by Eschenauer and Gligor [3]. In this scheme, there is no centralized entity like a base station for key distribution and management. Each node in the network is assigned a set of randomly selected keys from a large key set. Since the keys are distributed randomly, the two communicating nodes need to have at least one common key in their sets for secure communication. To further improve the network security, sharing of at least q-keys concept for establishing a secret key is introduced by Chan [6]. The prior knowledge of node's deployment in the network helps in increasing the network connectivity and reduce the memory requirements [7] combined with the Rabin's scheme [14]. To achieve better security and network connectivity with less memory requirements with low computational cost, NPKPS scheme is proposed by Zhang [8] for wireless sensor networks. To reduce the memory cost, Kim [9] introduced a level-based key management scheme while a two-layered dynamic key management for clustered based wireless sensor networks is presented by Chuang [10].

The management of secret keys (MASY) protocol is presented by Maerien in [11] which is based on the trust assumption among the networks managers/base stations. To further improve the network connectivity and reduce the memory requirements of the symmetric key distribution approaches, Du [4] presents an asymmetric key pre-distribution (AP) approach. Du sensor network model consists of two different types of nodes making it a Heterogeneous Sensor Networks (HSNs). This assumption significantly increases the network connectivity and reduces memory requirements compared to the existing symmetric key management approaches. Lu [12] proposes a framework for key management schemes in distributed peer-to-peer wireless sensor networks with heterogeneous sensor nodes and shows by simulation that heterogeneity results in higher connectivity and higher resilience. Du [13] proposes a routing-driven key management scheme

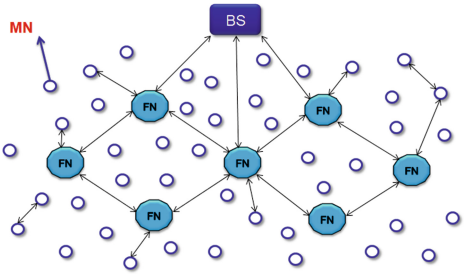

Fig. 1. Virtual network architecture

for heterogeneous wireless sensor networks, based on Elliptic Curve Cryptography (ECC), which provides better security with significant reduction of memory overhead.

The considered network model is a Heterogeneous Sensor Network (HSN) composed base station and H-sensors (fixed) while L-sensors are Mobile Nodes (MNs). The virtual network organization is shown in Fig. 1.

CH:	Cluster Head	MN:	Mobile Node
FN:	Fixed Node	P_{main}:	Main large key pool
P_{FN}:	Sub key pool for Fixed Nodes	P_{MN}:	Sub key pool for Mobile Nodes
K_{plc}:	Public key	K_{prt}:	Private key
prand():	Prime number generator	C_{auth}:	Authentication code
PN:	Generated prime number	S_{MN}:	Scalar product of a Mobile Node
T_{FN}:	Scalar product of a Fixed Node	SCC:	Secret communication code

3 Proposed Scheme

First we describe a list of abbreviations used in the proposed solution. Since the proposed key management scheme is built on top of the above network model to provide effective authentication and dynamic key establishment. The key material is generated at the BS. More specifically, a large key pool P_{main} is created and then divided into two sub key pools KP_{FN} and P_{MN} such that $P_{MN} \cap P_{FN} = \emptyset$.

The key pool P_{FN} is used by the FNs of the network while the key pool P_{MN} is used by the MNs of the network for the secret key establishment. For authentication purposes, Elliptic Curve Cryptography (ECC) is used during the initialization phase for key generation. Three different phases have been taken into account

1. Key pre-distribution
2. Node's authentication
3. Communication key establishment.

Further details will be provided in the following subsections.

3.1 Key Pre-distribution

Each FN i is assigned a randomly selected key pool P_{FN_i} from the key pool P_{FN} where $P_{FN_i} << P_{FN}$ and contains $|P_{FN_i}|$ keys while each MN j is assigned a randomly selected key pool P_{MN_j} from the key pool P_{MN} where $P_{MN_j} << P_{MN}$ and contains $|P_{MN_j}|$ keys. Since these two key pools are disjoint, $P_{FN_i} \cap P_{MN_j} = \emptyset$. These assigned key pools will be used by the FNs and by the MNs for the establishment of a secret communication key using the assigned key generation algorithm.

Concerning the authentication key material, each FN and each MN is assigned an elliptic curve $E(a, b)$ over a finite Galois field $F(G)$ and a base point G along with a unique authentication code C_{auth}. Each FN and each MN is also assigned an ECC-based public/private key pair (K_{plc}, K_{prt}) and a prime number generator ($prand()$).

All the previously introduced key material is transferred to each node of the network by means of secure side channels. Then, after this pre-distribution phase, the specific key material assigned to each type of node of the network is as follows:

- the BS owns all the key material that needs to be pre-distributed (plus, as already described, the public key of each FN)
- each FN i has been given $E(a, b)$, G and C_{auth_i} for authentication purposes and key pool P_{FN_i} for communication key establishment
- each MN j has been given $E(a, b)$, G and C_{auth_j} for authentication purposes and P_{MN_j} for communication key establishment.

3.2 Node Authentication

After the deployment and key pre-distribution phase, each FN of the network broadcasts periodic Hello messages. This mechanism enables each FN to fill a table with all neighboring MNs. The FN ID is included in the Hello message along with a random nonce signed by the FN's private key. Upon the reception of those Hello messages, each MN selects a FN as its Cluster Head (CH), e.g. the one with the highest signal strength, after the verification of Hello message by using the FN public key. Since Hello message verification is a part of the authentication phase, at this point the authentication phase among the FNs and the MNs can start. To this aim, each MN_j authenticates the Hello message of the selected FN_i as a CH as follow: First MN_j uses the FN_i ID and generates a prime number PN_{FN_i} using the prime number generator prand()

$$PN_{FN_j} = prand(ID_{FN_i}) \tag{1}$$

After the generation of PN_{FN_i}, the MN_j generates the public key of the FN_i as

$$K_{plc} = (PN_{FN_i} + ID_{FN_i}) \bullet G \tag{2}$$

Then the MN_j can verify the Hello message signature. Successful verification of the Hello message signature authenticates the CH i.e. FN_i to the MN_j. The MN then calculates the scalar product of the assigned authentication code C_{auth_j} and its private key as

$$S_{MN_j} = \left(C_{auth_j} + ID_{MN_j}\right) \bullet Kprt \tag{3}$$

Then the MN_j sends a joining request including its ID, S_{MN_j}, and the nonce it had received from the CH back to its selected CH, all signed by its private key. After receiving the MN_j's joining request message, the FN_i first authenticates MN_j before registering it as a trusted cluster member. The FN_i follows the same procedure as the MN_j did to check the authenticity of the received messages. First the FN_i use the MN_j ID and generate a prime number PN_{MN_j} using the prime number generator prand()

$$PN_{MN_j} = prand(ID_{MN_j}) \tag{4}$$

After the generation of PN_{MN_j}, the FN_i generates the public key of the MN_j using scalar multiplication as

$$K_{plc} = \left(PN_{MN_j} + ID_{MN_j}\right) \bullet G \tag{5}$$

After the generation of the MN_j public key, the FN_i verifies the joining message signature. Successful verification and reception of the correct nonce ensure that the MN_j is an authentic mobile node belonging to the network. The CH registers this MN_j into its authentic MN member list and calculates the scalar product of C_{auth_i} and its private key as

$$T_{FN_i} = (C_{auth_i} + ID_{FN_i}) \bullet K_{prt} \tag{6}$$

Finally the CH generates an authentication certificate for this MN using S_{MN_j} and T_{FN_i} as

$$Authentication\ Certificate = S_{MN_j} \bullet T_{FN_i} \mod G \tag{7}$$

The CH sends T_{FN_i} to the MN_j which uses in the secret key generation and for the authentication certificate generation.

3.3 Communication Key Establishment

Once the MN and CH/FN authenticate each other successfully, the key establishment phase starts. During this phase, the MN sends one of its secret communication codes SCC_1, randomly selected from P_{MN} and encrypted by the CH public key to its CH as described above. The CH also selects randomly another secret communication code SCC_2 from its pool P_{FN} and sends it to the corresponding MN. After the reception of this secret code by the MN, the MN and

the FN both have the same SCC_1 and SCC_2 and are able to generate a secret key using these two codes, S_{MN_j} and T_{FN_i} as

$$Secret\ Key = SCC_1 \bullet SCC_2 \ \ mod \ \ (S_{MN_j} \bullet T_{FN_i}) \tag{8}$$

Once a secret key is established between the CH and each MN, the CH has assigned a Shared Secret Code (SSC) to its all member MNs. This shared secret code is updated both periodically and when a MN compromission is detected. Since the MNs move in the network to perform their duties, they may need to establish a secure communication link also with neighboring MNs, possibly very frequently due to their movement within the network. In order to keep track of their neighboring MNs, each MN broadcasts a short range Hello message to know about its neighboring MNs. To establish a secret key with a neighboring MN, both MNs will share their secret communication code IDs assigned to them as P_{MN}. Now both the MNs will find the maximum number of shared codes with one another and will generate a secret key using all of them as

$$Secret\ Key = \prod_{l=1}^{f} SCC_{1l} \ \ mod \ \ SSC \tag{9}$$

where 'f' represents the total number of common secret communication codes. Since the distributions of the SCC_1 codes to the MNs is random and probabilistic, two neighboring MNs might not have any secret communication code in common. In this case, to avoid any discontinuity, the MNs will use the assigned Shared Secret Code (SSC) from their common CH and their IDs to establishment a secret key with its neighboring MNs. For example, if MN_m wants to establish a secret key with MN_n but these two nodes do not have any common secret communication code (SCC), then they establish a secret key by first calculating and sharing L and K with each other as

$$L = prand(ID_{MN_n}) \bullet S_{MN_m} \bullet C_{auth_m} \bullet SSC \ \ mod \ \ G \tag{10}$$

$$K = prand(ID_{MN_m}) \bullet S_{MN_n} \bullet C_{auth_n} \bullet SSC \ \ mod \ \ G \tag{11}$$

$$Secret\ key = L \bullet K \ \ mod \ \ SSC \tag{12}$$

4 Security Evaluation

4.1 Denial of Service Attack

In this section we describe some kind of Denial of Service attacks (DoS attacks) that can be brought against our proposed scheme, as well as possible counter measures. The main objective of DoS attacks is to make the resources unavailable to an intended user of the network.

1. *FN Hello messages:* The first possible DOS attack against the proposed scheme is to broadcast Hello messages pretending to be a FN of the network to exhaust the resources of the MNs. Since each Hello message is signed by the private key of the FN, MNs will verify it using the public key of that FN. Since the adversary FN is not an authentic node, the MN would not be able to verify that Hello message and once a MN detects this attack, it will inform its other neighboring authentic FNs. The authentic FNs would then inform the BS and neighboring MNs about this fake FN ID so that they can avoid the messages from that node.
2. *MN Hello messages:* When a MN finds its current CH signal strength value below a threshold value, it starts broadcasting the MN Hello messages to know about its new neighboring FNs. The attacker can launch such MN Hello message broadcast attack by introducing a fake MN. Since the MN Hello broadcast message is also signed by the MN private key, the new FNs first verify it by using the MN public key. This would not be possible for a fake MN. Thus the FNs inform the BS and other neighboring FNs about this malicious MN.

4.2 Sybil Attack

Sybil attacks are those in which a malicious node illegitimately taking on multiple identities. We call the nodes performing these attacks as sybil nodes. Sybil attacks can be of different forms e.g. using direct or indirect communication and fabricated or stolen identities. In the direct communication sybil attacks, a Sybil node communicates directly with a legitimate node. But since, in the proposed scheme, the sybil node is first authenticated by sending a message signed with its private key, the FN would not be able to authenticate it. In the indirect communication sybil attacks, malicious node (who deploy sybil nodes in the network) becomes a router for forwarding the communication to the Sybil node from the FN which is not possible in the proposed scheme because each MN is the end user of the network. In the fabricated sybil attacks, the attacker assigns an unuse identity to the sybil node. In this case, this sybil node needs to authenticate itself to the FNs which would again not be possible in the proposed scheme as described above. Stolen identity based sybil attacks are very dangerous in such resource constrained networks. But this type of sybil attack does not affect the proposed scheme because each communication is encrypted with the key agreed already with the original node having this ID, and the sybil node does not have these keys.

In the key pre-distribution approach, if every MN is assigned KP_{MN} keys and every FN is assigned KP_{FN} keys from a key pool of size KP_{main} and an attacker compromises 'c' nodes to create a compromised key pool of size 'n', then the probability of a sybil node to be successful created is

$$Pr_{sybil\ node} = \sum_{t=1}^{KP_{MN}} \frac{\binom{n}{t}\binom{KP_{main}-n}{KP_{MN}-t}}{\binom{KP_{main}}{KP_{MN}}} \frac{\binom{KP_{main}-KP_{MN}+t}{KP_{MN}}}{\binom{KP_{main}}{KP_{MN}}} \tag{13}$$

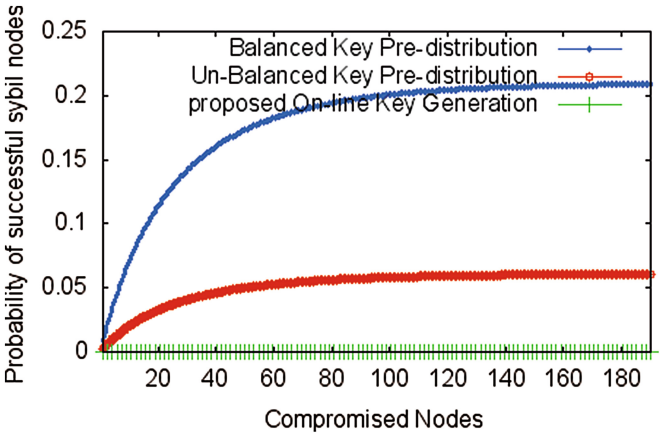

Fig. 2. Probability of generation sybil nodes

Figure 2 shows the probability of successfully generated sybil nodes in the proposed scheme compared with scheme [7,9].

5 Conclusion

In this paper, we proposed a new authentication and key management scheme for Heterogeneous Sensor Networks including mobile nodes. The proposed key management scheme is based on two different types of the key pools i.e. an authentication key pool and a communication key pool. Based on these pools, a key pre-distribution mechanism has been defined. The results showed that the two considered key pools provide better security. Furthermore, the proposed solution provides better network resilience against attacks compared to the other reference protocols considered.

References

1. Xiao, Y., Rayi, V., Sun, B., Du, X., Hu, F., Galloway, M.: A survey of key management schemes in wireless sensor network. Computer Commun. J. **30**(11–12), 2314–2341 (2007). Special Issue on Wireless Ad Hoc and Sensor Networks
2. Xu, K., Hong, X., Gerla, M.: An ad hoc network with mobile backbones. In: 2002 IEEE International Conference on Communications, ICC 2002, vol. 5, pp. 3138–3143 (2002). doi:10.1109/ICC.2002.997415
3. Eschenauer, L., Gligor, V.D.: A key management scheme for distributed sensor networks. In: Proceedings of the 9th ACM Conference on Computer and Communication Security, pp. 41–47, November 2002
4. Du, X., Xiao, Y., Guizani, M., Chen, H.H.: An effective key management scheme for heterogeneous sensor networks. Ad Hoc Netw. **5**(1), 24–34 (2007)
5. Perrig, R., Szewczyk, J., Tygar, V., Culler, D.E.: Spins: security protocols for sensor networks. ACM Wireless Netw. **8**(5), 521–534 (2002)

6. Chan, H., Perrig, A., Song, D.: Random key predistribution schemes for sensor networks. In: 2003 Proceedings of Symposium on Security and Privacy, pp. 197–213, 11–14 May 2003. doi:10.1109/SECPRI.2003.1199337
7. Liu, F., Rivera, M.J.M., Cheng, X.: Location-aware key establishment in wireless sensor networks. In: IWCMC 2006 (2006)
8. Zhang, J., Sun, Y., Liu, L.: NPKPS: a novel pairwise key pre-distribution scheme for wireless sensor networks. In: 2007 IET Conference on Wireless, Mobile and Sensor Networks, CCWMSN 2007, pp. 446–449, 12–14 December 2007
9. Kim, K.T., Ramakrishna, R.S.: A Level-based key management for both in-network processing and mobility in WSNs. In: 2007 IEEE International Conference on Mobile Adhoc and Sensor Systems, MASS 2007, pp. 1–8, 8–11 October 2007. doi:10.1109/MOBHOC.2007.4428761
10. Chuang, I.-H., Su, W.-T., Wu, C.-Y., Hsu, J.-P., Kuo, Y.-H.: Two-layered dynamic key management in mobile and long-lived cluster-based wireless sensor networks. In: 2007 Wireless Communications and Networking Conference, WCNC 2007, pp. 4145–4150. IEEE, 11–15 March 2007. doi:10.1109/WCNC.2007.757
11. Maerien, J., Michiels, S., Huygens, C., Joosen, W.: MASY: management of secret keys for federated mobile wireless sensor networks. In: 2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 121–128, 11–13 October 2010. doi:10.1109/WIMOB.2010.5644977
12. Lu, K., Qian, Y., Hu, J.: A framework for distributed key management schemes in heterogeneous wireless sensor networks. In: 2006 25th IEEE International Performance, Computing, and Communications Conference, IPCCC 2006, pp. 7–520, 10–12 April 2006. doi:10.1109/.2006.1629447
13. Du, X., Xiao, Y., Ci, S., Guizani, M., Chen, H.-H.: A routing-driven key management scheme for heterogeneous sensor networks. In: 2007 IEEE International Conference on Communications, ICC 2007, pp. 3407–3412, 24–28 June 2007. doi:10.1109/ICC.2007.564
14. Rabin, M.O.: Digitalized signatures and public-key functions as intractable as factorization. Technical report MIT/LCS/TR-212, Laboratory for Computer Science, MIT (1979)