An Efficient Authentication and Key Establishment Scheme for Heterogeneous Sensor Networks


Published in:
Proceedings of 2011 7th International Conference on Emerging Technologies (ICET)

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Publisher rights
© 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.
An Efficient Authentication and Key Establishment Scheme for Heterogeneous Sensor Networks

Sarmad Ullah Khan
Politecnico di Torino,
Corso Duca Degli Abruzzi 24,
10129, Torino, Italy
Email: Sarmad.Khan@polito.it

Rafiullah Khan
University of Engineering and Technology,
Peshawar, Khybar Pakhtunkhwa,
Pakistan
Email: Rafiuk7@gmail.com

Abstract—Wireless Sensor Networks (WSNs) consists of a large number of small and resource constrained devices. Due to their low cost, flexibility and self-organizing nature, these devices are used in various applications such as home and industrial automation, environmental and habitat monitoring and even in more security sensitive military applications in which security of the information in terms of both snooping and tampering is very important. Such applications introduce various security challenges to these resource constrained devices. To this aim, a proper authentication and key establishment is required. In this paper, we present an effective authentication and key establishment scheme for the heterogeneous sensor networks. The proposed solution is tested using the OMNET++ simulator and results show that it provides better network connectivity, consume less memory and has better network resilience against attacks.

Keywords—Heterogeneous Sensor Networks; Security; Authentication; Key Establishment

I. INTRODUCTION

Wireless Sensor Networks (WSNs) consist of a large number of small low power and resource constrained sensor devices called nods. Since wireless sensor networks operate in ad hoc manner, they attract the attention of numerous applications such as military, health care, industry automation, environmental and habitat monitoring. WSNs are usually deployed in an unattended, adverse or even in hostile environment and face a number of challenges, especially in implementing the security functionalities such as authentication, integrity, privacy, and medium access control. An efficient authentication and key establishment scheme is required to cope with these challenges and provide a secure data communication.

Many key management schemes have been proposed in the literature to address the tradeoff between the security and memory cost for the homogeneous and heterogeneous sensor networks. Most of the existing key management schemes are based on the symmetric key approaches while asymmetric key approaches are considered to be not suitable for the WSNs due to their computational complexity and energy consumption. For instance, asymmetric cryptography such as RSA and the Elliptic Curve Cryptography (ECC) (which have high computational cost, energy consumption and storage requirements) were considered to be not suitable for WSNs but recent studies showed that public key cryptography such as Elliptic Curve cryptography (due to small key size and low computational overhead) and Rabin’s scheme (due to fast encryption/decryption time compared to the RSA) might be feasible even in sensor networks [6][7]. Several alternate key management schemes have been developed for WSNs, such as symmetric key, key pre-distribution, plain text key exchange and master key.

The probabilistic key pre-distribution approaches [8,9,4,5] are considered to be the most promising approaches for the WSNs. To ensure the scheme works, a node must have a high probability of sharing at least one common key with its neighboring nodes. To further improve the network connectivity and reduce the energy consumption and memory cost, LEACH-Low Energy Adaptive Clustering Hierarchy protocol [3] was proposed for clustered based WSNs. Zhu [2] proposed LEAP-Localized Encryption and Authentication Protocol for the clustered based WSNs with the changing cluster head. The key pre-distribution techniques require a large storage space and have a tradeoff between the security and memory cost. This tradeoff is somehow reduced by introducing the heterogeneity [12] in WSNs such that WSNs consists of at least two different types of nodes having different capabilities in terms of computation, communication, memory and power. In this paper, we proposed an efficient authentication and key establishment scheme for the heterogeneous sensor networks and show that how the heterogeneity of WSNs is exploited to overcome the above mentioned tradeoff. The proposed scheme reduces the memory overhead by using an on-line key generation method based on ECC and provides better network resilience against node capture attacks. The rest of the paper is organized as follow: Section 2 provides the related work; Section 3 describes the proposed scheme while analytical and OMNET++ simulation results are discussed in Section 4 and finally Section 5 concludes our paper.

II. RELATED WORK

Since, cryptography is considered as the basis of any security architecture, a number of key management schemes have been designed to secure the communication between the nodes in the WSNs. Eschenauer and Gligor [8] proposed a probabilistic random key pre-distribution scheme for WSNs in which each node is assigned a key pool of randomly selected keys from a large key pool before the network deployment. Single common key sharing between the two nodes depend on the key sharing probability and the probability should be high enough so that the two neighboring nodes find at least one key in common in their assigned key pools and use it as a secret
key in order to secure their communication. A path key is established between the two nodes through the intermediate nodes if they do not have a common key. Chan [9] further improved this scheme by proposing that the two neighboring nodes must share at least \( q \) keys to establish a secret key. This scheme requires a large number of keys to store in each sensor node in order to meet the required level of network connectivity. To reduce the memory cost, Liu [6] presented a key establishment scheme by taking the advantage of node deployment knowledge coupled with Rabin’s scheme [1] to achieve a high degree of connectivity and network resilience against the node capture attacks. NPKPS pairwise key pre-distribution scheme [15] presented by Zhang for WSNs also focused on the reduction of memory cost and increasing the network connectivity. But all these schemes are developed for the static networks and the probabilistic approach may further reduce the connectivity in case of no key match and increase the communication overhead to get the key from BS.

To reduce the routing and mobility management overhead, level-based secure and energy efficient key management scheme for multicast communication exploiting in-network processing was proposed in [16]. To support mobility in WSNs, Chuang in [17] proposed a two layered dynamic key management scheme for WSNs. Poornima [10] proposed two key management schemes for clustered WSNs called Simple Secure Logical Ring (SSLR) and Burmester Desmedt Logical Ring (BDLR) to reduce the communication and computation cost while increasing the network resilience against node capture attacks. J. Maerien in [20] proposed MANagement of Secret keYs protocol (MASY) for mobile WSNs that allow the secure deployment of a key to a sensor node when it enters into an unknown network. But there is an assumption that a trust relationship exist between the node’s new entering network and its parent network (Company).

Asymmetric key pre-distribution (AP) scheme was presented in [12] for Heterogeneous Sensor Networks (HSNs) which significantly reduces memory overhead and provides better network connectivity and security compared with [8] for the homogeneous sensor networks. High capability nodes are assigned large number of keys \( m \), while low capability nodes are assigned less number of keys \( l \), \( m >> l \). Their analyses show that such approach provides better network connectivity and better network resilience against the node capture attacks. Saber [11] exploits the heterogeneity in terms of clusters and presented a level based key management scheme for HSNs providing high connectivity and low memory cost. Further, each cluster is divided into different levels based on their distance from the Cluster Heads (CHs) and each level has a separate seed for the key establishment. Sajid [14] discussed a key management scheme based on key pre-distribution for HSNs. In this scheme, instead of storing all the assigned keys in low-capabilities sensor nodes, only a small number of generation keys are stored. Two low capability nodes request a high capability node to discover a shared generation key between them and assign a random number to generate secret communication key. Juwei [13] presented group-based key management scheme for HSNs. Nodes are divided into groups. Inner groups use the random key distribution approach for secure communication links establishment while for secure intergroup communication, nodes need t-shares of pairwise numbers to reconstruct the secret key while \( t < w \) and \( w \) is the size of the group key. This scheme provides better connectivity, has low memory requirements and better network resilience.

Since all key pre-distribution schemes assume the probabilistic networks connectivity, an authentic node in the network might not be able to join the network due to lack of shared key. So there should be robust key management schemes that provide 100% network connectivity especially in sensitive applications. A routing-driven key management scheme is proposed in [7] for HSNs which is based on Elliptic Curve Cryptography (ECC). It provides 100% network connectivity and better security but has significant memory overhead in case of dense network.

### III. PROPOSED SCHEME

In this section, we present our proposed authentication and key establishment scheme for HSNs. Proposed network consists of a Base Station (BS) and more powerful High end Nodes (H-Nodes) and less powerful Low end Nodes (L-Nodes). The BS is assumed to be the most secure and authentic device of the network and will act as a trusted server for the H-Nodes while the H-Nodes are nominated as authentic nodes by the BS and serve as trusted servers for the L-Nodes and manage the L-Nodes authentication and key establishment. The main objective of the proposed scheme is to deal with authentication and key establishment in heterogeneous wireless sensor networks. Since the network consists of two different types of nodes excluding the BS, authentication phase is divided into two sub phases, (1) H-Nodes authentication phase, (2) L-Nodes authentication phase. In the H-Nodes authentication phase, all H-Nodes are authenticated by the BS and after the successful authentication, L-Nodes authentication phase is initiated by the authentic H-Nodes. After the authentication phases, key establishment phase starts among the L-Nodes and H-Nodes.

#### A. Key Pre-Distribution

For authentication and secure communication, different keys are required. Here we use the Elliptic Curve Cryptography (ECC) to generate the public/private key pairs for the BS and H-Nodes authentication. To generate the public/private key pair, we have an elliptic curve \( E \) over the Galois Field \( GF(m) \) \( (m \) is a large prime number) and a base point \( G \) of order \( q \) whose size is equal to \( m \). A node private key ‘\( R \)’ is randomly chosen prime number from \([1, \ldots, q-1]\) and a node public key \( Y \) is calculated as

\[
Y = RG, \quad \text{a point on } E.
\]

Each H-Node is assigned the public key of the BS \( (Y_{BS}) \), its own public/private key pair \((R_i,Y_i)\) where ‘\( i \)’ is node ID,
one way key generation hash function \( KH() \) [19] for the generation of secret key, two unique prime numbers \( Z_{li} \) and \( U_l \) from \([1,...,q-1]\). Each L-Node is assigned network public key \( (K_{plc} = K_{prt} \cdot G) \), two unique prime numbers \( Z_l \) and \( U_l \) from \([1,...,q-1]\), G, and \( KH() \) where \( K_{prt} \) is the network private key.

If \( J_l = J_{BS} \), the BS verifies the requested node. Then H-Node verify the MAC using the \( Y_{hi} \) and sends it to the BS along with the received signal strength and these messages include the H-Nodes IDs and a random nonce. Once these Hello messages received by the L-Nodes, they verify them using the \( K_{plc} \). Successful verification authenticates the H-Nodes to the L-Nodes. After that L-Nodes select their CHs. L-Nodes also keep a record of other neighboring H-Nodes, in case of more than one Hello messages reception, as backup CHs. Fig. 2 describes the virtual network organization of the proposed scheme.

### C. Cluster Formation

Since the clustered networks have more advantage than the un-clustered networks in terms of high availability of resources to the end devices, load balancing in the network and efficient data handling, hence we adopt the cluster based network approach for the proposed scheme and H-Nodes will act as Cluster Heads (CHs). During the cluster formation, all the H-Nodes periodically broadcast a Hello messages for a specific number of times signed by the \( K_{prt} \). These Hello messages help the L-Nodes in selecting their CH depending on the received signal strength and these messages include the H-Nodes IDs and a random nonce. Once these Hello messages received by the L-Nodes, they verify them using the \( K_{plc} \). Successful verification authenticates the H-Nodes to the L-Nodes. After that L-Nodes select their CHs. L-Nodes also keep a record of other neighboring H-Nodes, in case of more than one Hello messages reception, as backup CHs. Fig. 2 describes the virtual network organization of the proposed scheme.

### D. L-Nodes Authentication Phase

Once selected the proper authentic CH, L-Nodes authentication phase starts and it includes the following steps:

1. **Step 1**: L-Node calculate \( W_l = K_{plc} \cdot Z_{l} \cdot C_{li} \) and send it to the CH.
2. **Step 2**: CH calculates \( W_H = K_{prt} \cdot Z_{hi} \cdot C_{hi} \) and also calculate \( V_{hi} \) from the received \( W_l \) as \( V_{hi} = W_l \cdot C_{li} \cdot Z_{hi} \) and send \( W_H \) to the L-Node.
3. **Step 3**: L-Node calculates \( V_L = W_H \cdot C_{li} \cdot G \cdot Z_{li} \) and send it to the CH.
4. **Step 4**: Compare \( V_L \) and \( V_{hi} \) to accomplish the authentication as

\[
V_L = W_H \cdot C_{li} \cdot G \cdot Z_{li} = K_{prt} \cdot Z_{hi} \cdot C_{hi} \cdot C_{li} \cdot G \cdot Z_{li} \\
V_{hi} = W_l \cdot C_{li} \cdot Z_{hi} = K_{plc} \cdot G \cdot Z_{hi} \cdot C_{li} \cdot C_{li} \cdot Z_{li} = K_{plc} \cdot K_{prt} \cdot G
\]

Each L-Node also broadcast a Hello message to know about its other neighboring L-Nodes and their unique prime numbers \( U_l \). Once the L-Node gets its neighboring L-Nodes prime numbers, it informs the CH about its neighboring L-Nodes IDs.

### B. H-Nodes Authentication Phase

During the network initialization phase, first the BS authenticates all the H-Nodes of the network using the modified version of algorithm described in [21] which will act as trusted servers for the L-Nodes. The steps involved are:

1. **Step 1**: H-Nodes first calculate \( T_{hi} = R_{hi} \cdot Y_{BS} \cdot C_{hi} \) and then send the Hello messages which contains \( (T_{hi}, MAC_{BS}(T_{hi})) \) to the BS where \( C_{hi} \) is the random number from \([1,...,q-1]\).
2. **Step 2**: BS verifies the MAC using \( Y_{hi} \) and generates \( T_{BS} = R_{BS} \cdot Y_{hi} \cdot C_{BS} \) and send \( (T_{BS}, MAC_{BS}(T_{BS})) \) to the H-Node where \( C_{BS} \) is the random number from \([1,...,q-1]\).
3. **Step 3**: Now H-Node verify the MAC using the \( Y_{BS} \) and calculates \( J_{hi} = T_{BS} \cdot C_{hi} \mod G \), and send it to the BS and request for the network private key \( K_{prt} \).
4. **Step 4**: BS calculates \( J_{BS} = T_{hi} \cdot C_{BS} \mod G \), and compare \( J_{hi} \) with \( J_{BS} \) for the authentication.

If \( J_{hi} = J_{BS} \), H-Node is authentic node and BS sends \( K_{prt} \) and \( J_{BS} \) to the H-Node encrypted by the \( Y_{hi} \) which will be used for H-Node authentication to the L-Nodes. The use of the node IDs further ensure that the message is received from the correct requested node.
E. Secret Key Establishment and Management

For the secure communication between the CH and the L-Node, there must be a secret key. Each L-Node and each H-Node is assigned \( KH() \) for secret key generation before the deployment. Since L-Nodes send their neighboring L-Nodes IDs to their selected CHs during the authentication phase so each CH knows the neighbors of its each member L-Node. For secret key generation, CH selects randomly few neighboring nodes of an L-Node with whom it is establishing secret key and sends their IDs to that L-Node. Now both, L-Node and CH will use the prime numbers of those selected neighboring L-Nodes, their own prime numbers, \( V_L \) or \( V_H \) and \( KH() \) for secret key generation. For example, node A want to establish a secret key with it CH and it has five neighboring nodes (B,D,F,H,I) and three of them (F,H,I) are selected by the CH, then the secret key is generated as

1. \( O = U_{CH} \times U_A \times U_F \times U_H \times U_I \)
2. Secret key = \( KH(O, V_L \text{ or } V_H, G) \)

For secure communication between the L-Nodes, a secret key for them is generated by the CH. For instance, if a node A wants to establish a direct communication link with node B, it sends its ID A along with the ID B to its CH. Then the CH generates a secret key for them using their prime numbers, common neighboring L-Nodes prime numbers and \( KH() \). It sends the generated secret key to both the L-Nodes encrypted by the secret key shared with each of them. CHs also inform the BS about their member L-Nodes periodically to avoid the node replication attacks in the network.

IV. Analysis and Evaluation

In order to show the effectiveness of the proposed scheme in terms of network connectivity, memory cost, and network resilience against node capture attacks, we simulate the proposed scheme using the OMNET++ simulator. Each simulation consists of 500 L-Nodes and 16 H-Nodes deployed randomly in a network having the simulation area of dimensions 400m x 400m. We assumed that all the H-Node are one hop away from the L-Nodes and for that we kept the same transmission power of both the H-Nodes and the L-Nodes.

We compare the results of our proposed key management scheme with some of the existing key management protocols and the results show in better network connectivity, significant reduction in memory requirement, and better network resilience.

A. Network Connectivity

Eschenauer and Gligor presented a probabilistic key pre-distribution scheme in [8] for WSNs. Two neighboring nodes are required to store large number of keys to meet the required probability of connectivity i.e. sharing at least one common key. According to this approach, each node is assigned a key pool of size K randomly selected from a large key pool P before the network deployment. Then the probability of connectivity in terms of single key sharing is given by

\[
Pr[Conn] = 1 - \frac{(P - K)! \times (P - K)!}{P! \times (P - 2K)!}
\]

In order to increase the single key sharing probability, a large key pool K should assign to each node but increasing K increases the security risks in case of a node compromised attacks. Also, single key sharing probability can be increased by reducing the size of P but this again poses the same security risks. This issue is resolved by Du. X. in [12] by proposing unbalanced key pre-distribution scheme for heterogeneous WNSs which significantly improves the network connectivity and reduces the memory cost by assigning different size of key pools K and S to the L-Nodes and to the H-Nodes respectively (S>>K). The probability of sharing at least one common key in this approach is given by

\[
Pr[Conn] = 1 - \frac{(P - K)! \times (P - S)!}{P! \times (P - S - K)!}
\]

In our proposed scheme, we assign network public key \( K_{pub} \) to each L-Node and network private key \( K_{prv} \) to each H-Node and these two keys are used during the initial authentication and key establishments, so the connectivity of the network is almost 100% if H-Node remains authentic member of the network. Also, our proposed scheme consumes much less memory and provides better connectivity compared to both the balanced [8] and the unbalanced [12] key pre-distribution schemes. Fig. 3 shows the comparison of OMNET++ simulation results for the network connectivity of our proposed scheme with the [8], [12] and [13] proposed for the homogeneous and heterogeneous sensor networks.

B. Memory Cost

This section describe the comparison of the proposed solution with some existing key management schemes for HSNs in terms of total number of keys deployed in the network.

Xiaojiang [7] presented a routing driven key management scheme for heterogeneous WSNs based on ECC. Each L-Node is assigned its private key and the public keys of all H-Nodes while each H-Node is assigned the public keys of all L-Nodes, a pair of public/private keys for itself and a key for the newly
deployed sensors. The total keys deployed in the network are $(n_L + 3) \times n_H + 2n_L$, where $n_L$ and $n_H$ are the number of L-Nodes and H-Nodes respectively.

Since [8] and [12] are based on the probabilistic key pre-distribution, total number of keys assigned to each node depends on the required probability of connectivity. For example, each H-Node has the capability of making a maximum of $d$ connections with its neighboring L-Nodes out of $N_c$ neighbors. Then the required key sharing probability should be

$$Pr = d / N_c$$

For example, if $d = 30$ and $N_c = 38$, the single key sharing probability would be 0.80. It is clear from fig. 2 that for 80% key sharing probability, each node in [8] should carry 400 keys and each H-Node in [12] should carry 700 keys while each L-Node should carry 228 keys from the key pool size of $P=100000$.

The memory overhead of our scheme is $5n_H + 4n_L$. In the proposed scheme, each L-Node is loaded with one network public key, two prime numbers and generator G and each H-Node is loaded with 3-keys (i.e., BS public key, its own public/private key pair and $K_{prt}$) and two prime numbers. The required key sharing probability would be 0.80. It is clear from fig. 2 that for 80% key sharing probability, each node in [8] should carry 400 keys and each H-Node in [12] should carry 700 keys while each L-Node should carry 228 keys from the key pool size of $P=100000$.

The memory overhead of our scheme is $5n_H + 4n_L$. In the proposed scheme, each L-Node is loaded with one network public key, two prime numbers and generator G and each H-Node is loaded with 3-keys (i.e., BS public key, its own public/private key pair and $K_{prt}$) and two prime numbers. The memory overhead of our scheme is $5n_H + 4n_L$. In the proposed scheme, each L-Node is loaded with one network public key, two prime numbers and generator G and each H-Node is loaded with 3-keys (i.e., BS public key, its own public/private key pair and $K_{prt}$) and two prime numbers. The memory overhead of our scheme is $5n_H + 4n_L$.

The memory overhead of our scheme is $5n_H + 4n_L$. In the proposed scheme, each L-Node is loaded with one network public key, two prime numbers and generator G and each H-Node is loaded with 3-keys (i.e., BS public key, its own public/private key pair and $K_{prt}$) and two prime numbers. The memory overhead of our scheme is $5n_H + 4n_L$.

Fig. 4 summarizes performance offered by different solutions in terms of the total number of the keys deployed for different size of the sensor networks and the results shows that our proposed scheme requires much less number of keys compared with other approaches.

**C. Network Resilience to Node Compromised Attacks**

Since WSNs are usually deployed in unattended locations, every node of the network should be protected against the possible attacks e.g. node capturing attacks and physical tampering attacks. Here we provide different security measures to the H-Nodes and to the L-Nodes to deal with node capturing attacks and physical tampering attacks. Since after the H-Nodes authentication, each H-Node act as a CH to the L-Nodes and performs the L-Nodes authentication and also act as data sinks, so as they play an important role in the network operation, they are provided with the tamper resistant hardware to protect their secrets. Once the H-Node is captured, all the H-Node secret key materials are deleted and it would no longer be able to authenticate itself to the BS and also would not be able to authenticate any L-Node of the network. The L-Nodes are provided with the tamper resistant hardware just to protect G.

Since the key management techniques described in [8, 12, 13] are based on the probabilistic key pre-distribution, the node compromised attacks can severely affect the network performance as well as the security of networks. As each node is assigned certain number of keys for secret key establishment and if one of them is compromised then adversary can easily use that key pool to establish a secret key with the other nodes of the networks. The probability of the fraction of communications compromised by compromising more than one i.e., $n$ L-Nodes is given by

$$Pr[Compromised] = 1 - (1 - K/p)^n$$

Where $K$ is the size of key pool assigned to each L-Node and $P$ is the key pool size from which $K$ is randomly selected. Once the L-Node in the proposed scheme is compromised, its G is deleted by the tamper resistant hardware which plays an important role in key establishment. Thus L-Node would not be able to establish a secure communication link with the other L-Nodes of the network. Fig. 5 shows the OMNET++ simulation results that how many communication links can be compromised by capturing more than one L-Nodes without involving the CH in our proposed scheme and the schemes proposed in [8], [12] and [13].
improvement in the network resilience against attacks. In future work, we are considering the mobility scenario in which the L-Nodes will move in the network.

REFERENCES


