A Secure Key Management Framework for Heterogeneous Wireless Sensor Networks

Mahdi R. Alagheband, Mohammad Reza Aref

A Secure Key Management Framework for Heterogeneous Wireless Sensor Networks

Mahdi R. Alagheband[1] * and Mohammad Reza Aref[2]

[1] EE Department, Science and Research branch, Islamic Azad University,
Tehran,Iran
m.alaghband@srbiau.ac.ir
[2] EE Department, ISSL Laboratory, Sharif University of Technology, Tehran,Iran
aref@sharif.edu

Abstract. A Wireless sensor network (WSN) is composed of numerous sensor nodes with both insecurely limited hardware and restricted communication capabilities. Thus WSNs suffer from some inherent weaknesses. Key management is an interesting subject in WSNs because it is the fundamental element for all security operations. A few key management models for heterogeneous sensor networks have been proposed in recent years. In this paper, we propose a new key management scheme based on elliptic curve cryptography and signcryption method for hierarchical heterogeneous WSNs. Our scheme as a secure infrastructure has superior sensor node mobility and network scalability. Furthermore, we propose both a periodic authentication and a new registration mechanism in our scheme due to prevention of sensor node compromising. Also, the proposed scheme does not increase the number of keys in sensor nodes and has a reasonable communication and computation overhead compared with the other schemes.

Keywords: Key management, Heterogeneous sensor network, Signcryption , Elliptic curve cryptography, Authentication.

1 Introduction

A wireless sensor network (WSN) has ability to monitor and control events in a specified environment with the aid of numerous sensor devices. However, these sensor nodes (SNs) have noticeable constraints on energy, computation and bandwidth resources. Despite cited restrictions, WSNs have unique characteristics such as SN mobility, large scalability, limited resources, special traffic patterns and uncertain to many types of attacks. The structure of WSNs divides into two kinds: homogeneous and heterogeneous on the whole. All SNs are similar to each other and are deployed in a flat architecture in homogeneous WSNs, while in heterogeneous both are two or more kinds of sensors are defined and the whole of SNs are separated in some clusters. Hence, not only does the

* This work was supported in part by Iran National Science Fund (INSF)-cryptography chair, and in part Iran Telecommunication Research Center (ITRC).

average of communication overhead and energy consumption decrease, but also the network scalability and performance increase in heterogeneous WSN [1].

Due to the fact that WSNs are susceptible to many attacks and have widespread constraints, the design of security mechanisms is highly important. Key management is the first crucial function to achieve security objectives because sensor nodes and cluster leaders need valid common key to utilize cryptography mechanisms. According to SN technology development, the key management protocols are classified based on encryption techniques in three categories, including symmetric, asymmetric and hybrid key management models [1].

Symmetric schemes that also called pre-distribution schemes are responsible for loading some keys into the sensor nodes prior to deployment phase, based on either their physical or wireless interfaces. These schemes suffer from some problems such as probabilistic key distribution between SNs, non-scalability after deployment, weakness against node compromising, lack of mobility and high communication overhead [2, 3].

Asymmetric schemes use both elliptic curve cryptography (ECC) and identity based cryptography (IBC) in recent years [6]. Asymmetric models are more flexible but very heavyweight in the sensor networks. The recent progress in ECC and IBC awards new opportunities to apply public key cryptography in WSNs. Since ECC keys are defined on an additive group with *160-bit* length, this family of public key cryptography is as secure as RSA keys with 1024-bit length [4]. Also, recent implementation on MICA2 or MICAz mote has approved the feasibility of ECC in WSN [4, 5].

Hybrid schemes have been designed based on heterogeneous WSNs with different kinds of nodes. Despite distinction among base station, cluster leaders and SNs, each element performs distinctive responsibility in hybrid hierarchical architecture. As computational cost of cluster leaders is more than SNs, cluster leaders usually have more obligations such as aggregation, routing, control and cluster leading.

In this paper, we present a secure hybrid key management infrastructure in hierarchical heterogeneous WSN (HHWSN). ECC is used among cluster leaders and base station in the proposed scheme. Moreover, a special mechanism is used in the clusters for periodic authentication and SN mobility among the clusters. The contributions of this paper are four folds. i) In order to achieve complete security, a specific signcryption method with forward security characteristic is utilized in inter-cluster communication. ii) Our scheme supports SN mobility to move among the clusters. iii) We design a periodic authentication to prevent SN compromising. iv) A new registration model is designed for SNs enrollment after network deployment. The rest of the paper is organized as follows: section 2 describes the preliminaries which are practical for understanding the proposed protocol and related works. In section 3, some related works are analyzed. In section 4, we propose the new key management scheme. In section 5 we compare the scheme with a few related schemes. Section 6 gives comparison result. Finally, conclusion is presented.

2 Preliminaries

In this section, we describe some essential points used in this paper. BS should select some primitive parameters in initialization phase. F is the selected elliptic curve over finite field q: $y^2 = x^3 + ax + b \pmod q$. G is base point of elliptic curve F with order n and O is point of F at infinite. n is the order of point G, where n is a prime,$n \times G = O$ and $n > 2^{160}$. (The symbol '×' denotes the elliptic curve point multiplication [6]. For simplicity, a list of notations used in the paper is shown in Table 1.

Table 1. List of notations

Notation	Description	Notation	Description
BS	Base station	P_{bs}	BS's Private key
CL	Cluster leader	U_{bs}	BS's public key
SN	Sensor node	K_N	Network key [128 bit] (just for registration)
Adjacent CL	Neighbour leaders of CL	K_{SN_i}	Sensor node key
ID_{cl} or ID_{SN}	Identity of CL or SN	K_{cl}	Cluster key
P_{cl_i}	CL$_i$'s Private key	Sgn	Signcryption algorithm
U_{cl_i}	CL$_i$'s Public key	$t.s.$	Timestamp
t_{comp}	Least time duration for node compromising	t_{move}	Maximum movement time for SN
$meta$	A public and fixed message	H	A lightweight and secure one-way hash function
$E_k(.)/D_k(.)$	Lightweight symmetric encryption/decryption algorithm with key k		

The security of asymmetric and hybrid key management especially in BS-CL links are based on ECDLP (Elliptic Curve Discrete Logarithm Problem)that is a hard problem until now [6]. Furthermore, the security of SN-SN links is supported by lightweight symmetric cryptography generally [7].

BS generates public-private keypairs based on ECDLP. These keys are assigned to all nodes in the asymmetric key management or just CLs in the hybrid key management schemes. BS performs following terms for key generation.

- Choose P a random number as a private key $P \in [1 \quad q-1]$.
- Compute $U = P \times G$ as a public key.
- Embed (P, U) in node securely after deployment and save it in its database.

After this phase, every CL in heterogeneous WSN has a unique pairwise key.

Key generation aside, signcryption is used in the paper particularly in CL-BS links too. Not only does signcryption technique combine the digital signature and encryption algorithms to achieve authentication and confidentiality but also signcryption has lower computation and communication overhead. Thus the utilization of signcryption in WSNs is highly profitable. Besides, we utilize the signcryption scheme (Sgn) with extra characteristics such as public verifiability

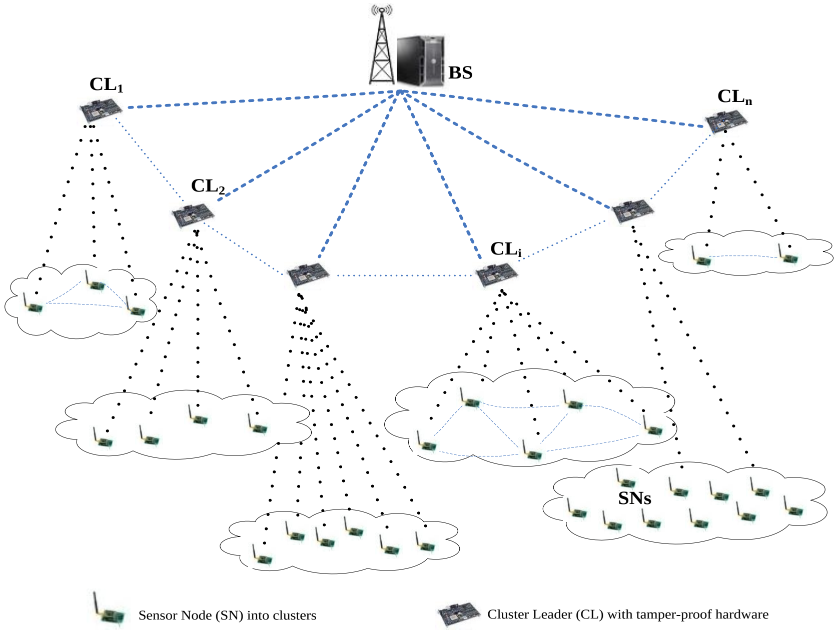
BS
CL1
CL2
CLi
CLn
SNs
Sensor Node (SN) into clusters
Cluster Leader (CL) with tamper-proof hardware

Fig. 1. A sample hierarchical heterogeneous sensor network model

and forward secrecy in our proposed scheme [8, 9]. If a cluster leader is revealed, the authenticity of past transmitted messages from the compromised CL to BS is valid because of forward security attribute. It protects the authenticity of messages even though the private key of the sender is disclosed [8, 10]. Every CL has both the public key of other CLs and BS. A typical signcryption model with cited attributes will be used in our scheme. The details of some reasonable signcryption scheme for WSN have been explained in [8, 10].

A HHWSN is composed of a BS as a sink node, a small number of CLs and numerous SNs that classified in clusters (Fig. 1). Number of CLs is not noticeable compared with density of SNs. The following assumptions are noted in our network model:

1. SNs are not equipped with tamper-proof hardware due to inherently constraints.
2. CLs have better resources and more responsibility compared with SNs. Since the ability of asymmetric cryptography computation is absolutely essential for them. Therefore, every CL has unique public-private keypairs and are equipped with tamper-proof hardware.
3. Each SN and CL have a unique ID (ID_{cl_i} or ID_{SN_i}).
4. BS does not have any restriction on computation, storage or power supply. BS know all CLs public key (U_{cl_i}) and SNs keys (K_{SN_i}).
5. CLs are static but SNs are mobile.

All CLs and SNs are usually deployed in uncontrolled regions without strict supervision. Every cluster of SNs sense environments and send raw data to corresponding CL. Each CL aggregates information and routes it to the BS by respective protocols.

3 The analysis of related works

In this section, we demonstrate some considerable hierarchical heterogeneous key management schemes proposed until now and analyse their advantages and disadvantages [11–14].

Riaz et al. have proposed SACK [11] as a secure key management framework for a HHWSN. Every SN has a unique key with the BS and CLs have an extra key to communicate with BS and other CLs. Besides, all SNs in every cluster have a distinctive common key for secure intra cluster connection. One master key of 1024 bits is stored in each SN and CL after deployment in SACK. CLs and SNs use it to compute shared key after cluster formation. Furthermore, SACK has a revocation mechanism for compromised node.

But intruder can abuse master key to penetrate the network as. Since initial seeds for key generation are sent plainly after cluster formation in key assignment phase, the adversary can simply eavesdrop it. Now the newcomer malicious adversary with both compromised master key and eavesdropped seed can compute intra cluster key subsequently. Indeed, the security of the whole WSN will seriously be failed if just one SN is compromised. Besides, SACK has some other damaging problems in key generation algorithm. As authors have pointed out, a single polynomial can generate only 895 distinct keys. After 895 times, a Re-keying algorithm should be employed for solving this weakness. But SACK undergoes substantial communication and computation overhead with the Rekeying algorithm. Moreover, 1024 bits as a master key is partially heavy burden for the sensor nodes.

X. Du et al. [12] proposed a routing-driven key management scheme based on ECC (RDEC) in HHWSN. Although, every SN and CL has a pairwise private-public key based on ECC in RDEC, SNs do not have shared key with all neighbors in intra cluster connections. All SNs have a common key with just some neighbor SNs in the specific routes that the routing protocol has already defined to send data for BS. Each SN firstly sends Key-Request message to CL. Then the CL computes diverse shared key between every two neighbor SNs and sends it based on the defined route in RDEC scheme.

RDEC has some damaging feature. i) Every CL requires enormous storage space to save all SNs public keys for common key generation because SNs are clustered after deployment phase. This amount of storage space is ineligible for WSNs. ii) All SNs have a certain time to send the un-encrypted Key-Request message to CL. An adversary can replace the parts of Key-Request messages and deceives CL in the defined time because the Key-Request message is sent un-encrypted. iii) K_H is a pre-loaded symmetric key that is embedded in the newly-deploy SNs and CLs. An adversary can reveal K_H because the hardware of

SNs is not tamper proof. However, it is probable that the compromised SNs key is revoked but adversary can damage the network as a newcomer SN. Furthermore, after K_H revelation, RDEC does not have any mechanism to distinguish this catastrophe. iv) Every CL has keys related to all SNs after pre-deployment phase. Therefore, apart from pre-loaded SNs, any new SN cannot register its public key at the WSN based on RDEC scheme after deployment phase.

Mizanur and Khalil [13] have proposed another key management framework (PKAS) on pairing based cryptography. PKAS has tried to improve RDEC scheme based on IBC. Every CL or SN has an ID and two distinctly random numbers embedded in the pre-deployment phase. Each CL has IDs and random numbers of all SNs and authenticates its SNs in its cluster. Thus the information of clustering is prerequisite in PKAS.

In PKAS, although the random number of SNs is periodically updated by the BS and distributed to SNs via CLs, WSN should undergo enormous amount of communication overhead. The SACK's solution to solve this challenge looks better than PKAS's out because the cost of transmission is much more than the cost of computation. Moreover, each SN requires the nearest CL's ID for mutual authentication. So either every SN should save all CL's IDs or authentication should be run after cluster formation in PKAS scheme. Not only is saving of all CL's IDs very heavy for the feeble SNs but also the clustering information declines the network scalability and flexibility.

PIBK is another identity based key management protocol for HHWSN [14]. PIBK has been designed for a static network with fixed and location aware SNs that use IBC to establish pairwise keys. Each SN gets three keys (network key, cluster key and SN key) in pre-deployment phase. Then, every SN should communicate ID with neighbors in a restricted time duration (Bootstrapping time). After Bootstrapping time, all SN should save neighborhood IDs so that every two nodes can make shared secure key in their cluster.

4 The proposed framework

In this section, we describe our proposed key management infrastructure for HHWSN in six parts.

4.1 Key assignment in pre-deployment phase

Prior to initialization and cluster formation phase, some symmetric and asymmetric keys should be embedded in all SNs in pre-deployment phase. We have used more strict security policies for CL-BS links because of the high emphasis on communication between CLs and the BS. Therefore, public key cryptography is tapped to achieve a higher level of security in WSN.

As it was pointed out in Table 1, U_{cl} is the public key and P_{cl} is the private key of any CL ($U_{cl} = P_{cl} \times G$). The P_{cl} is called the discrete logarithm of the U_{cl} to the base G. Also CLs have a common symmetric key as a group key (K_{cl}) for secure communication together. The key will be useful in periodic authentication.

Likewise, BS has two keys U_{bs} and P_{bs} ($U_{bs} = P_{bs} \times G$). P_{bs} will be secret key for BS that CLs and SNs do not know it forever. U_{bs} is embedded in CLs to execute signcryption algorithm after deployment phase. Indeed, CL computes the signcryption of messages by U_{bs} and P_{cl} , sends it to BS completely secure and verifies the authenticity of BS with the aid of U_{bs}.

On the other hand, all SNs have a common network key (K_N). This key just is used in the registration procedure after network deployment that will be explained in section 4.3. In order to perform periodic authentication, every SN has an exclusive key with BS (K_{SN_i}) which BS knows both K_{SN_i} and ID_{SN_i}.

4.2 Inter-cluster communication

The structure of heterogeneous WSN emphasizes the importance of security in CL-BS and CL-CL links. The network needs a method to communicate securely between BS and CLs prior to SN's registration. If an adversary discloses either a CL-BS or a CL-CL links, the network security will be damaged increasingly. Hence, every CL as well as BS has distinct public and private pairwise keys.

Since message confidentiality and sender's authentication in CLs-BS links have a particular emphasis, digital signature and ECC have been used in many key management schemes to drive confidentiality, integrity and authenticity [11–16]. In contrast, according to the computational and memory constraints in WSN, it is not acceptable to utilize signature-then-encryption method to keep message confidentiality and authenticity permanently among WSN's nodes.

4.3 SN's registration

After WSN deployment, SNs should find the nearest CL for registration into its cluster. Fig. 2 illustrates the registration procedure among SN, CL and BS. A SN will be enrolled in the nearest CL by the following steps:

1. SN sends $\alpha = ID_{SN_i}$ and $\beta = H_{K_N}(ID_{SN_i})$ to the nearest CL by means of keyed one-way hash function (H).
2. CL verifies whether $H_{K_N}(\alpha)$ is equal to β. If it is true, goes to step 3, otherwise rejects the message and alarms to BS.
3. CL computes $Sgn(ID_{SN_i}, t.s.)$ with its private key and sends it to BS (Sgn is the Signcryption algorithm).
4. As soon as Unsigncryption and verification phase are done, BS responds to CL by $Sgn(ID_{SN_i}, K_{SN_i}, t.s.)$.
5. CL saves ID and K_{SN_i} after verification.
6. CL uses a lightweight symmetric encryption algorithm to generate ciphertext $\gamma = E_{K_{SN}}(meta \parallel K_{cl_j})$, where *meta* is a public and fixed passage that all nodes know it.
7. The SN computes $D_{K_{SN_i}}(\gamma)$ where the secret key K_{SN_i} has been embedded in SN at pre-deployment phase. SN verifies if the first part of $D_{K_{SN_i}}(\gamma)$ is equal to *meta*. If it is true, SN generates $K_N^{'}$ from K_N with a lightweight one way hash function. Thus, the computation of K_N from $K_N^{'}$ is impossible.

The beginning of registration procedure without K_N is impossible. Therefore, in order to prevent disclosure of K_N, each SN should change K_N to $K_N^{'}$ after membership in a cluster immediately. Since K_N is revealed entirely after registration, the adversary cannot compromise a SN subsequently. The transformation is based on a one-way function and computation K_N from $K_N^{'}$ is impossible. Indeed, if a registered SN is compromised imaginatively, the adversary cannot take part in registration procedure as a legal node because he has just achieved to $K_N^{'}$ and K_N was completely deleted. Every newcomer SN can use K_N to do registration procedure in a defined range of time after WSN deployment. The time duration is not enough for newcomer SN compromising by means of adversary.

Moreover, it is plainly visible that the transformation does not impose constraint on network scalability and new SNs are added during WSN's life, as all nodes derive K_N to $K_N^{'}$. The adversary can obtain K_{cl} but he is unable to disorder secure connections between SNs and CLs with the aid of periodic authentication explained in the next section.

4.4 Periodic authentication and SN mobility

One of the crucial parts of the proposed key management infrastructure that is usually ignored in heterogeneous WSN is "periodic authentication" [11–14]. Since SNs in contrast to CLs are not equipped with tamper-proof hardware, it is completely probable that a SN is compromised after deployment. Although K_N as unique parameter for registration has been deleted, the adversary can grab K_{cl} and K_{SN_i} readily and disorder the SN-CL and SN-SN links. Thus, the proposed key management scheme has a periodic authentication to preserve SNs against compromising as well as to support SN's mobility among clusters especially in liquid environments. Fig. 3 illustrates the periodic authentication mechanism between SN and CL in every cluster. Every CL should regularly authenticate the SNs which have registered in its cluster. The period of this mechanism (t_{comp}) depends on the duration of node compromising. WSNs usually utilize ZigBee or IEEE 802.15.4 platform for communication. Since the time duration compared with the period of ZigBee's MAC layer is negligible, the periodic authentication does not impose extra overhead [17]. Furthermore, the overhead of periodic authentication compared with overhead of other policies such as key updating in SACK, RDEC, PKAS and PIBK is rational. According to the Fig. 3, CL$_j$ sends the query for all registered SN periodically in its cluster. SN$_i$ checks the truth of query. SN$_i$ sends flow 2 if the flow 1 be true. As soon as the CL receives the flow 2, it computes $H_{K_{SN}}(K_N^{'})$ and checks with α inasmuch as just CL knew both K_{SN_i} and $K_N^{'}$ after SN registration. The SN is confirmed for next t_{comp} period provided the flow 2 is verified. Otherwise, CL will alarm to BS that the mentioned SN is uncertain.

In the normal conditions just phase 1 and 2 (Fig. 4) are performed but if CL does not receive any message in the defined time, the SN has presumably moved to another cluster. Thus the CL sends $Sgn(K_N^{'}, ID, ProbeRequest)$ to adjacent CLs to track the SN (phase 5). Since every CL has U_{cl} of other CLs, they can

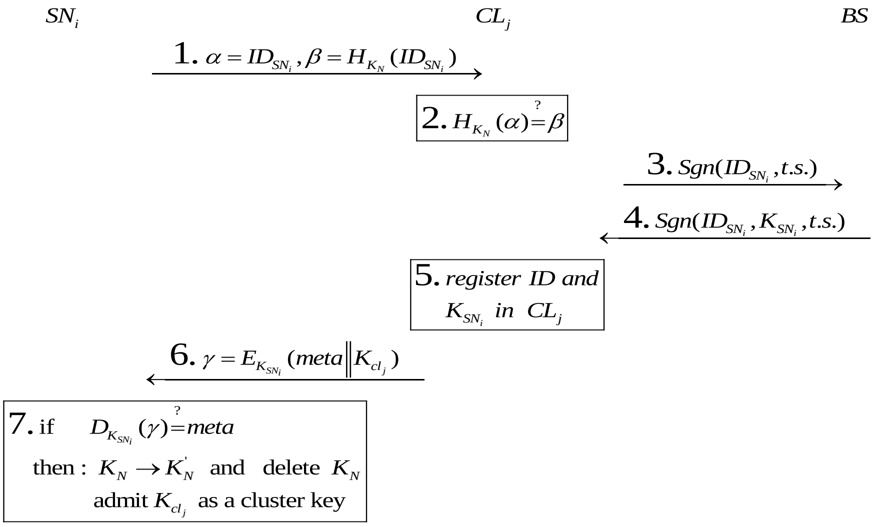
SN_i
CL_j
BS
1. $\alpha = ID_{SN_i}, \beta = H_{K_N}(ID_{SN_i})$
2. $H_{K_N}(\alpha) \overset{?}{=} \beta$
3. $Sgn(ID_{SN_i}, t.s.)$
4. $Sgn(ID_{SN_i}, K_{SN_i}, t.s.)$
5. register ID and K_{SN_i} in CL_j
6. $\gamma = E_{K_{SN_i}}(meta \| K_{cl_j})$
7. if $D_{K_{SN_i}}(\gamma) \overset{?}{=} meta$
then : $K_N \rightarrow K_N^{'}$ and delete K_N
admit K_{cl_j} as a cluster key

Fig. 2. SN registration procedure with CL and BS cooperation

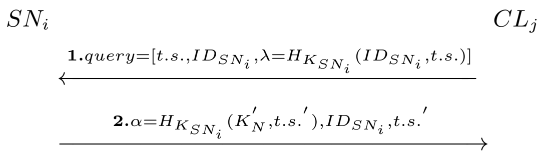
SN_i
CL_j
1.$query=[t.s.,ID_{SN_i},\lambda=H_{K_{SN_i}}(ID_{SN_i},t.s.)]$
2.$\alpha=H_{K_{SN_i}}(K_N^{'},t.s.^{'}),ID_{SN_i},t.s.^{'}$

Fig. 3. The periodic authentication mechanism between every SN and CL inside clusters

do unsigncryption algorithm. If the SN moved to another cluster, one of the adjacent CLs will find it in the defined time (t_{mov}). All adjacent CLs perform authentication mechanism again to find the moved SN. If an adjacent CL finds the moved SN and $t < t_{mov}$, it sends a report to the prime CL. Otherwise, the prime CL supposes that the lost SN is compromising when $t > t_{mov}$. In this condition, the lost SN should be revoked from the whole of WSN. The prime CL announces to adjacent CLs that the registered node with ID_{SN_i} is revoked. Also the prime CL sends the revocation message accompanied by ID_{SN_i} to BS and other registered SNs in its cluster.

According to this model an adversary cannot enter to WSN by node compromising because K_N as only registration key had been deleted and ID_{SN_i} had been revoked in WSN. Although each node must share a key with BS, the all

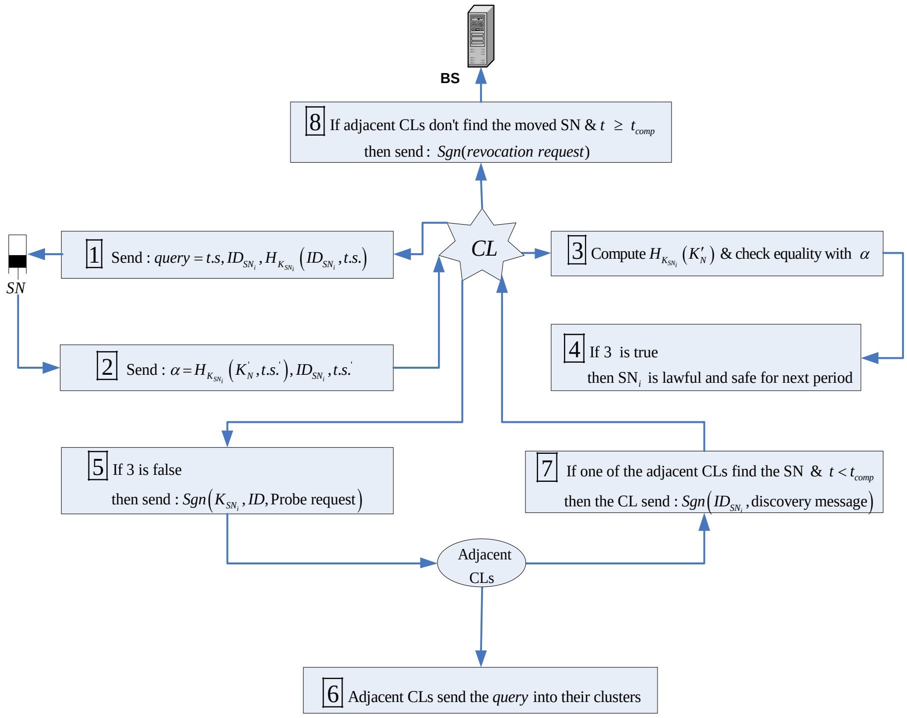
BS
8 If adjacent CLs don't find the moved SN & $t \geq t_{comp}$
then send : $Sgn(revocation\ request)$
CL
SN
1 Send : $query = t.s, ID_{SN_i}, H_{K_{SN_i}}(ID_{SN_i}, t.s.)$
2 Send : $\alpha = H_{K_{SN_i}}(K'_N, t.s.'), ID_{SN_i}, t.s.'$
3 Compute $H_{K_{SN_i}}(K'_N)$ & check equality with α
4 If 3 is true
then SN_i is lawful and safe for next period
5 If 3 is false
then send : $Sgn(K_{SN_i}, ID, \text{Probe request})$
7 If one of the adjacent CLs find the SN & $t < t_{comp}$
then the CL send : $Sgn(ID_{SN_i}, \text{discovery message})$
Adjacent CLs
6 Adjacent CLs send the *query* into their clusters

Fig. 4. The flowchart of periodic authentication mechanism due to prevention of sensor node compromising and mobility among clusters

authentication processes conduct without the aid of BS. Indeed, CL just informs BS in phase 8 (Fig. 4) for revocation provided the SN does not respond in time.

On the other side, "SN mobility" is one of the most striking features of the periodic authentication and WSN can be easily deployed in liquid and unsteady environments easily. The moved SN can communicate with new CL after authentication because the new CL has received K_{SN_i} from the prime CL. Fig. 4 depicts vividly the mentioned mechanism. Although the process seems complicated, it is lightweight and straightforward because its period compared with similar policies in other key management frameworks in heterogeneous WSN (e.g. key updating) is logical and profitable.

4.5 Intra-cluster communication between SNs

In this section, the model of intra-cluster communication between SNs in every cluster is described. Every SN has three embedded keys $(K'_N, K_{SN_i}, K_{cl_j})$ as well as ID_{SN_i} after cluster formation. K'_N was used at periodic authentication. All registered SNs in a cluster have a common cluster key (K_{cl_j}). However, K_{cl_j} will be changed provided SN_i moves to other cluster. Therefore, they have mutual secure communication. Although an adversary can eavesdrop intra-cluster links and compromise K_{cl}, he cannot disclose any message and disorder intra-cluster transactions since the ID of revealed SN is revoked with the aid of periodic authentication mechanism and mutual intra-cluster communication without valid ID is impossible.

Also intra-cluster links need a mechanism to achieve authenticity. In contrast to inter-cluster links, the computation and communication overhead of digital signature and signcryption is irrational in intra-cluster links among limited SNs. Hence, each SN should accompany its ID in every encrypted message in order that receiver recognizes the identity of sender (Eq. 1). It is plainly visible that every SN can find its neighbors after some transactions.

$$SN_i \xrightarrow{ID_i, E_{K_{cl}}(ID_i, m)} SN_j \tag{1}$$

As we indicated in section 4.4, our proposed scheme can detect compromised SNs, while the attacker is compromising it. Although it is undeniable that the adversary can obtain K_{cl}, the periodic authentication mechanism finds this malicious node at once.

5 Security analysis and comparison

In this section we both compare our scheme with the last schemes on heterogeneous WSNs and demonstrate how it is resistant on important attacks. Firstly, we define well known attacks on WSNs and explain how our proposed scheme can prevent them.

Node Capture Attack : In node capture attack, an adversary gains full control over sensor nodes through direct physical access [14]. According to the importance of CL-BS and CL-CL links, not only is public key cryptography (signcryption method with forward secrecy) used in BS-CL and CL-CL links but also the hardware of CLs is defined tamper-proof in our scheme. An adversary cannot compromise a CL and cannot do manipulation, replay and impersonation attacks, inasmuch as he should solve ECDLP. Furthermore, if a CL's private key is compromised imaginatively, the adversary cannot still reveal previous plaintexts from signcrypted messages because of forward secrecy.

On the other side, SN compromising is highly probable because the hardware of SNs is not tamper-proof. In order to increase persistence against the defect, registration mechanism and periodic authentication were designed to prevent penetration of an adversary to the WSN. Hence intra-cluster links will be secure.

In the worst case, if an adversary compromises a SN after deployment in $t > t_{comp}$, the adversary cannot impersonate a legal SN with its compromised ID because the CL has already revoked it via periodic authentication mechanism. Although the adversary grabs K_{cl}, ID_{SN_i} and $K_N^{'}$, he does not have enough time to send correct response to CL in the authentication protocol. Thus the CL revokes the compromised SN immediately. Moreover, if a random ID is chosen by adversary, the CL will reveal it in next periodic authentication as well.

If the adversary obstructs the flows 1 or 6 in Fig. 2, he will have enough time to compromise the SN but he cannot generate desyncronization attack because K_N has changed to $K_N^{'}$ in the last stage of SN's registration mechanism and the adversary cannot take part in registration procedure with the aid of $K_N^{'}$. On the other side, the maximum time duration for registration into a cluster to prevent this disturbance is bootstrapping time (t_{boot}). When t_{boot} is finished, all SNs should have been registered. Otherwise unregistered SNs will delete K_N and will send out at the WSN practically. Since the time requirement for registration is very shorter than t_{boot}, this policy has not decreased the throughput of network. Moreover, our scheme is extensible and it is possible to add new SNs during the life of WSN. Although the registered SNs do not have K_N, the new SNs join a cluster with the aid of K_N, compute $K_N^{'}$ and then delete K_N.

Replay Attack: An adversary can record ID_i ans $H_{K_N}(ID_i)$ (flow 1 in Fig. 2) in one location and sends it again either there or another location. Since BS has verified the ID_i previously, the adversary cannot introduce itself as a trusted SN to CL and BS. Also in authentication protocol, upon receiving the response of SN at $t.s.^{'}$ (flow 2 in Fig. 4), CL verifies whether $t.s.^{'} - t.s. \leq \triangle T$ for prevention of replay attack. If it holds, SN_i will be safe and valid for next period. If an adversary reveals K_{SN_i} , the SN with K_{SN_i} is revoked immediately based on periodic authentication.

Message Manipulation Attack: In this attack, an adversary may drop, change, or even forge exchanged messages in order to interrupt the communication process but he cannot manipulate messages in our proposed scheme because an adversary is not a valid node at all. The ways of this attack are three aspects. i) It is probable that an adversary manipulates query flow in periodic authentication (Fig. 3) but the SN checks the equality between α and $H_{K_{SN_i}}(K_N^{'}, t.s.^{'}, ID_{SN_i})$ and then SN will realize this disturbance immediately because the adversary does not have the SN's key and cannot impersonate SN without K_{SN_i}. ii) Despite the fact that the adversary knows *meta*, if the adversary modifies flow 6 (Fig. 2), the SN will not admit the received K_{cl_i} as the cluster key. The adversary cannot reveal K_{SN_i} in t_{boot} duration. iii) All CL-CL and CL-BS links are resistant to every kind of manipulation or impersonation attacks as they are based on Signcryption method.

Masquerade Attack: In this attack, an adversary can pretend to be a valid node and participate in the network communication. In our proposed scheme, all the nodes in the network are authenticated to each other along the way. Thus, the adversary cannot pretend to be valid nodes and cannot exchange the wrong information among the valid nodes. Therefore, a masquerade attack is not applicable on our proposed protocol.

To sum up, we compare our proposed key management infrastructure with the SACK, RDEC, PKAS, PIBK schemes that have been designed based on HH-WSN. Our scheme has some unique predominant features including SN mobility, periodic authentication, preventative mechanism against SN compromising and utilization of signcryption rather than signature-encryption (Table 2).

Table 2. The comparison of five schemes (Enc.=Encryption, Sig.=Signature, Key Agr.=Key Agreement)

Scheme Feature	SACK[11]	RDEC[12]	PKAS[13]	PIBK[14]	Our scheme
Mobility	No	No	No	No	YES
Number of saved key in every SN	2+1 (1024 bit)	2	3	4	3
Situation of network after one node compromising	The Whole of WSN fail	More than one SN fail	Just the SN fail	Whole of WSN fail	Just the SN fail
Difference among SN and CL	No	Yes	Yes	No	Yes
Authentication	-	-	once	-	periodic
The type of used PKC	Enc.	Enc.+ Sig.	Enc.	Key Agr.	Sgn
The position of PKC	CL-BS	SN-CL & CL-BS	SN-CL & CL-BS	SN-CL & CL-BS	CL-BS
Scalability after network deployment	Yes	Yes	Yes	Yes	Yes
Clustering as a prerequisite for Key management	Yes	No	Yes	Yes	No

6 Conclusion

A few key management frameworks have been designed for HHWSN in recent years. In this paper we proposed a novel and secure key management infrastructure for HHWSN. Our proposed scheme has number of striking features, including ECC utilization just between CL and BS, using signcryption rather than encryption with signature by forward security and public verifiability characteristics, SN mobility, periodic authentication to prevent SN compromising and a unique SN registration model in clusters. Furthermore, SNs just have undergone light computation and power consumption.

References

1. J. Zhang, V. Varadharajan: Wireless sensor network key management survey and taxonomy. Journal of Network and Computer Applications 33, Elsevier, pp. 63-75 (2010)
2. L. Eschenauer, V.D. Gligor: A key management scheme for distributed sensor networks. Proceeding of the 9th ACM Conference on Computer and Communication Security, pp. 41-47 November (2002)
3. Perrig A, Szewczyk R, Wen V, Cullar D, Tygar JD.: SPINS: security protocols for sensor networks. Proceedings of the 7th annual ACM/IEEE international conference on mobile computing and networking, pp. 189-99. (2001)

4. N. Gura, A. Patel, A. Wander, H. Eberle, S.C. Shantz: Comparing elliptic curve cryptography and RSA on 8-bit CPUs. Proceedings of the 6th International Workshop on Cryptographic Hardware and Embedded Systems, Boston,Massachusetts, (2004)
5. Malan DJ, Welsh M, Smith MD.: A public-key infrastructure for key distribution in Tinyos based on elliptic curve cryptography. First annual IEEE communications society conference on sensor and ad hoc communications and networks, (2004)
6. D. Hankerson, A. Menezes,S. Vanstone.: Guide to elliptic curve cryptography. Springer Verlag (2004)
7. Jongdeog Lee, Krasimira Kapitanova, Sang H. Son.: The price of security in wireless sensor networks. computer networks journal, Elsevier (2010)
8. R.-J. Hwang, C.-H. Lai, and F.-F. Su.: An efficient signcryption scheme with forward secrecy based on elliptic curve. Journal of Applied Mathematics and Computation, Vol.167, No.2, pp. 870-881, Elsevier (2005)
9. Y. Zheng, and H. Imai.: How to construct efficient signcryption schemes on elliptic curves. Information Processing Letters, Vol.68, pp.227-233, Elsevier (1998)
10. M. Alaghband, M. Soleimanipour, M. Aref,: A new signcryption scheme with forward security, Fourth information security and cryptology international conference (ISCISC), (2007)
11. Xiaojiang Du, Mohsen Guizani, Yang Xiao and Hsiao-Hwa Chen: A Routing-Driven Elliptic Curve Cryptography Based Key Management Scheme for Heterogeneous Sensor Networks. IEEE transaction on wireless communications, Vol. 8, No. 3, (2009)
12. R. Riaz,A. Naureen, A. Akram, A.Hammad K. Hyung Kim, H. Farooq.: A unified security framework with three key management schemes for wireless sensor networks. International journal Computer Communications 31, pp. 4269-4280 (2008)
13. Sk. Md. Mizanur Rahman Khalil El-Khatib.: Private key agreement and secure communication for heterogeneous sensor networks. Journal of parallel and distributed computing 70 , pp. 858-870, (2010)
14. Manel Boujelben,Omar Cheikhrouhou, Mohamed Abid ,Habib Youssef.: A Pairing Identity based Key Management Protocol for Heterogeneous Wireless Sensor Networks. IEEE transaction on wireless communications conference, (2009)
15. Michael Collins, S. Dobson, Paddy Nixon: A Secure Lightweight Architecture for Wireless Sensor Networks. The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and technologies, IEEE computer society (2008)
16. Q. Pei, Lei Wang, Hao Yin, Liaojun Pang and Hong Tang.: Layer Key Management Scheme on Wireless Sensor Networks. Fifth International Conference on Information Assurance and Security, IEEE computer society (2009)
17. Paolo Baronti, Prashant Pillai, Vince W.C. Chook, Stefano Chessa, Alberto Gotta, Y. Fun Hu: Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards, Computer Communications 30, pp. 1655-1695, (2007)