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 in many applications of Wireless Sensor Networks (WSNs) is ensuring security. Particularly, in military applications, sensors are usually deployed in hostile areas where they can be easily captured and operated by an adversary. Most of security attacks in WSNs are due to the lack of security guaranties in terms of authentication, integrity, and confidentiality. These services are often provided using cryptographic primitives where sensor nodes need to agree on a set of secret keys. Current key distribution schemes are not fully adapted to the tiny, low-cost, and fragile nature of sensors that are equipped 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 that is applied at different levels of the network topology. In addition, a re-keying procedure is performed using secret sharing techniques. This scheme is more efficient and less complex than existing approaches, due to the reduced number of messages and the less processing overhead required to accomplish key exchange. Furthermore, few number of encryption 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, Key distribution and management, Elliptic curve cryptography, threshold secret sharing**

I. Introduction

Since their advent, Wireless Sensor Networks (WSNs), have been the subject of an increasing interest from the academic and industrial communities, due to their wide and varied number of applications in military and civilian domains. These networks demonstrated high effectiveness in the development of many innovative applications such as battlefield surveillance, border control, structural health monitoring of aircraft, environment parameters measurement, and patient health care[1], [2], [3].

Conceptually, a WSN is composed of a number of sensor nodes, deployed in a specific zone to detect particular events and transmit messages to a base station (sink node) in a multi-hop communication fashion using the wireless medium. Sensor nodes are characterized by their reduced size, limited processing capability, and battery-based power supply. These characteristics must be taken into consideration in developing appropriate communication protocols. Particularly, ensuring communication security is one of the major issues in WSNs, especially when they are distributed in hostile regions where sensors can be captured and easily manipulated by an adversary. Furthermore, with advances achieved in wireless technology, WSNs are being used in critical domains, such as controlling aircraft and avionic systems, surveying health states, and monitoring toxic gas emission where security attacks can have very dangerous consequences on human safety. Therefore, providing security services for data communication in WSNs becomes a main requirement to avoid malicious activities and even terrorist attacks.

Typically, to ensure communication security, at least four services must be provided, namely, confidentiality, authentication, integrity, and availability. Most of these services are based on the implementation of cryptographic techniques which require the establishment of a set of shared encryption keys. A wide range of cryptographic algorithms and schemes had been developed to enable dynamic key distribution and management in classical networking infrastructure, such as asymmetric encryption techniques, digital signature schemes, and Public Key Infrastructure (PKI). However, these techniques cannot be directly used in WSNs. Indeed, sensor nodes cannot sustain the high processing overhead and complexity of these techniques due to their limited computational capacity and reduced memory size. In addition, most of the existing key management protocols must perform extensive message exchanges to establish keys, which increases power consumption, depletes the sensor node limited energy, and shortens the network operational lifetime. Besides, in a WSN, nodes can leave the network because they run out of energy or are captured and eliminated by an adversarial party. Therefore, new sensor nodes must be added after the initial deployment phase to replace the removed nodes

Manuscript received July 1, 2014. This work was supported in part by by the National Research Foundation of Korea (NRF) grant funded by the Korean government and the Tunisian government (No. NRF2012K1A3A1-A09026959).

Walid Abdallah is an Assistant Professor and member of the Communication Networks and Security research Lab, Tunisia (corresponding author; Phone: +21698229840; e-mail: ab.walid@gmail.com). Pr. Noureddine Boudirga is a Professor of Telecommunications in the School of Communication Engineering, SupCom, Carthage University and the Director of the Communication Networks and Security Research Laboratory (CNAS), Tunisia (email:noure.boudriga2@gmail.com). Deahe Kim is a Phd student in the Computer Network research Lab, Department of Electronics Engineering, Korea University, Seoul, Korea (email: dhkim@dsys.korea.ac.kr). Pr. Sunshin An is a Professor of Electronic and Computer Engineering in Korea University, Seoul, Korea (email: sunshin@dsys.korea.ac.kr)

or to enhance the connectivity of the network. Consequently, the key management scheme should deal with this issue to avoid the failure of the network and optimize the re-keying procedure when nodes are deleted or added.

Several research works[4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] had been devoted to design appropriate key distribution schemes for WSN. The proposed schemes were based on pre-distribution of symmetric or asymmetric keys before the network deployment or on secret sharing using threshold cryptographic techniques. Although some of these schemes can offer extensive security to data transfer on wireless sensor network, they are complex to apply in real environment and do not scale to large networks. In addition, key establishment protocols require a high number of message exchanges which exhausts the limited available energy of sensor nodes and shorten the network operational live time. Moreover, all existing key distribution and management schemes assume static topology and do not consider the case where mobile sensor nodes can be employed.

This paper proposes the design of a key management scheme for wireless sensor networks adapted to hierarchical topologies. It is an enhancement of the work presented in [15]. Our key distribution scheme can perform efficient and scalable generation and sharing of cryptographic keys to provide authentication, integrity, and confidentiality services to all types of data traffic exchanged at the different layers of the network topology. Our proposal uses elliptic curve based Diffie-Hellman like exchange procedure to establish individual secret keys between different elements of the WSN, such as sensor node and base station, the sensor and its cluster head, and each cluster head and the base station. These exchanges are exploited to generate secure group keys to ensure intra-cluster and inter-cluster communications privacy. Authentication of the exchanged values is implemented to overcome vulnerability to the man-in-the-middle attack. Our secret key establishment approach is less complex and requires reduced message exchanges than existing schemes whilst it improves offered security level. Moreover, the use of elliptic curve techniques allows shorter key sizes and decreases the processing overhead of the cryptographic operations while ensuring the same security level as current public key schemes. 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 and taking into consideration node mobility.
- 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 exposes related works to the key distribution problem in wireless sensor networks. Section III describes the proposed key management scheme. Section IV, analyzes the security level provided by the proposed scheme. Section V performs a performance analysis of the proposed scheme, Section VI, presents simulation work conducted to evaluate the effectiveness of the key management approach and demonstrate its scalability. Section VII, concludes the achieved work in this paper and gives some perspectives.

II. Related Work

Key distribution problem in wireless sensor networks, had been the subject of many research works during the last decade[16]. Key distribution and management procedure play a crucial role in guaranteeing the security of any data exchange using cryptographic primitives. Key management encompasses the processes of generating, distributing, storing, and updating encryption keys. The main target is to prevent attacker from exploiting weaknesses in the key management procedure to derive encryption keys and break the security of the wireless sensor network. Due to the limited resources of sensor nodes, the large number of deployed nodes, and missing of infrastructure, key distribution and management is a major issue in wireless sensor networks.

Secret key cryptographic techniques are the most suitable to secure communication in wireless sensor networks. These techniques can be executed in reduced computational capability processors with an acceptable delays. In addition, they manipulate short key sizes requiring few memory occupancy. This has the advantage to reduce the energy consumption, increase efficiency, and ensure reliability of the network. TinySec [17] is an effective implementation adapted to wireless sensor nodes to ensure link layer security in terms of confidentiality, integrity and authentication. Authors studied, specific operational modes for a set of secret key encryption algorithms and message authentication codes, that can satisfy resource constraints of sensor nodes. For instance, in TinySec, RC5 and Skipjack are considered as the most suitable encryption algorithms to offer data confidentiality. One of the main problems in secret key cryptosystems is key distribution consisting in the procedure to securely share secret keys between sensor nodes. Many research works[5], [6], [8], [18], [7], [9], [10], [19], [20], [4], [21], [14] have been devoted for studying key distribution and management issue in wireless sensor networks proposing multiple schemes with various approaches.

Most of the proposed key distribution schemes are based on the pre-distribution of a set of secret keys in the sensor nodes before network deployment [5], [6], [8], [7], [12]. The work presented by Eschenauer et al [5] is the first in this context. The authors proposed a key distribution scheme based on pre-loading a set of secret keys in each sensor node that are randomly selected from a common pool of keys. At the initial phase of network deployment, each sensor node exchanges information about the pre-configured keys with its neighbors to find those that share with it the same keys. When two neighbors find that they have a common key, they establish a secure communication link between themselves. Furthermore, the established secure links can be used to negotiate the sharing of pairwise keys between nodes that their key sets did not overlap. It was proved, using random graph theory that, if the probability that two selected sets of keys share at least one key is greater than a given value, then secure connectivity of the network can be achieved with high probability. Although, this scheme may be very efficient in establishing shared keys in wireless sensor networks, its main drawback is that when the number of jeopardized nodes increases, the security level significantly degrades. When sensor nodes are captured, all shared keys can be discovered and encrypted data will be disclosed to an adversary. Moreover, given that the same key can be used to secure many links, the attacker may even be able to decrypt data being currently transmitted between non compromised nodes. Besides, to offer full communication security, each sensor node needs to store and manage an important number of keys, which requires a high memory capacity and limits the scalability of this solution to large size networks.

Chan et al [6] introduced the concept of q-composite. In this approach, a secure link between two neighbor nodes is established if they have at least q common keys, where $q \geq 2$. Authors show that increasing the number of shared keys, q, boosts the resilience to node capture, in the sens that the attacker will need to compromise a higher number of nodes than in the original scheme described in [5] to decrypt the same amount of data. Despite resilience enhancement against node capture, this scheme has not resolved the main limits, which are, the complex procedure and the communication overhead needed to establish a full one-hop secure connectivity between neighbor nodes, and the required memory to store and manage shared keys. To enhance random key pre-distribution approach, Du et al [7] presented a technique to establish pairwise keys between sensor nodes based on the random selection of rows and columns in a key matrix. In this scheme, multiple key generation spaces are used to enhance resilience to node capture. Nevertheless, this approach cannot guarantee that two nodes can share a direct secure link, and the key path establishment procedure of the original scheme [5] is still needed. All these schemes are developed for wireless sensor networks configured in a flat topology where all nodes have the same capabilities and thus a key pairwise must be setup between each pair of sensors. This can reduce scalability to larger size networks due to higher power consumption, extensive processing requirement, and communication overhead.

Using a hierarchical topology can simplify and improve the scalability and efficiency of the key distribution procedure. In this case, 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 (CH) and cluster members; this contributes in reducing the communication overhead and saving energy. Several works have studied the design of key distribution mechanism for hierarchical sensor networks [18], [22], [12], [14], [10].

Localized Encryption and authentication protocol (LEAP) [18], [22] is an energy efficient key distribution mechanism developed for large scale hierarchical sensor networks, that is able to generate specific keys for securing various types of uni-cast and broadcast traffics. Four kinds of encryption keys are defined: individual key shared between the base station and each sensor node, pairwise key that is a unique key established between the node and its cluster head, cluster key is a common key used to secure data intended for all members of the cluster, and group key used to secure data broadcast to all nodes of the network. All these keys are derived from a unique master key that is pre-loaded in each node before deployment. This master key is erased from the memory at the end of the initial key distribution process, to avoid that an adversary party capturing a single node, can compromise all data transmitted in the network.

A similar approach was described in [12], where a security architecture was proposed for wireless sensor networks based on a master secret key that is embedded in the source code of the operational system in every sensor node. The authors claim that this can prevent the disclose of the encryption keys derived from the master key and stored in non-volatile memory even if the node is captured. Although, we can agree that configuring the master key in the source code of the application program can harden the task of an attacker to retrieve it, but this is not impossible. Also, it will be very hard to upgrade the security parameters of the encryption scheme such as the key size, encryption algorithms, and the master key itself, because this will require the upload of another operation system in all deployed nodes.

Secret sharing techniques have been investigated in [14] to design a key management mechanism in hierarchical wireless sensor networks. This scheme allows the distribution of keys in different levels of the topology. Indeed, individual secrets are distributed to all nodes of the network. Group keys can be constructed by resembling a minimum number of individual secrets and applying a polynomial interpolation. This 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, to ensure security of the transmitted data, the shared secrets must be modified for every session to prevent key compromising due to the capture of a single node participating in the reconstruction process. This generates an extensive communication and processing overheads and increases power consumption. In addition, sensor nodes are required to store an important number of session secrets overloading the limited memory capacity of sensor nodes.

The previously described key distribution schemes for WSNs are static in the sense that they are based on the preloading of secret keys that remain valid during all the network life-time. More specifically, these schemes do not define rekeying procedures to update encryption keys. This fact, can constitute a serious security limit in these mechanisms, as using an encryption key for a long period of time can increase the probability of being compromised. Some dynamic key management schemes that enable key updating had been proposed for wireless sensor networks [21], [23], [24]. These schemes are mainly based on secret key encryption techniques which makes them vulnerable to node capture attacks. In these dynamic key management schemes, capturing a specific number of sensor nodes can lead to revealing keys used by non compromised nodes. Public key encryption techniques can be investigated to resolve these problems.

Although, public key cryptosystems have not been considered at the beginning in WSNs to secure key distribution owing to their key sizes and high computation capacity requirement, they are being investigated in some research works[25], [19], [4]. This is motivated by the advances achieved in physical node architecture technology and the enhancement of their computation capacity. In addition, a promising solution is the use of elliptic curve cryptography which significantly reduces key size, achieves key generation in a limited delay, and consumes a few amount of power [13].

In this paper, we investigate the design of a key distribution and management scheme for wireless sensor networks with hierarchical topology. Our approach consists in combining different techniques, each one will be used in a specific context in order to ensure the highest security level while guaranteeing efficiency and scalability of the key distribution process. Our proposal is based on using elliptic curve public key cryptography, in particular on the Diffie-Hellman like key exchange procedure, to establish secret keys in the different levels of the hierarchy. A unique private key is generated by each sensor node at the initial phase of the network deployment. The generation process is based on the identity of the node and a key that is pre-loaded in the sensor node and deleted after the generation of the private key. The validation of the corresponding calculated public key is achieved by the base station. The public and private keys are then used to establish individual and group secret keys to secure different kinds of uni-cast and broadcast traffics. In addition, our scheme enables secure re-keying procedure by using secret sharing techniques to regenerate the initial key and reconfigure the overall security parameters.

III. Key distribution and management scheme description

In this section, we describe the proposed scalable key distribution and 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 issues related to node addition, deletion, and mobility.

A. Network topology and assumptions

Wireless sensor networks can be configured into two main topologies: flat homogenous and heterogeneous hierarchical. In flat topology, all sensors have the same capabilities in terms of sensing, computing and communication. Whereas, in hierarchical topology, the network is composed of many kinds of nodes with divers capabilities and perform different functions. Flat wireless sensor networks are more simple to deploy, however hierarchical architectures are more efficient and scalable.

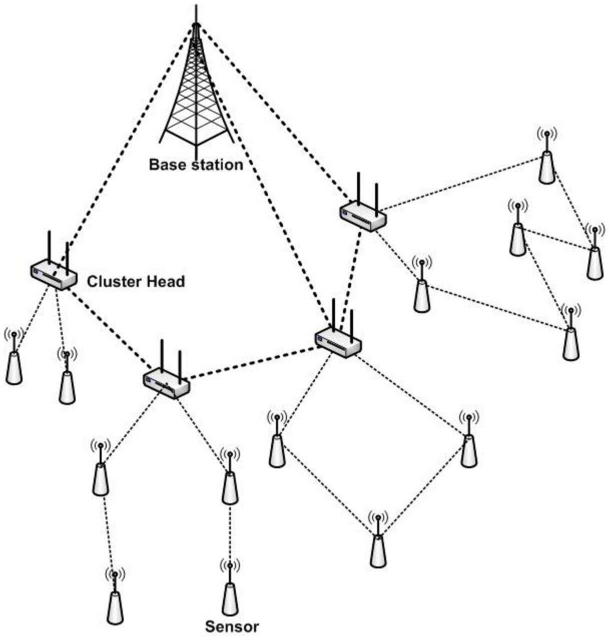

Fig. 1. Network architecture

In this work, we consider a hierarchical wireless sensor network composed of a large number of sensor nodes that are organized into a number of clusters. Each cluster is controlled and managed by a cluster head which is a device with higher processing and communication capabilities than sensor nodes. After deployment, the cluster heads need to execute an appropriate clustering algorithm [26] to divide the network into an optimized number of clusters.

The considered network topology is depicted by Figure 1. As we can see, the network architecture encompasses three types of network devices, sensor nodes, cluster heads, and the base station (or Sink node). In the sequel, we describe the functionality of each one of these devices.

1) Sensor nodes : Sensor nodes are in the lowest level of the hierarchy. They are low-cost devices with a very limited computing, storage, and communication capabilities. Also, they are power supplied using a finite life battery. The main mission of a sensor node is to detect particular events and to exchange messages with its cluster head and the base station. Also, a sensor node can relay messages transmitted by sensor nodes which their communication ranges do not reach the cluster head. In some situations, the base station can exchange messages with the sensor nodes. This can happen for example when the configuration of sensor nodes needs to be updated or when a particular event happens. In addition, we suppose that at any time a sensor can be attached to only one cluster. However, sensor nodes can be mobile and move from one cluster to another with a very low speed.

2) Cluster heads : The cluster head 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 controls all procedures of member join and departure. A cluster head needs to be equipped with an extensively higher amount of resources than the sensor node.

In our architecture we suppose that cluster heads encompasses a higher processing devices with large storage capacity and more powered and long live batteries. Moreover, we consider that they are able to achieve more complex operations and have a wider communication range than sensors. Cluster heads can communicate with each other directly and relay data to the base station. Due to their limited number, it can be cost-effective to assume that cluster heads are endowed with a tamper-proof hardware that ensures resistance to node capture attack. Moreover, some advanced security functionality such as auto-destruction and memory eraser in case of unauthorized access attempts can be implemented in these devices.

3) Base station : The base station is the network element that implements the most higher capabilities. We assume that it has unlimited resources such as, computing power, storage capacity and energy. Moreover, the base station has a very large communication range that can reach all nodes in the network. Depending on the application, the base station can be localized either in the center or the corner of the network. In any case, it is supposed that the base station is installed in a well known and secure location. Also, it is considered as the most secure element of the topology and is trusted by all entities of the wireless sensor network.

B. Key generation and distribution procedure

The main objective of our work is to design a key management procedure that ensures robust authentication, integrity and confidentiality services in the sensor network and takes into consideration the limited resources and reduced processing capability of the sensor nodes. The key management mechanism should allow secure generation and distribution of keys in every level of the hierarchy. In addition, it must enable the establishment of different group communication keys that can be used to perform in-network processing. The in-network processing capability consists in a the ability of sensor nodes to decrypt packets transmitted by neighbor nodes in order to avoid event detection redundancy and allow data aggregation. These operations are very useful in many applications and permit energy saving and channel usage optimization. Consequently, we can distinguish the following kinds of keys:

- Individual keys: used to secure communication between a sensor node and the base station.
- Intra-cluster pairwise keys: shared between a sensor node and its cluster head and neighbor sensor nodes belonging to the same cluster
- Cluster key established between all sensor nodes of the same cluster to secure group communications.
- Inter-clusters key: used to secure communication between all clusters heads and the base station
- Network key: shared between all nodes of the network and used to secure message broadcast.

In this work, we investigate the use of elliptic curve public key cryptography to enable efficient and secure key exchange in wireless sensor networks. In the upcoming subsections, we present our Elliptic Curve Public Key Cryptography (ECPKC) based key management mechanism proposed to carry out dynamic establishment of the aforementioned kinds of keys. First, an overview of elliptic curve cryptography is given in this paper. Then, the generation and distribution processes are described.

1) Elliptic curve fields selection: Elliptic curve techniques [27] offer a valuable opportunity to efficiently apply public key cryptography approach to secure wireless sensor networks. These techniques are able to provide equivalent security level as classical public key cryptosystems, namely the Diffie-Hellman key exchange procedure, with significantly reduced key size. For example, in Diffie-Hellman a minimum key size of 1024 bits is required to ensure the security of the key exchange procedure. Indeed, the discrete logarithm problem, on which is based the security of this key establishment protocol, becomes intractable for a key size higher than this value. However, with elliptic curve equivalent approach a key size of 160 bits is sufficient to ensure security. In the following we explain how this was achieved.

Given a Galois field F_p, where p is an integer number, an elliptic curve, $E(F_p)$ is defined by the set of points that satisfy the Weierstrass form defined by the following equality:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

where, $a_i \in F_p$.

In cryptography, two forms of the Galois finite fields are of interest. The first form considers a field F_p, with p a prime number, and an elliptic curve satisfies the equation:

$$E(F_p) = \{(x, y) \in F_p^2,\, y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\} \tag{2}$$

where a, b are satisfying $4a^3 + 27b^2 \neq 0$ and $\mathcal{O}$is the neutral element of the curve. This form is very useful for a software implementation of the elliptic curve encryption paradigm.

The second form considers a field F_p, with $p = 2^k$, and k is prime number. The elliptic curve in this case is characterized by the equality:

$$E(F_p) = \{(x, y) \in F_p^2,\, y^2 + xy = x^3 + ax + b\} \cup \{\mathcal{O}\} \tag{3}$$

This form is more adapted for hardware implementation of elliptic curve based encryption algorithms.

These two forms are not vulnerable to the sub-exponential attack and can guarantee the security of the key exchange procedure. For both forms a specific addition operation is defined. The more interesting is the equivalent form of the discrete logarithm problem in the elliptic curve field. Recall that, the discrete logarithm problem consists in, given a prime number p, and a generator g and a value h belonging to $\mathbb{Z}_p^*$, it is difficult to find, x where $h = g^x mod(p)$. In elliptic curve cryptography, it is believed that, given a field $F_p$satisfying one of the aforementioned forms, 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 discrete logarithm problem. Therefore, mapping between the classical Diffie-Hellman key exchange scheme and its equivalent using elliptic curve paradigm can be simply performed by replacing the exponentiation operation by an integer multiplication (or more precisely n-time addition) in $E(F_p)$.

2) Individual keys establishment : Individual keys are established between each sensor node and the base station during 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, the elliptic curve $E(F_p)$ according to the second form as discussed above, and a generator point $G \in F_p$. Then, it generates its private key, $x_B \in \mathbb{Z}_p$, where $2 \leq x_B \leq p-1$ and calculates the corresponding elliptic curve public key, $Y_B = x_B G$. The parameters p, $E(F_p)$, G , Y_B, and an initial key K_0 will be pre-loaded in each deployed sensor node. The initial key, K_0 will be used to verify the genuineness of the deployed sensor nodes. It is valid only during the short period of the initial deployment phase and will be deleted immediately after the accomplishment of the key establishment procedure. In the following , we denote by N the total number of deployed nodes, where each node is uniquely identified by an id value.

Private/public key pair generation and individual key calculation: Immediately after network deployment and the establishment of clustered communication architecture, every node i, $1 \leq i \leq N$, will generate its private key, $x_i \in \mathbb{Z}_p$. This is performed by applying a hash function as follows:

$$x_i = Hash(id_i||K_0||N_i)mod(p) \tag{4}$$

where N_i is a randomly generated nonce. This generation procedure ensures that all private keys are different from each other, which can enable data origin authentication.

Then, the sensor node calculates its elliptic curve public key, $Y_i = x_i G$. At this point the sensor node is able to calculate its individual pairwise secret key, $K_i = x_i Y_B = x_i x_B G$. The sensor node sends its public key Y_i to the base station to be validated and stored in the public keys repository. The message is authenticated by a Hash Message Authentication Code (HMAC) using K_0, to ensures that it is sent by a genuine deployed sensor node.

Public key validation and individual key establishment in the base station: After verifying the identity of the sensor node and the MAC of the received request, the base station validates the public key of the sensor node, saves it in its public keys repository, and establish the shared individual pairwise key, $K_i = x_B Y_i = x_i x_B G$. Finally, the base station sends to the sensor node an acknowledgment that is authenticated by a MAC calculated using the individual key, K_i. Then, the sensor deletes immediately the initial key, K_0 from its memory.

3) Intra-cluster pairwise keys and cluster key establishment : Intra-cluster pairwise keys must be established to secure communication between each sensor node and its cluster head. In addition, the sensor node can establish a pairwise key with each one of its node neighbors in order to communicate with the cluster head. 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 public values must be retrieved from the base station and each party verifies its validity before key establishment. To this end, the base station calculates a MAC of the public key using the individual key shared with the sensor node that had generated the key request message. After that, neighbor nodes i and j can establish a pairwise key, $K_{ij} = x_j Y_i = x_i Y_j = x_i x_j G$. Similarly, the pairwise key $K_i^c = x_i Y_c = x_c Y_i = x_i x_c G$ can be established between the node i and its CH c.

To set the cluster key a group communication secret key sharing procedure was proposed. This scheme is more efficient than the existing techniques [28] 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

$$Y_{cj} = x_c \sum_{n=1,\, n \neq j}^{m_c} Y_n \tag{5}$$

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

$$K^c = K_j^c + Y_{cj} = x_c \sum_{n=1}^{m_c} Y_n \tag{6}$$

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, K_B and transmits it to all CHs. In the second stage, each CH, c decrypts the network key and encrypts it with the cluster key, K^c before broadcasting it to all cluster members.

5) Session keys derivation: All the keys that they had been described in last subsections are note used directly to ensure data confidentiality and authentication. To this end, some session keys are derived from these keys. This is performed by the following procedure.

The elliptic curve approach, allows the sharing of a point with two coordinates. Therefore, each key K is a point in $E(F_p)$ that is composed by abscissa, K_x and an ordinate K_y. In order to be conform to the rule of separation between the keys for encryption and authentication, two session keys denoted as, Ke_i, and, Ka_i are derived from the abscissa and the ordinate of the key K. These two keys are used respectively for encrypting and generating the MAC of each message exchanged between the sensor node and any other entity of the network. Here, the key K denotes any kind of the described keys such as individual key, pairwise key, cluster key, inter-cluster key, and a network key. New session key is derived for each session. If i is the current session number, the session keys used securely exchange messages during this session is given by the following formula

$$Ke_i = Hash(K_x, Ke_{i-1}, MAC_{i-1}) \qquad (7)$$

$$Ka_i = Hash(K_y, Ka_{i-1}, MAC_{i-1}) \qquad (8)$$

MAC_{i-1} is referred to the message authentication code of the packets that had been correctly received during the last session by the entity with witch the key K was shared. Similarly, $Ke_{i-1}$and $Ka_{i-1}$are the session keys used in the last session for ensuring encryption and authentication during the last session. As it will be explained later in this paper, the selected derivation procedure can ensure backward and forward secrecy of the transmitted data and resilience to node capture and replication attacks.

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 initial 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 from 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[29] based on the Lagrange interpolation. This approach consists in randomly selecting a polynomial function, $f(x) = K_r + a_1 x + a_2 x^2 + ... + a_{t-1} x^{t-1} mod(Q)$ 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 $(S_1, S_2, ..., S_t)$ using the following formula

$$f(x) \quad = \quad \sum_{i=1}^{t} S_i \left(\prod_{i \neq j} \frac{x - id_j}{id_i - id_j} \right) mod(p) \qquad (9)$$

Particularly, the key K_r can be reconstructed by applying the equality

$$K_r = f(0) = \sum_{i=1}^{t} S_i \left(\prod_{i \neq j} \frac{id_j}{id_j - id_i} \right) mod(Q) \qquad (10)$$

IV. Security Analysis

The evaluation of the security schemes intended for WSNs is significantly different from those used in conventional networks. Indeed, the evaluation criteria should consider the characteristics of the WSNs deployment and their resource constraints. In this section we analyze the security level offered by our key 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) the vulnerability to energy depletion attack.

A. Backward and forward security

The forward security propriety is to prevent the possibility to an attacker to predict a future key if he captures a currently used key. On the other hand, backward security is to preclude an attacker from obtaining information about previously used keys when he can capture the currently used key. These two proprieties are very important in key distribution schemes to ensure data confidentiality. To ensure the forward and backward secrecy, our proposed key distribution scheme is based on public key encryption paradigm where at each re-keying period the private and public keys of any sensor node are generated independently of any previously used keys. Therefore, all symmetric keys established between network entities are not derived from any used key and are recalculated based on the newly generated public and private key pairs. In addition, no future keys must be encrypted by currently used key to be shared. Moreover, all group communication keys are modified each time a change in the network topology occurs in the sensor level or in the cluster level.

B. Node capture

In many applications, sensor nodes are usually randomly deployed by aerial dropping in 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.

On the other hand, a sensor node can be physically captured. In our case, an attacker can capture either a sensor node or a cluster head. When an attacker captures a sensor node, he can access to the individual keys it shares with the base station and the cluster head, and the cluster key it shares with all members of its cluster. The later key can affect security within the cluster and should be modified by eliminating the public value of the captured node in the key calculation operation. In addition, the sensor node stores a single part of the shared secret used to reconstruct the network re-initialization key. To prevent the discover of this key the number of captured nodes should not exceed the degree of the polynomial function, t.

Besides, getting access to a cluster head is more critical than in the case of a sensor node. In this case, the attacker can access to the individual pairwise key shared between the cluster head and the base station, all individual keys shared between the cluster head and each sensor node, the cluster key, and the inter-cluster key shared with all other cluster heads and the base station. Therefore, all group communication keys must be changed by recalculating the keys without the public value of the compromised cluster head. Also, the cluster member should establish another individual keys with other cluster heads.

In addition, several techniques can be used to prevent that an adversary can access to the content of a sensor or a cluster head such as triggering of a physical auto-destruction, or a soft erasing of the content of all memories when an attempt to access to the sensor is detected. These techniques are mainly appropriate for cluster heads which encompasses a large quantity of information.

C. 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, cluster head, or even the base station. These nodes should be detected and isolated from the network. In the sequel, we describe how the proposed scheme can resist to cloning attacks according to different situations.

1) Node duplication at the initial deployment phase: In this situation, the attacker tries to add new nodes to the network with copied identities at the initial deployment phase of the network. The inserted nodes will attempt to generate private and public keys and establish symmetric keys with the cluster heads and the base station. The proposed scheme can guarantee resistance against this attack. Indeed, before establishing connections in the network any new node should generate a private key based on its identity and the secret initial key. All initial keys are eliminated from the memory after the deployment of sensor nodes, the generation of their private keys, and the validation of the corresponding public keys. Consequently, these keys cannot be recovered by capturing already deployed sensor nodes. Furthermore, the base station tracks all identities of the deployed nodes and validates and distributes public keys to all entities requesting the establishment of secret keys. Therefore, unused or compromised public keys can be revoked by the base station. Consequently, 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.

2) The replication of an active node or a sleeping node: In this case the attacker will replicate a number of already deployed nodes that are either in an active state or in a sleeping state. We suppose that the attacker will get access to the private/public key pair of the cloned node. The attacker will try to affect the copies of the replicated nodes to different cluster in order to avoid their detection. In fact, if a clone tries to connect to the same cluster as the original node, it can be easily detected by the cluster head. This can be performed by identifying traffic patterns. For example, the cluster head can notice that two packets are transmitted in a very short period from the same node identity using two different routes. In this case, it can either trigger the revocation of the two nodes or, based on its history, it can detect the false node. Also the inserted node will not be able to reconstruct the session keys which are generated in every session based on the packet transmission history.

Consequently, the clones will try to establish shared keys with new clusters. In our scheme, we can detect copies of nodes even in this situation. As it had been previously detailed, when the new cluster head receives the key establishment request from the false node. It first consults the base station to gather information about its original cluster. After that, it informs the old cluster which verifies the connectivity status of the node by sending a beacon message to see if the node had effectively left the cluster. If it detects that the original node is still connected to its cluster it will communicate this information to the new cluster head which isolates the false node.

3) The replication of disconnected sensor nodes: In this scenario, the intruder will try to insert duplicated copies of a node that is disconnected from the network due to some reason. The replicated nodes are configured with the private and public keys of the original node. They try to establish keys with different cluster heads. The main problem in this situation is that the original cluster head is not able to detect if the node is still connected to its original cluster or not. To resolve this problem we use authentication using the last known session key of the node. In other words, when the false node wants to establish new secret keys, we will first verify that the original node is not connected to its original cluster by applying the procedure described in the last subsection. Then, the new cluster head will challenge the node by requesting that is encrypts a given packet using its last session key. The encrypted message is then sent to the original cluster head to verify the genuineness of the node. This procedure allows an efficient detection and isolation of the false nodes even if they are deployed in many clusters.

D. Energy depletion attack

Sensor nodes are battery based devices with a very limited life time. Hence, energy management is a very important issue in wireless sensor networks. An attacker can try a denial of service by sending many false key establishment requests with different identities in order to deplete the available energy of a sensor node. We can classify situations where this attack can be performed in three classes:

1) Attacks performed during the identification process: In this situation, a certain number of malicious nodes try to send an important number of false keying requests to the base station in order to make intermediate nodes that are relaying the message, out of energy. One solution to combat this attack is that when the base station receives a number of false keying requests from a specific source higher than a specific value, it will send a message to the neighbors of the source to not relay any packet from it in the future. The attacking node is therefore detected and isolated. However, this solution generates false positives and does reduce completely such an attack.

2) Attacks performed during key establishment between sensor nodes: In this case, a malicious node will send false key establishment requests to a neighbor that will execute costly processing operations that deplete its available energy. Our scheme can prevent this kind of attacks because sensor nodes will not perform any costly key establishment operation before validating the identity and receiving the appropriate public key from the base station. Also, we can set that if a sensor node receives a number of key establishment requests with invalid public keys it will isolate the source of these requests.

3) Attacks performed during data transmission: In this case, the attacker will try to send an important number of false encrypted messages where the destination will apply the costly power consuming and unnecessary decryption procedure. This can dangerously reduce the available energy of the sensor node. One solution for this problem is that the base station stores a profile for each sensor node transmission. The profiling operation can be based, for example, on transmission frequency and sampling. If a transmission deviates from a given profile by a certain threshold, the base station will order the neighbor nodes of the source to not relay any new packet from it.

V. Performances Analysis

In this section, we assess the performance of the proposed key distribution scheme in terms of scalability, key storage requirement, communication overhead and computation power cost.

A. Scalability

The scalability is the ability of the scheme to maintain an acceptable security level regardless of the network size. This is very important is wireless sensor networks that usually encompass a very large number of sensor nodes. To be scalable the number of encryption keys managed by each sensor must not extensively increases when the number of nodes increases in the network. This is due to the limited storage capacity of sensor nodes.

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 the resource consumption in the network.

B. Key storage requirement

To provide security for data transmission, in any key distribution scheme, each sensor node should store and manage a specific number of keys in its memory. Due to the large size of WSNs and the limited memory capacity of sensors, the amount of the consumed memory, needed for storing keys is a very important parameter. In our scheme, every sensor node should store very limited number of keys. These are the private key, individual key shared with the base station, an intra-cluster pairwise key established with the cluster head, a cluster key, and the network key. Also, in case where a sensor does not have a direct connectivity with the cluster head, it should share individual keys with the neighbors that are closer to the cluster head to relay transmitted packet. Although, the number of these keys will depend on the connectivity level of the cluster, it will be very limited. Typically, a sensor node will need to set shared keys with two of its neighbors in order to ensure communication reliability. Consequently, by using an appropriate clustering model, the storage capacity required to manage the encryption keys in each sensor node will decrease.

On the other hand, the cluster head will need higher storage capacity than sensor nodes. Indeed, in addition to the encryption keys managed by normal nodes, this device should store an individual pairwise key with each sensor belonging to the cluster. Furthermore, it should set an individual key with neighbor cluster heads and manage the intra-cluster key. For this reason, cluster heads should be equipped with higher storage resources. The amount of needed memory capacity will depend on average number of sensor that can compose the cluster.

Besides the few number of keys generated by the proposed scheme in each sensor, the use of elliptic curve encryption reduces the keys size. The public keys managed in every node have a size that is almost similar to that of secret encryption keys. This contributes also in decreasing the memory occupancy needed for storing keys in each sensor bonging to the wireless sensor network.

C. Communication Overhead

The communication overhead is referred to the number of exchanged messages needed to achieve keys establishment between different entities of the network. This parameter is very important in WSN due to the fact that communication procedures are the most energy consuming tasks. Therefore, an efficient key distribution scheme should minimize the communications required to share different kinds of keys while ensuring an acceptable security level and effective data management procedure by enabling the in-network processing capability.

In the proposed key distribution scheme, each sensor node needs to exchange two messages with the base station to validate its public key and generate the individual pairwise key. Three other messages and an acknowledgment are required to establish intra-cluster pairwise key with the cluster head and the cluster key. A last message and an acknowledgment are exchanged between the sensor node and the cluster head to share the network key that is sent encrypted with the cluster key. These communication messages should be exchanged independently of the network size and connectivity.

Moreover, each sensor that has not direct connectivity with the cluster head should share keys with neighbors that can relay its packets. This will generate the need for exchanging at least two packets per neighbor to retrieve the public keys of the nodes. However the number of these messages will be very limited and can be significantly reduced if the number of clusters and their deployment is chosen adequately.

Withal, the mobility of nodes or the deployment of new nodes can require extensive exchange of messages to perform group keys update between the cluster heads. However, this is less critical because this devices are supposed to have sufficient energy to sustain these operations.

D. Computation power requirement

Another performance parameter for any key distribution scheme is the computation power required to perform key distribution. Indeed, sensor nodes are tiny devices that are endowed with a cheap processor having a very limited processing capability. In public encryption scheme, the most complicated operations are the computation of the public keys and the establishment of shared keys using the Diffie-Hellman elliptic curve key exchange procedure. Arithmetic operations in the elliptic curve Galois Field are shown to have little complexity compared to conventional public key encryption. Furthermore, sensor nodes execute a very limited number of these operations which are triggered during initial deployment, topology changes, and re-keying procedure. This is due to the clustering topology adopted in our scheme where the sensor node will share keys with only the base station, its cluster head and a very limited number of its neighbor nodes belonging to the same cluster.

Table I
SIMULATION PARAMETERS

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
MAC computation energy (SHA1)	5.9$\mu J/Byte$
Encryption/Decryption energy (AES)	1.62/2.49 $\mu J/Byte$

However, the re-keying procedure can have also an extensive computation cost. In this case, the sensor should recover the re-initialization key after collecting $t-1$ partial secrets. This operation depends on the threshold t that must be appropriately selected to make a trade-off between security and computation complexity.

VI. PERFORMANCE EVALUATION

In this section we assess the performance of the proposed scheme with regard to the required key storage capacity, communication overhead, and energy consumption. In a first set of simulations, we compare the performance of the proposed security to LEAP [18], [22]. In the second the part of the performance evaluation work we assess the scalability of our scheme by evaluating its performance with regard to the number of clusters for different network sizes. Finally, the last set of simulations is devoted for evaluating the performance of the re-keying procedure.

To this end, we developed a simulation model using the Matlab tool. We consider a clustered topology and we compute performance parameters by varying the number sensor nodes. The number of cluster in each topology is taken as : $M = \lceil 0.05 * N \rceil$ where N is the number of sensor nodes. In the implementation of the simulation model we used the values given by table I.

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 for every network. The final results are obtained by taking the average on all values measured for all generated topologies. The maximum number of topologies (5) is selected based on the observation that this value guarantees a confidence interval of more than 90%.

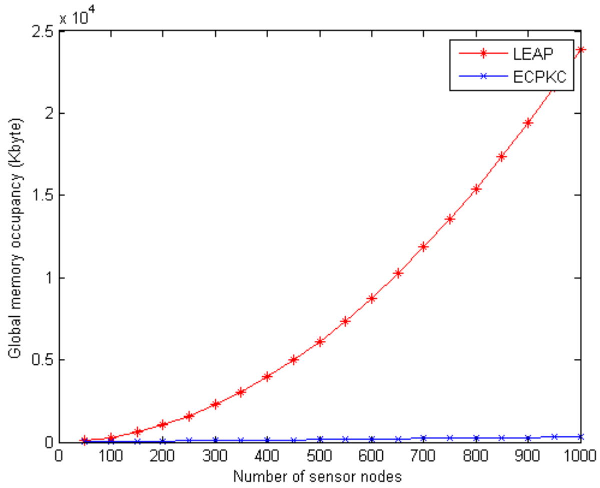

Fig. 2. Memory occupancy

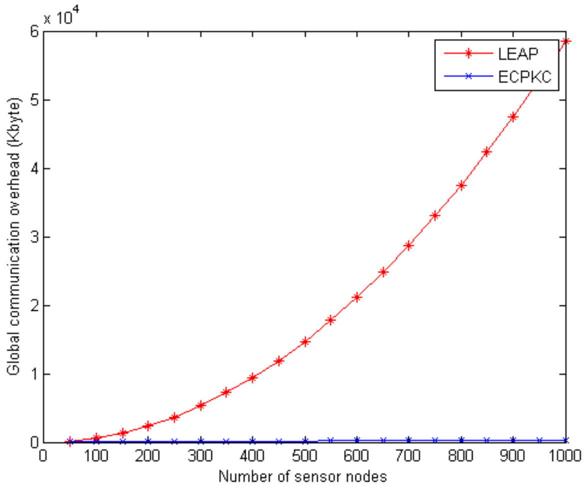

Fig. 3. Communication overhead

A. Comparison with the LEAP protocol

We compare the performances of our scheme to those of the LEAP scheme[18], [22] which implements the in-network processing concept but using symmetric pairwise key pre-distribution paradigm. In each simulation, we execute the proposed elliptic curve public key cryptography based approach, denoted as ECPKC, and the LEAP scheme, on a set of randomly generated topologies composed of a number of sensors with one sink node.

Figure 2 depicts the required storage capacity for managing key distribution in the proposed ECPKC. We can notice that our scheme has remarkably reduced memory occupancy when compared to the occupancy of the LEAP protocol. Moreover, the needed storage capacity of our scheme varies 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 manage a limited number of public/private keys and symmetric keys that are shared with the base station and the cluster head. Also, each node shares several symmetric keys with its neighbors that can not directly reach the cluster head. 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 will increase with the density of the network.

The same observation can be formulated for the communication overhead presented by Figure 3. In our the ECPKC approach the sensor nodes will initiate key exchange procedure with the base station, the cluster head, 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 the sensor node and its cluster head.

An important parameter for any key distribution scheme is energy consumption. Figure 4 shows the total energy consumption of the proposed scheme compared to the energy consumption of the LEAP scheme. It can be observed that elliptic curve based scheme needs less energy and that it varies linearly with the number of sensor nodes. This can ensure the scalability of our scheme to large scale networks.

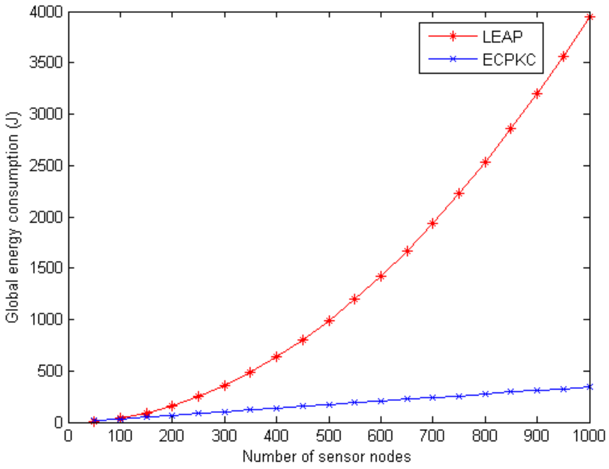

Fig. 4. Energy consumption

B. Evaluation of scalability

In this subsection, we evaluate the scalability of the proposed key distribution scheme. To this end we measure variation of the memory occupancy, the generated communication overhead, and the consumed energy in function of the number of clusters for different network sizes. The simulation results are depicted by Figures 5,6, and 7. We can observe that the needed storage capacity, the communication overhead, and the overall energy consumption of our scheme decreases very fast with the increasing in the number of clusters that compose the network. Also, we can notice the existence of an optimal value for the number of cluster from which the performances become almost constant. This value is around 5% of the number of nodes composing the network. This can be explained by the

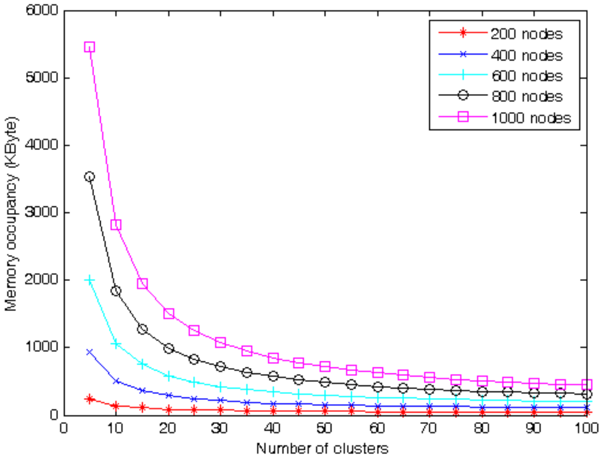

Fig. 5. Variation of memory occupancy in function of the number of clusters for different network sizes

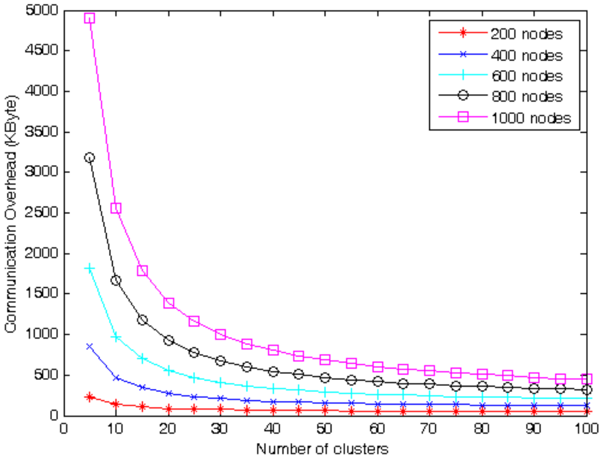

Fig. 6. Variation of the communication overhead needed to establish keys in function of the number of clusters for different network sizes

fact that when we divide the network in a higher number of clusters the number of sensor nodes that can connect directly to its cluster head increases. However, the limit is reached when all nodes of the network can be directly connected to its cluster head without any relay. This corresponds to the aforementioned optimal value of the of clusters.

C. Evaluation of the re-keying mechanism

In this subsection, we present results of simulation work conducted to the evaluate the re-keying procedure of the presented scheme that is based on threshold secret key sharing technique. In these simulations the number of sensor nodes composing the network is fixed to 1000 nodes which are divided into 50 clusters. We assess the communication overhead and the energy consumption variation in function of the re-keying period for different values of the threshold t. We execute each simulation during 3600 minutes. At the end of every re-keying interval every sensor node will share its individual secret with $t-1$ nodes to recover the re-initialization key, K_r and after that it executes the key establishment procedure of our scheme. Figures 8 and 9 present the simulation results. We can notice that increasing the re-keying interval contributes to decreasing the communication overhead and the energy consumption. However, for a re-keying period higher that 30 minutes the variation of the performance parameters becomes almost linear. This values can be considered as an optimal values for the re-keying period. We can see also that when the threshold value increases the communication overhead increases. Nevertheless, the variation is less important for the energy consumption parameter.

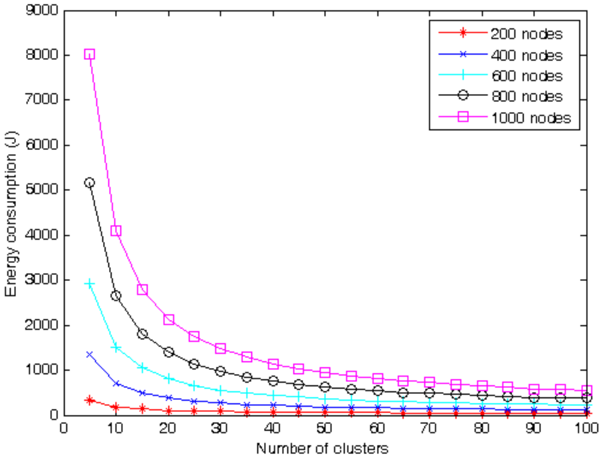

Fig. 7. Variation of energy consumption during key establishment process in function of the number of clusters for different network sizes

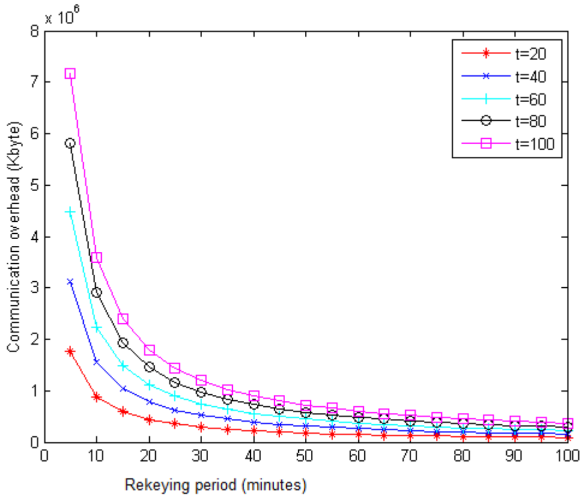

Fig. 8. Variation of the communication overhead in function of the re-keying period for different threshold values

VII. Conclusion

Key distribution and management in WSNs is much more difficult to achieve than 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

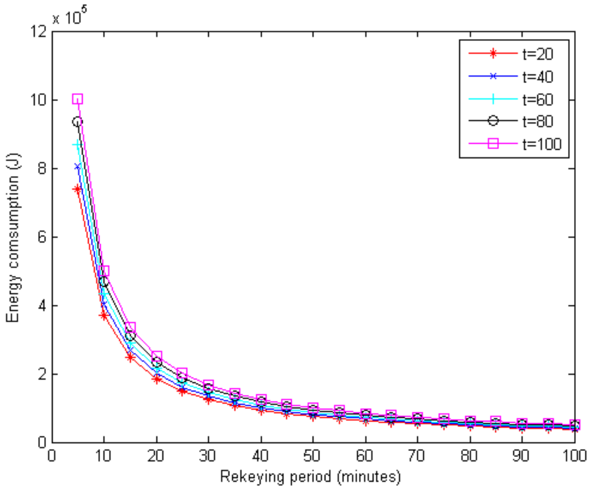

Fig. 9. Variation of energy consumption in function of the re-keying period for different threshold values

limited computation 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 proposed key distribution scheme and ensuring adaptive security in WSNs can be envisioned in a future work.

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] R. Belleazreg, N. Boudriga, and S. An, "Border surveillance using sensor based thick-lines," in *the proceedings of the 27th International Conference on Information Networking (ICOIN 2013)*, Bangkok, Thailand, January 2013, pp. 221–225.

[2] D. krichen, W. Abdallah, and N. Boudriga, "WSN-based flutter control application for aircraft wings structural health monitoring," in *the proceedings of the 29th Symposium on Applied Computing (SAC 2014)*, Gyeongju, South Korea, March 2014.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications Magazine, IEEE*, vol. 40, no. 8, pp. 102–114, August 2002.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *proceedings of the Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004* , March 2004, pp. 586–597.

[9] S. A. Çamtepe and B. Yener, "Combinatorial design of key distribution mechanisms for wireless sensor networks," *IEEE/ACM Transactions on networking*, vol. 15, no. 2, pp. 346–358, April 2007.

[10] Y. Cheng and D. P. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks," *journal of ad hoc Networks*, vol. 5, no. 1, pp. 35–48, January 2007.

[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] V. Thiruppathy Kesavan and S. Radhakrishnan, "Secret key cryptography based security approach for wireless sensor networks," in *proceedings of the International Conference on Recent Advances in Computing and Software Systems RACSS 2012*, Chennai, April 2012, pp. 185–191.

[13] 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.

[14] 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.

[15] W. Abdallah, N. Boudriga, D. Kim, and S. An, "An efficient and scalable key management mechanism for wireless sensor networks," in *the proceeding of the 16th international Conference on Advanced communication technology (ICACT 2014)*, Phonix Parc, South Korea, February 2014, pp. 686–692.

[16] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, "A survey of key management schemes in wireless sensor networks," *Journal of Computer Communications*, vol. 30, no. 11-12, pp. 2314–2341, September 2007.

[17] 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.

[18] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, Washington D.C, October 2003 2003.

[19] Q. Jing, J. Hu, and Z. Chen, "C4w: An energy efficient public key cryptosystem for large-scale wireless sensor networks," in *proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems MASS 2006*, Vancouver, BC, October 2006, pp. 827–832.

[20] W. Zhang, S. Zhu, and G. Cao, "Predistribution and local collaboration-based group rekeying for wireless sensor networks," *Journal of Ad Hoc Networks*, vol. 7, no. 6, pp. 1229–1242, August 2009.

[21] X. He, M. Niedermeier, and H. de Meer, "Dynamic keymanagement in wireless sensor networks:asurvey," *Journal of Networkand Computer Applications*, vol. 36, no. 2, pp. 611–622, March 2013.

[22] 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.

[23] R. Divya and T. Thirumurugan, "A novel dynamic key management scheme based on hamming distance for wireless sensor networks," *International Journal of Scientific and Engineering Research*, vol. 2, no. 5, pp. 1–7, May 2011.

[24] X. Zhang, J. He, and Q. Wei, "Eddk: Energy-efficient distributed deterministic key management for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, 2011.

[25] 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.

[26] S. K. Gupta, N. Jain, and P. Sinha, "Clustering protocols in wireless sensor networks: A survey," *International Journal of Applied Information Systems*, vol. 5, no. 2, pp. 41–50, January 2013, published by Foundation of Computer Science, New York, USA.

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

[28] 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.

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

Walid Abdallah is an Assistant Professor at the aviation school of Borj Elamri, Tunisia. He received his PhD in Information and communication technologies and the Diploma of engineer in telecommunications from the School of Communications Engineering (Sup'Com), Tunisia. He received his Master Diploma from the National School of Engineer of Tunis (Tunisia). From 2001 to 2005 he worked for the National Digital Certification Agency (NDCA, Tunisia) and from 1997 to 2001 he worked for the national telecommunication operator (Tunisia Telecom). Currently, he is a member of the Communication Networks and Security Lab, where he is conducting research in optical networks and wireless sensor networks.

Noureddine Boudriga is an internationally known scientist/academic. He received his PhD in algebraic topology from University Paris XI (France) and his PhD in computer science from the University of Tunis (Tunisia). He is currently a full Professor of Telecommunications at the University of Carthage, Tunisia and the Director of the Communication Networks and Security Research Laboratory (CNAS). He has served as the General Director and founder of the Tunisian National Digital Certification Agency. He is the recipient of the Tunisian Presidential award in Science and Research (2004). He was involved in very active research and authored and co-authored many journal papers, book chapter, and books on networks and security.

Daehee Kim received the B.S. degree in Electronics Engineering from Yonsei University, Korea, in 2003 and M.S. degree in Electronic and Computer Engineering from Korea University, Korea, in 2006. Currently, he is working for Ph.D. degree on Electronic and Computer Engineering in Korea University, Korea. His research interests include the wireless sensor network, LTE, and security in wireless networks.

SunshinAn received the B.S. degree from Seoul National University, Korea in 1973, and the M.S. degree in Electrical Engineering from KAIST (Korea Advanced Institute of Science and Technology), Korea in 1975 and the Ph.D. degree in Electric and Information from ENSEEIHT, France in 1979. He joined the faculty of Korea University in 1982, where he is currently a Professor of Electronic and Computer Engineering. Prior to joining Korea University, Prof. An was Assistant Professor of Electronic Engineering in Ajou University, Suwon, Korea. He was with NIST (National Institute of Standards and Technology) in U.S.A., as a visiting scientist in 1991. His research interests include the distributed system, communication networks and protocols, information network, intelligent network, multimedia communication system, wireless sensor network and mobile RFID network.