An Efficient and Scalable Key Management Mechanism for Wireless Sensor Networks

Walid Abdallah*, Noureddine Boudriga*, Daehee Kim**, and Sunshin An**
(*)Communication Networks and Security research Lab, University of Carthage, Tunisia;
(**) Computer Network research Lab, Department of Electronics Engineering, Korea University, Seoul, Korea
ab.walid@gmail.com, noure.boudriga2@gmail.com, dhkim@dsys.korea.ac.kr, sunshin@dsys.korea.ac.kr

***Abstract*—A major issue to secure wireless sensor networks is key distribution. Current key distribution schemes are not fully adapted to the tiny, low-cost, and fragile sensors with limited computation capability, reduced memory size, and battery-based power supply. This paper investigates the design of an efficient key distribution and management scheme for wireless sensor networks. The proposed scheme can ensure the generation and distribution of different encryption keys intended to secure individual and group communications. This is performed based on elliptic curve public key encryption using Diffie-Hellman like key exchange and secret sharing techniques that are applied at different levels of the network topology. This scheme is more efficient and less complex than existing approaches, due to the reduced communication and processing overheads required to accomplish key exchange. Furthermore, few keys with reduced sizes are managed in sensor nodes which optimizes memory usage, and enhances scalability to large size networks.**

***Index Terms*—Wireless sensor networks, Security, Elliptic curve cryptography, Key management.**

I. Introduction

Since their advent, Wireless Sensor Networks (WSNs) have demonstrated high effectiveness in developing a wide range of innovative applications in military and civilian domains, such as battlefield surveillance, border control, structural health monitoring, and patient health care. Conceptually, a WSN is composed of a number of sensor nodes, deployed in a specific zone to detect particular events and to transmit messages to a base station (sink node) in a multi-hop communication fashion using the wireless medium. Sensor nodes are characterized by their low-cost, reduced size, limited processing and communication capabilities, and battery-based power supply.

Ensuring communication security is one of the major issues in WSNs, especially when they are deployed in hostile regions where sensors can be captured and manipulated by adversary or when they are used in critical domains. This is often achieved using symmetric encryption techniques [1] requiring the establishment of shared secret keys, referred to as, the key distribution problem. This problem is much harder to resolve in WSNs than in classical networks, due to the limited resources of sensors. Key management schemes in WSN should deal with many specific issues, namely, decreasing the processing and communication overheads to save energy, using few keys with reduced size to minimize memory occupancy, and optimizing re-keying procedure.

Several research works had been devoted to design appropriate key distribution schemes for WSNs. One of the most used approaches is to pre-load sensor nodes by a set of secret keys randomly selected from a common pool [2], [3], [4]. The main drawback of these schemes is that capturing sensor nodes can reveal keys that are used by non-captured nodes. Besides, a flat wireless sensor network, where all nodes have the same capabilities, is assumed in this approach. In this case, a pairwise key must be setup between each pair of sensors, and thus each node needs to store and manage an important number of keys, which requires a high memory capacity and limits the scalability to large size networks.

Using a hierarchical topology can simplify and improve the scalability and efficiency of key distribution procedure. Indeed, the sensor node doesn't need to establish a pairwise keys with all nodes in the network, but only with those that are in its communication range. Particularly, a sensor will share keys with its cluster head and cluster members; this contributes in reducing the communication overhead and saving energy. LEAP [5] is a key distribution mechanism developed for large scale hierarchical sensor networks, that is able to generate specific keys to secure various types of unicast and broadcast traffics. The main objective of this scheme is to enable the in-network processing, to prevent redundant transmission and optimize resources usage. Authors in [6], investigated the use of secret sharing techniques to design a key management mechanism in hierarchical wireless sensor networks. This scheme has the advantage of ensuring the survivability of the network if a minimum number of nodes are still active and a maximum number of nodes had not been compromised. Nevertheless, it has the limits of generating an extensive communication and processing overheads and requiring the management of an important number of secrets.

Although public key cryptosystems have not been considered at the beginning in WSNs due to their large key sizes and high computation capacity requirement, they are being investigated in some research works [7], [8], [9]. In this context, elliptic curve cryptography [10], [9], [11] is a promising solution which significantly reduces key size, key generation delays, and power consumption. However, these schemes assume static topology, and do not enable in-network processing and re-keying procedure.

This paper proposes an efficient and scalable key distribu-

tion and management mechanism for hierarchical heterogeneous WSNs, that can generate and share keys to provide security services to all traffic types exchanged at different topology layers. Our proposal uses elliptic curve equivalent Diffie-Hellman like key exchange procedure to dynamically establish individual secret keys and group keys between different elements of the WSN. In addition, authentication using digital signature is implemented to overcome the man-in-the-middle attack. This scheme is shown to significantly decrease energy consumption, communication overhead and memory occupancy whilst improving the offered security level and resilience to node capture and replication attacks. Indeed, the use of elliptic curve cryptography provides equivalent security level as classical public key encryption schemes using shorter key size and less computation power. Moreover, it is demonstrated that the key management mechanism can effectively deal with node addition, elimination, and mobility and can be used in large size networks. The main contributions of this work, with regard to existing literature, are as follows:

- The development of an Elliptic Curve Public Key Cryptography (ECPKC) based key management mechanism for WSNs, allowing dynamic establishment of many kinds of secret keys intended for different usages in various levels of the network topology.
- The design of an efficient group key establishment procedure to enable in-network processing and secure intra-cluster and inter-cluster broadcast traffics. This procedure achieves group key sharing in only two rounds, which reduces the processing and communication overheads and saves sensor's energy.
- The proposal of a re-keying procedure based on secret sharing techniques to ensure backward and forward secrecy and improve resilience to node capture attack.

The remaining parts of the paper are as follows: Section II describes the proposed key management scheme. Section III investigates security analysis and performance evaluation. Section IV concludes the paper.

II. Key management mechanism description

In this section, we describe the proposed scalable key management scheme to secure wireless sensor networks. Firstly, we introduce the considered network architecture; then we detail the initial key generation and distribution procedure; finally we investigate node addition, deletion, and mobility.

A. Network topology and assumptions

As depicted by Figure 1, we consider an hierarchical WSN where sensor nodes are organized into a number of clusters using a clustering algorithm. Each cluster is controlled and managed by a Cluster Head (CH) that has higher processing and communication capabilities. In addition, a base station (sink node) collects events from CHs and controls the operation of the WSN. In the sequel, we describe the functionalities and assumptions about these devices.

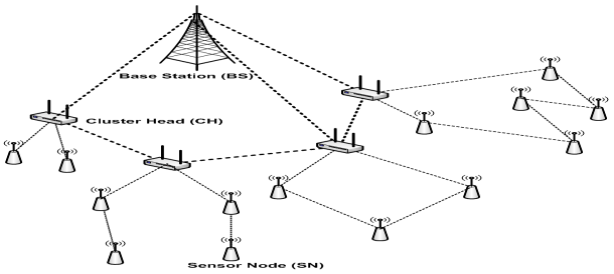

Figure 1. Network architecture

1) Sensor node : Sensor nodes are in the lowest level of the hierarchy. They are low-cost devices with limited computing, storage, and power capabilities. The main mission of a sensor node is to detect particular events and exchange messages with the cluster head and the base station. When the communication range of a sensor node can not reach the cluster head, the packet is relayed by intermediate nodes belonging to the same cluster. We suppose that at any time a sensor can be attached to only one cluster. However, some sensor nodes can be mobile and move from one cluster to another with a low speed.

2) Cluster Head : The Cluster Head (CH) is responsible of collecting data from the members of its cluster and aggregating them in order to optimize transmission channel usage. Also, it manages and control members join and departure procedures. A CH needs to be equipped with an extensively higher amount of resources than sensors, such as higher processing capability, larger storage capacity, longer live batteries, and wider communication range. CHs can communicate directly or relay data to the base station. Due to their limited number, it can be cost-effective to consider that CHs are endowed with a tamper-proof hardware that can resist to physical capture. Moreover, some advanced security technologies such as auto-destruction and memory erasing in case of unauthorized access attempts can be implemented in these devices.

3) Base station : The base station is the most important and secure element of the network implementing the highest capabilities in terms of computing power, storage capacity, and energy, that we assume that are unlimited. In addition, we suppose that the base station is localized in a well known and secured location, trusted by all parties, and has a large communication range that can reach all nodes in the network.

B. Key generation and distribution procedure

The main objective of our work is to design a key management mechanism that can ensure robust authentication, integrity and confidentiality services in the WSN, while taking into consideration the limited resources and processing capability of sensor nodes. This mechanism should allow the generation and distribution of keys in each level of the hierarchy intended to secure individual and group communications. Therefore, we can distinguish the following kinds of keys: individual keys, intra-cluster pairwise keys, cluster key, inter-clusters key, and network key.

In this subsection, we present the Elliptic Curve Public Key Cryptography (ECPKC) based key management mechanism used for dynamic establishment of these keys. First, an overview of elliptic curve cryptography is given. Then, the generation and distribution processes are described.

1) Elliptic curve technique description: Elliptic curve technique [10] offers a valuable opportunity to efficiently apply public key cryptography approach to secure WSNs. Indeed, it provides equivalent security level as in classical public cryptosystems, with significantly reduced key size and processing overhead. Particularly, in Diffie-Hellman, to secure the key exchange process based on the intractability of the Discrete Logarithm Problem (DLP) a minimum key size of 1024 bits is required. However, in elliptic curve equivalent approach a key size of 160 bits is sufficient.

An elliptic curve $E(F_p)$ on the Galois field F_p, where p is an integer, is defined as the set of points that satisfy the general form of the Weierstrass equation. In cryptography, two particular forms of elliptic curves are of interest. The first form, considers p as a prime number. This form is more appropriate for a software implementation of the elliptic curve encryption technique. In the second form, which is adequate for a hardware implementation, $p = 2^k$, where k is prime number. For the both forms, a specific addition operation is defined. Although the proposed scheme can be implemented using the two forms, the second seems to be more suitable to the embedded nature of sensor networks. The elliptic curve in this case is characterized by: $E(F_p) = \{(x, y) \in F_p^2,\ y^2 + xy = x^3 + ax + b\} \cup \{\mathcal{O}\}$, $\mathcal{O}$ is the neutral element.

The more interesting is the equivalent of the DLP in the elliptic curve field. Recall that, given a prime number p, a generator g, and a value h belonging to $\mathbb{Z}_p^*$, the DLP consists in finding x such that $h = g^x mod(p)$. In elliptic curve cryptography, it is believed that, given a field F_p, and two points P and Q belonging to $E(F_p)$, the problem of finding an integer n, such that $Q = nP = P + P + ... + P$ is more difficult than the DLP which allows the use of shorter key sizes. Therefore, mapping between DLP-based classical public key algorithms, such as Diffie-Hellman and ElGamal, and their equivalent in elliptic curve approach can be simply performed by replacing the exponentiation operation with an integer multiplication (i.e. n-time addition) in $E(F_p)$.

2) Individual keys establishment : Individual keys are established between each sensor node and the base station in the initial phase of network deployment. We assume that the hierarchical network topology has been created and that sensor nodes can communicate with the base station to establish secret keys. This is performed in our scheme using elliptic curve based Diffie-Hellman key exchange procedure according to the following steps:

- Pre-deployment : Before deployment, the base station randomly selects an integer number p, and the elliptic curve $E(F_p)$ according to the second form as discussed above. A point $G \in F_p$ and a private value $x_B \in \mathbb{Z}_p$, where $2 \leq x_B \leq p - 1$, are also selected. Furthermore, the base station calculates the elliptic curve public point $Y_B = x_B G$. The parameters p, $E(F_p)$, G , Y_B, and an initial key K_0 will be pre-loaded in each deployed sensor node. K_0 is used to verify the genuineness of the sensor node. It is only valid during the short period of the initial deployment phase and will be deleted after key establishment. We denote by N the total number of deployed nodes, where each node is uniquely identified by an id value.
- Private/public keys generation and individual key calculation: Every node i, $1 \leq i \leq N$, will generate a private value $x_i \in \mathbb{Z}_p$. This is performed by applying a hash function as follows: $x_i = Hash(id_i||K_0||N_i)mod(p)$, N_i is a randomly generated nonce. Then, the sensor node calculates the elliptic curve public point $Y_i = x_i G$ and the individual secret key $K_i = x_i Y_B = x_i x_B G$. The sensor node sends the public point Y_i to the base station to be digitally signed. The message is authenticated by a Hash Message Authentication Code (HMAC) using K_0 to ensure that it comes from a genuine deployed sensor. This generation procedure ensures that all private values are different from each other and enable data origin authentication.
- Public key validation and individual key establishment in the base station: After verifying the MAC, the base station signs the public key using its private value x_B, saves it in its data-base, and establish the shared key $K_i = x_B Y_i = x_i x_B G$. Finally, the base station sends and acknowledgment to the sensor node that deletes immediately the initial key K_0 from its memory. Elliptic curve approach allows the sharing of a point with two coordinates. In our case we choose the key K_i as the abscissa the point in $E(F_p)$. This rule will be applied to all subsequent keys.
  It is worthy to note that the individual key K_i is a master key that is not directly applied to secure packets exchanged between the base station and the sensor node. Two session keys denoted as, K_{ie}, and, K_{ia} are derived from the key, K_i and a counter value to prevent replay attack. These keys are used respectively for encrypting and generating the MAC of each message exchanged between the sensor node to the base station.

3) Intra-cluster pairwise keys and cluster key establishment : Intra-cluster pairwise keys must be established to secure communication between each sensor node, its cluster head, and each one of its neighbors. In addition, a cluster key shared between all nodes of the cluster is established to enable in-network processing and optimize resources usage.

Pairwise keys are established in a similar way as individual keys described above. The only difference is that signed public values must be retrieved from the base station and each party verifies the validity of the signature before key establishment. We denote by $K_{ij} = x_j Y_i = x_i Y_j = x_i x_j G$ the pairwise key established between neighbor nodes i and j, and $K_i^c = x_i Y_c = x_c Y_i = x_i x_c G$ the pairwise key established between

the node i and its CH c.

For the cluster key a group communication secret key sharing procedure was proposed. This scheme is more efficient than the existing techniques [12] because it allows key establishment in only two rounds. To this purpose, for each node j of the cluster, the CH calculates and sends a public value $Y_{cj} = x_c \sum_{n=1, n\neq j}^{m_c} Y_n$, where m_c denotes the number of sensor nodes in the cluster c. The cluster key, K^c can be determined in each node by simply adding this value to the already established intra-cluster pairwise key as, $K^c = K_j^c + Y_{cj} = x_c \sum_{n=1}^{m_c} Y_n$.

4) Inter-cluster key and network key establishment: Using the same procedure as for the cluster key, CHs and the base station can share an inter-cluster key $K^B = x_B \sum_{c=1}^{M} Y_c$ to secure message broadcast in the second level of the hierarchy. M is referred to as the number of clusters. In addition, a network key K_N can be securely distributed to all sensor nodes using two encryption stages. In the first stage the base station randomly generates K_N, encrypts it with the inter-cluster key, and transmits it to all CHs. In the second stage, each CH decrypts the network key and encrypts it with the cluster key before broadcasting it to all cluster members.

C. Keys management procedures

In this subsection we describe procedure of modifying the different types of keys due to new nodes deployment or elimination. Also, we detail the re-keying process that will be executed to initiate the establishment of new keys when the validity of the current keys expires.

1) New nodes deployment: When a new node is deployed in the WSN, it must first create its individual key shared with the base station using the same procedure as described in the previous section. The main difference is that the initial key will be different from the one used in the initial deployment phase. Indeed, suppose that a new node will be added at the instant t after the deployment. The base station will generate and configure the node with an initial key K_t. This procedure will prevent an adversary, that have access to previous initial keys, to add its own replicated nodes. Once the individual key is generated and the public value is validated, the sensor follows the previously described steps to establish the other keys.

2) Nodes elimination and revocation: When a compromised node is detected by the CH, it informs the base station to invalidate its public key and adds it to the revocation list. The CH will isolate the compromised node and establish a new cluster key by eliminating the public value of the compromised node. Also, the base station will generate and distribute a new network key using the new cluster key.

3) Mobility Management: The use of public key cryptography approach in the proposed key distribution mechanism enables an efficient key update even in case of mobile sensor nodes. We assume that some sensor nodes can move form one cluster to another with a moderate frequency. The sensor node should establish a pairwise key with its new CH and participate in the generation of a new common cluster key using the same procedures as described earlier. However, the new CH should verify the validity of the public value of the node that wants to join the cluster. Also, the old cluster should be informed that the node has left the cluster to initiate cluster key update.

4) Re-keying procedure: A global re-keying procedure is triggered when the number of compromised nodes reaches a given threshold or the validity period of the generated private keys expires. New private and public keys should be created to renew different shared keys. To this end, each sensor node should reconstruct the key, K_r that will play the same role as the initial key used in the deployment phase. We have investigated the use of threshold secret sharing techniques to manage the distribution and the reconstruction of this key. The basic idea is that every sensor node will possess a partial secret that can be used to reconstitute the key K_r. However, this cannot be achieved unless a minimum number of nodes, denoted by t, collaborate together and assemble their secrets. This approach has the advantage of maintaining the security of the key if the number of compromised nodes is less than $t-1$. Also, the re-keying procedure can be initiated if at least t trusted nodes are still operational in the network. Our proposal uses the Shamir's method[13] based on the Lagrange interpolation. This approach consists in randomly selecting a polynomial, $f(x) = K_r + a_1 x + a_2 x^2 + ... + a_{t-1} x^{t-1} mod(Q)$ by the base station, where Q is a prime number. We can notice that, $K_r = f(0)$ and all coefficients of $f(x)$ must belong to $\mathbb{Z}_Q$. For $i = 1, 2, ..., N$, the secret S_i of each sensor node i is calculated as $S_i = f(id_i)$, where id_i is a unique identifier of the node i. Each partial secret must be securely transmitted to the corresponding sensor node. To this end, the base station will encrypt every secret S_i by the individual shared key K_i. According to the Lagrange interpolation, $f(x)$, can be reconstructed by giving t points $(S_1, S_2, ..., S_t)$. Particularly, the key K_r can be reconstructed by the equality $K_r = f(0) = \sum_{i=1}^{t} S_i \left(\prod_{i\neq j} \frac{id_j}{id_j - id_i} \right) mod(Q)$.

III. Security analysis and performance evaluation

A. Security analysis

The evaluation of security schemes intended for WSNs is significantly different from those used in conventional networks. Indeed, evaluation criteria should consider the characteristics of WSNs deployment and their resource constraints. In this subsection, we evaluate the security offered by our keys distribution mechanism with regard to four proprieties that reflect the specificity of WSNs: (1) the possibility of providing backward and forward security for encrypted data, (2) the resilience to node capture, (3) resistance against node replication, (4) scalability to large size network.

1) Backward and forward security: The forward security propriety is the possibility that an attacker can predict a future key if he captures the currently used key. On the other hand, backward security is the possibility for an attacker to obtain information about previously used keys, by capturing the currently used key. This is not possible in our scheme because it is based on public key cryptography where previous and future private/public values used to calculate shared keys are generated independently. In addition, group communication keys are modified each time a change in the network topology occurs in the sensor level and in the cluster level.

2) Node capture: In many applications, sensor nodes are randomly deployed by aerial dropping in a large areas. Consequently, sensor nodes can be easily captured by an adversary, who can access to their memory content. Security schemes should maximize the network resilience by minimizing the amount of information revealed to attacker on non-captured nodes. A sensor node can be accessed either using soft capture or physical capture. In the soft capture, the attacker tries to establish a connection to access to the management console of the sensor node. Many techniques can be used to implement authentication in administrative mode, such as passwords, RFID technology, and challenge-response approaches. Concerning the physical capture, in our case an attacker can capture either a sensor node or a CH. When an attacker capture a sensor node, he can access to its individual key, pairwise keys, the cluster key, and the network key. The last two keys can affect security within the cluster and the network. They should be modified by eliminating the public value of the captured node and distributing new keys. In addition, sensor node stores a single part of the shared secret used to reconstruct the re-keying key. To prevent the discovery of this key, the number of captured nodes should not exceed t. On the other hand, getting access to a CH is more dangerous than a sensor node. In this case, the attacker can access to all pairwise keys, the cluster key, the inter-cluster key, and the network key. This can be prevented by equipping the CHs with a tamper-proof physical hardware. In addition, techniques, such as physical auto-destruction and soft memory erasing can be envisioned to avoid illegal access to the content of the CH. Besides, all the group communication keys must be modified when a CH is compromised. Also, the cluster members should be able to connect to other CHs.

3) Node replication: The node replication attack consists in the possibility that an adversary party can introduce malicious nodes after gathering information from captured nodes. In this case, the replicated nodes will try to establish connection with other nodes, CH, or even the base station. These nodes should be detected and isolated from the network. Our scheme can guarantee resistance against node replication attack, because any new node should generate a private value based on its identity and a valid secret initial key before establishing any connection in the network. However, initial keys are modified according to the instant of adding the new node and are eliminated from the memory after the generation process. In addition, all the public values are validated by the base station and any key establishment process uses only public values transmitted by the base station. The base station has track of all identities of deployed and revoked nodes. Therefore, before establishing any secure connection within a cluster, the identity of the node is checked and unauthorized nodes can be detected and eliminated from the network.

Parameter	Value
Number of sensors	100-1000
Packet size	36 Byte
Acknowledgment size	12 Byte
Private, Public keys length	160 bits
Symmetric key length	128 bits
Transmitting energy	$59.2\mu J/Byte$
Receiving energy	28.6 $\mu J/Byte$
ECC private, public key setup energy	22 mJ
ECC Signature verification	45 mJ
MAC computation energy (SHA1)	$5.9\mu J/Byte$
Encryption/Decryption energy (AES)	1.62/2.49 $\mu J/Byte$

Table I
SIMULATION PARAMETERS

4) Scalability: The scalability is the ability of the scheme to maintain an acceptable security level regardless of the network size. The designed key distribution system is fully scalable because it is based on public key encryption that provides an effective security independently of the number of nodes deployed in the network. In addition, the hierarchical topology ensures the scalability of the communication process and optimizes resources consumption.

B. Performance Evaluation

In this subsection, we assess the performances of the proposed scheme with regard to the required key storage capacity, communication overhead, and energy consumption. We compare the results to the LEAP scheme [5] which implements the in-network processing concept using symmetric pairwise key pre-distribution paradigm. To this end, we developed a simulation model using Matlab. In each simulation, the proposed ECPKC and LEAP are executed on a set of randomly generated topologies composed of a number of sensors. We consider clustered topology and we compute performance parameters by varying the number of sensor nodes. The number of cluster in each topology is taken as : $M = \lceil 0.2N \rceil$ where N is the number of sensor nodes. In the implementation of the simulation model we used the values given by Table I adopted from [7]. For each number of sensors we generate 5 topologies, and we compute the memory occupancy, the communication overhead, and the energy consumption needed to establish keys in every network. The final results are obtained by taking the average on all values measured for all generated topologies.

Figure 2 depicts the performances of our ECPCKC scheme in terms of required storage capacity, communication overhead, and energy consumption.

We can notice that our scheme has remarkably reduced memory occupancy when compared to the LEAP protocol. Moreover, the needed storage capacity of our scheme varies

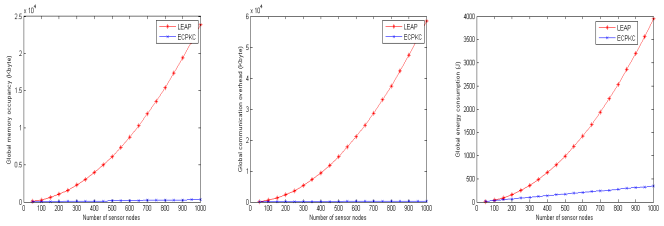

Figure 2. Performances of the proposed scheme

almost linearly with the number of sensor nodes. However, for LEAP, it increases rapidly with the number of sensor nodes. This is due to the fact that in our scheme, each sensor node manages a limited number of public keys and symmetric keys that are shared with the base station and the CH. Also, each node shares several public and symmetric keys with its neighbors that can not directly reach the CH. On the other hand, in LEAP, the number of keys that must be stored in each node depends on the number of its neighbors, since a one pairwise key and a cluster key should be shared with each neighbor node. Consequently, the number of needed keys increases with the density of the network.

The same observation can be formulated for the communication overhead. In our ECPKC approach, the sensor node will initiate the key exchange procedure with the base station, the CH, and a limited number of its neighbor nodes. This decreases the number of messages needed to establish shared keys. Also, the proposed group key establishment procedure requires only the exchange of one packet and an acknowledgment between a sensor and its CH. Another important parameter for any key distribution scheme is energy consumption. It can be observed that the elliptic curve based scheme consumes less energy than LEAP. Moreover, the energy consumption varies linearly with the number of sensor nodes which can ensure the scalability of our scheme to large scale networks.

IV. Conclusion

Key distribution and management in WSNs is much more difficult than it is in classical networks owing to the resource constraints, important number of nodes, and the lack of infrastructure support. Consequently, tailored key distribution schemes need to be developed taking into consideration the limited computing capability, the little storage capacity and the finite energy of sensor nodes. In this paper we addressed key management problem in WSNs. We proposed an elliptic curve public key cryptography based key management scheme. Our scheme is able to ensure secure sharing of many types of keys in each level of the network topology. Particularly, it uses elliptic curve Diffie-Hellman like key exchange procedure to establish pairwise keys between the sensor node, the base station, and its cluster head. Also, a group key establishment protocol was proposed to create a cluster key used to secure communication within each cluster and an inter-cluster key used to secure message exchange between the cluster heads and the base station. These keys enable in-network processing, which improves message transmission efficiency and resources usage in the WSN. Furthermore, the proposed approach enables re-keying procedure based on the concept of threshold secret sharing mechanism. Security analysis and performance evaluation using simulation works showed that the ECPKC mechanism ensures an enhanced security level while reducing the required storage capacity, communication overhead, and energy consumption which enables an efficient and scalable implementation of our scheme in large scale WSNs. Finally, developing a strong authentication method for broadcast traffic based of the the proposed key distribution scheme and ensuring adaptive security in WSNs can be envisioned in future works.

Acknowledgments

This work was partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government and the Tunisian government (No. NRF2012K1A3A1-A09026959).

References

[1] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: A link layer security architecture for wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems, SenSys '04*, 2004, pp. 162–175.

[2] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *proceedings of the 9th ACM conference on Computer and communications security, CCS'02*, Washington, DC, USA, November 2002, pp. 41–47.

[3] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *proceedings of the IEEE Symposium on Security and Privacy, SP03*, Berkeley, California, May 2003, pp. 197–213.

[4] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security TISSEC*, vol. 8, no. 2, pp. 228–258, May 2005.

[5] S. Zhu, S. Setia, and S. Jajodiadd, "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 4, pp. 500–528, November 2006.

[6] Y. Zhang, C. Wu, J. Cao, and X. Li, "A secret sharing-based key management in hierarchical wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 2013, pp. 1–7, 2013.

[7] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *proc. of the third IEEE International Conference on Pervasive Computing and Communications PerCom 2005*, March 2005, pp. 324–328.

[8] M. I. Salam, P. Kumar, and H. Lee, "An efficient key pre-distribution scheme for wireless sensor network using public key cryptography," in *proceedings of the sixth International Conference on Networked Computing and Advanced Information Management (NCM 2010)*, Seoul, South Korea, August 2010, pp. 402–407.

[9] D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," in *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, October 2004, pp. 71–80.

[10] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, pp. 203–209, 1987.

[11] X. Du, Y. Xiao, S. Ci, M. Guizani, and H.-H. Chen, "A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *IEEE Transaction on wireless communications*, vol. 8, no. 3, pp. 1223–1229, March 2009.

[12] G. Ateniese, M. Steiner, and G. Tsudik, "New multiparty authentication services and key agreement protocols," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 628–639, April 2000.

[13] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, November 1979.