Efficient and energy-aware key management framework for dynamic sensor networks

Mawloud Omar, Imene Belalouache, Samia Amrane, Bournane Abbache

Efficient and energy-aware key management framework for dynamic sensor networks

Mawloud Omar, Imene Belalouache, Samia Amrane and Bournane Abbache

Laboratoire d'Informatique Médicale, Faculté des Sciences Exactes, Université de Bejaia, 06000 Bejaia, Algérie.

Abstract

Wireless sensor networks consist of a set of connected devices deployed to report sensitive environmental data. Key management in wireless sensor networks remains a challenging issue due to the limited resource capacity of devices. Most existing solutions focus only on the key storage and updating optimization giving less attention to the mobility, which is more needed in the nowadays applications. In this paper, we propose a secure and efficient key management system with mobility support. The proposed scheme is based on hybrid key establishment to meet both the robustness and efficiency requirements. The sensor nodes can be mobile, where they could leave, rejoin their cluster, or join other ones. We incorporate lightweight techniques for sensor node integration, departure, revocation and key updating. Its efficiency is evaluated by comparison with other concurrent schemes, where it demonstrates the best results.

Keywords: Wireless sensor network, Security, Key management, Mobility, Energy consumption.

1. Introduction

In the last two decades, the tremendous advances in smart micro-devices, wireless communications and mobile robotics offered researchers the opportunity to tackle an important real-world problem: sensing, monitoring and remote control of complex processes distributed within unstructured, dynamic or even hostile environments. As a result, the development of fully-autonomous networks of collaborative devices being able to adapt for complex situations, to effectively react for unpredictable events and to control critical processes within their coverage area. The road towards this desideratum is marked by an important conceptual milestone: Wireless Sensor Networks (WSNs) [1].

The sensor is a tiny embedded system with low cost and wireless transmission, able to sense different physical phenomena, as temperature, humidity, presence, fire, etc. This technology has modernized the domain of networking, due to their ability to gather and transfer different types of information from the real environment. WSNs are becoming highly present in several types of application domain [2, 3, 4], as the environmental applications (e.g., fire detection, pollution supervising, etc.), the industrial

applications (e.g., machine monitoring), the healthcare systems (e.g., medical remote diagnostics), the military applications (e.g., troop support), etc. The sensor nodes are restricted in terms of resources. In case of a failure, even of a single sensor node, the whole network will require to be reconfigured. The network lifetime depends on the individual lifetime of sensor nodes. The sensor node failure may be caused by the battery depletion. A significant aspect that affects the consumption of power is that each sensor node consumes extra-power when routing packets of other sensor nodes. These are the main purposes for which the efficiency of power consumption is one of the principal problems that should be addressed when designing protocols for this kind of networks.

Due to the resource, space, and cost constraints of sensor nodes, the security solutions for classical networks are not suitable for WSNs. Security in WSNs is a very active research area, where new and creative solutions to the security issues are suggested on a regular basis [5]. One security aspect that receives a great deal of attention in WSNs is the key management. WSNs are unique in this aspect due to their size, mobility and computational/power constraints. Indeed, researchers envision WSN to be orders of magnitude larger than their traditional embedded counterparts. This, coupled with the operational constraints described previously, makes secure key management an absolute necessity in most WSN designs. Encryption and key management/establishment are so crucial to the defense of a WSN, with nearly all aspects of WSN defenses relying on solid encryption [6]. In WSNs, the key management is a challenging issue. Centralizing a key management server, for example on the Base Station (BS) is an impractical solution. The main problem with this approach is that the central server becomes a target of attacks. Nonetheless, when such a server is available and secure, these approaches may become very attractive. In the other hand, designing a distributed scheme may involve additional overhead on sensor nodes. The key pre-distribution and updating must be adapted to the resource constraints of WSNs. The symmetric encryption seems at first the most adapted technique, however, is less robust against the passive attacks. In the other hand, the asymmetric encryption enhances the robustness aspect, but involves overhead in terms of computation. In this paper, an efficient and dynamic key management scheme is proposed for dynamic WSNs. The principal purpose of this work is to address the security problems involved by the key distribution schemes, where the main contributions are as follows:

- We propose a secure and efficient key management system, taking in charge the inherent characteristics of WSN environments;
- The proposed scheme supports the sensor mobility;
- The proposed scheme is based on hybrid key establishment to meet both robustness and efficiency;

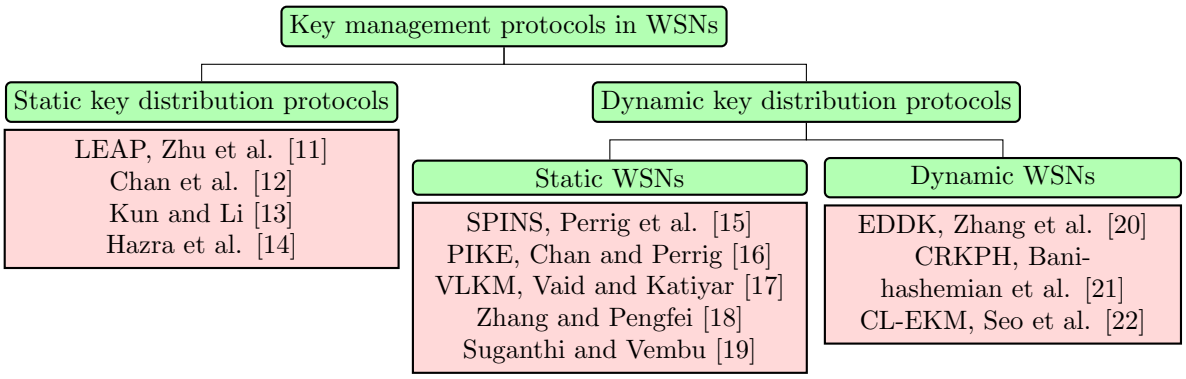

Figure 1: Classification of the reviewed solutions

- The proposed scheme incorporates lightweight techniques of sensor integration, departure and exclusion;
- The proposed scheme involves a reduced load in terms of key storage, communication, and energy consumption.

The remaining of this paper is structured as follows. In Section 2, we review some relevant and recent schemes from the literature. In Section 3, we present the system model and in Section 4, the detailed description of the proposed solution. In Section 5, we analyze its performances with comparison to two concurrent protocols. Finally, we conclude this paper in Section 6.

2. Related work

There are hundreds of consistent works which address the key management problem in WSNs. The authors of [7, 8, 9, 10] summarize a good representative part of them. In this section, we present some relevant and recent schemes. We have classified them into two main categories, namely the static and dynamic key distribution solutions. We have further classified the latter category into solutions for the framework of static and dynamic networks. We illustrate in Figure 1, the classification of the reviewed solutions.

2.1. Static key pre-distribution

In [11], Zhu et al. have proposed LEAP (Localized Encryption and Authentication Protocol). The proposed protocol operates over a symmetrical-based scheme providing confidentiality and authentication. This protocol is executed in two phases: key management phase and broadcast authentication phase. The proposed protocol generates and updates four types of key for each sensor node: an individual key

shared with the BS, a pairwise-key shared with another sensor node, a cluster-key shared with multiple neighboring nodes, and a group-key shared by all the sensor nodes in the network. LEAP provides broadcast authentication on one-way key chains, in which the BS authenticates the control packets of all the network sensor nodes.

In [12], Chan et al. have proposed a q-composite random key pre-distribution scheme, where any two neighboring sensor nodes need to find q common keys from their key rings to establish a secure link. The choice of q depends on the key pool size and the required probability of successfully performing key-setup with some neighbors. They have also proposed a multi-path key reinforcement to enhance the security of an established communication link-key. The established key between two sensor nodes may be residing in some other nodes. The proposed approach consists of identifying k independent secure paths created during the initial key-setup and send k random values via them. They have also proposed an improved random-pairwise key scheme. It is based on the observation that not all the keys need to be stored in the sensor node key ring to have a connected random graph with high probability. Each sensor node needs to store a random set of pairwise-keys chosen through a specific probability.

In [13], Kun and Li have proposed a key pre-distribution scheme based on the probability theory and the hash functions. A key pool is generated and for each sensor node is assigned a part of these keys. In order to ensure the secure communication between two sensor nodes, a process of key discovery is executed. The sensor nodes discover their neighbors with which they share at least a common key by diffusing their part of the key pool.

In [14], Hazra et al. have proposed a key pre-distribution scheme based on the hashed key chain. Before the network deployment, a key generator center uses a hash function in order to establish a common key with other sensor nodes basing on the generation of linear key chains. A number of key chains are generated such as a key chain is formed by merging two hash chains with equal size. Then, the key chains are generated and a set of key ring is computed and assigned to a sensor node. The sensor nodes diffuse the key identifiers of their key rings and the common key list between each pair of nodes will be used as pairwise-keys.

2.2. Dynamic key distribution for static WSNs

In [15], Perrig et al. have proposed SPINS (Security Protocols for Sensor Networks). SPINS is composed of two security protocols for sensor networks: SNEP (Secure Network Encryption Protocol), and μTESLA (Micro Timed Efficient Stream of the Loss-tolerant Authentication). SNEP is a symmetrical-based scheme using MACs, providing a dynamic encryption. Even in case of the same information, the

encryption process gives different output at each time within the lifetime of a sensor node. μTESLA is a symmetrical-based scheme using MACs with delayed key disclosure minimizing the key distribution overhead. In order to send an authenticated data packet, the BS computes the packet's MAC with a secret-key. Upon receiving the packet, the sensor node saves it in its memory and waits for the secret-key disclosure. Timely, the BS broadcasts the verification key allowing the receiver node to authenticate the received packet.

In [16], Chan and Perrig have proposed PIKE (Peer Intermediaries for Key Establishment). For each sensor node is associated an identifier under the form (x, y). A sensor node (x, y) shares a common key with nodes that have the same value of either x or y. New keys will be established between the neighboring sensor nodes. These keys are used in an identical manner as the original pairwise-keys. PIKE uses a simple sensor as a trusted intermediate node in the key establishment process.

In [17], Vaid and Katiyar have proposed VLKM (Virtual Location-based Key Management). VLKM uses two types of key, namely the sensor node key and the cluster-key. The sensor nodes map their virtual locations following their cluster-head (CH). Each sensor node has an initial virtual location, a virtual angle of movement, a speed and a virtual direction. Each sensor node applies a hash function on its both identifier and virtual location in order to generate its own key.

In [18], Zhang and Pengfei have proposed a hybrid key management method for heterogeneous WSNs based on Deffie-Hellman exchange protocol and Elliptic Curve Cryptography (ECC). This protocol manages four types of key, namely the key pair (public and private), pairwise-keys, session-keys and the cluster-keys. The BS is preconfigured with all the CH public-keys, each CH with all the member node public-keys, and each sensor node with its own private-key. Upon the network deployment, the CH establishes a pairwise-key with each member node. The CH generates and diffuses in the cluster a random number with which each pair of sensor nodes establish a session-key, and collectively the cluster-key.

In [19], Suganthi and Vembu have proposed an algorithm allowing the establishment of three types of keys for each sensor node without key broadcasting and a minimal BS involvement. The sensor node individual key is used for initial communication with the BS and is calculated through system parameters distributed before the network deployment. The system parameters include a shared pseudo random function, an initial key and the individual sensor node identifier. A pairwise-key is used to secure the communication between two neighbors and a group-key is shared by all the network sensor nodes. These keys are calculated dynamically using a polynomial function among a set of preconfigured functions in the sensor nodes. The pairwise-keys and the group-key are periodically updated by changing the coefficient of the polynomial function.

2.3. Dynamic key distribution for dynamic WSNs

In [20], Zhang et al. have proposed EDDK (Energy-efficient Distributed Deterministic Key management). EDDK manages two types of key, namely the cluster-keys and the pairwise keys. In each sensor node, an initial key is precharged with which it computes the pairwise-keys. The sensor nodes diffuse a join message in order to discover their neighboring and generate the corresponding pairwise-keys. If a sensor node moves to a new location, it will establish pairwise-keys with its new neighbors and its old neighbors will also revoke the corresponding pairwise-keys.

In [21], Banihashemian et al. have proposed CRKPH (Centralized Replacement Key Protocol for Heterogeneous WSNs). CRKPH manages the basic keys, the derived keys and the cluster-keys. A key pool is generated and each sensor node is preconfigured with specific keys with which it establishes secure links with its neighbors. The BS buildups a set of levels. Using the preloaded keys, the sensor nodes derive the corresponding key for each level. In order to establish a secure link with the CH, each sensor node sends to it the basic key identifier list and the level identifier. Then the CH responds by a key sub-pool for each sensor node. When a sensor node leaves its cluster, it averts the other member nodes. Before its departure, the CH sends to it a specific key which should be used when the node rejoins later the cluster.

In [22], Seo et al. have proposed CL-EKM (CertificateLess-Effective Key Management) for dynamic hierarchical WSNs. Before the network deployment, the BS establishes a pair of public and private keys for each sensor node. An individual sensor node key is shared with the BS by a Hash-based Message Authentication Code (HMAC). After the network deployment, each sensor node establishes pairwise-keys with its neighbors. A pairwise master-key is established and their respective certificate-less public/private key pairs. In the next step, the CH forms the cluster with its authenticated neighboring sensor nodes and asks the BS to their validation. A cluster-key is shared by all the sensor node members and is used to secure the broadcasted messages. The cluster-key is generated by the CH using HMAC (on a secret parameter and the identifier of the CH) and exchanged using the pairwise-keys. The keys update occurs when a sensor node moves between clusters. For pairwise-key update, it is not necessary to change the pairwise master-keys, just perform again the pairwise-key establishment. For the cluster-key update, the CH chooses a new secret parameter and computes again the HMAC.

3. System model

In this section, we present the network and attack models, and an overall view of the proposed solution.

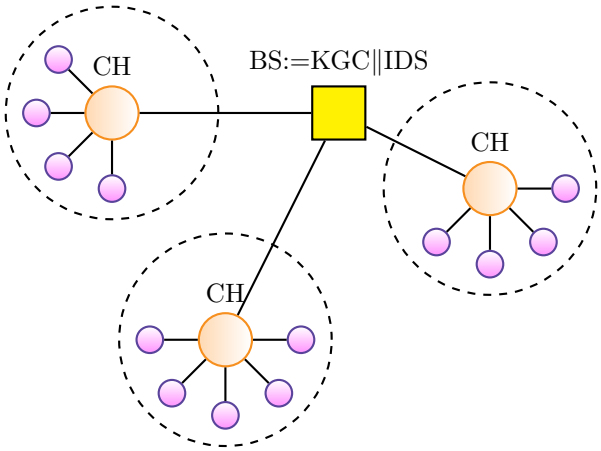

Figure 2: Network model

3.1. Network model

We consider a network composed of a set of wireless sensor nodes deployed on a hostile zone of interest, supervised by a single BS. The network follows a cluster-based architecture, where each cluster is supervised by a single CH, which could communicate directly with the BS without intermediate sensor nodes. The clustering is established between a CH and all the neighbor nodes, which are in one hope distance. We illustrate in Figure 2, the network model. We assume that the sensor nodes are homogeneous regarding the hardware characteristics, such as storage, battery power, sensing, processing, communication capacities, and is given for each one a unique identifier. The CHs are supposed to be more powerful in terms of resources compared to the other sensor nodes and are certified by the BS before the network deployment. The CHs are stationary, however the sensor nodes are mobile, where they could leave, rejoin their cluster, or join another one. We assume that the BS is sufficiently secured and has no constraints of storage and computation. A Key Generator Center (KGC) is installed on the BS, which generates the key pairs of the network nodes. We assume also that in the BS is installed an Intrusion Detection System (IDS), which supervise continually the network nodes behavior, and accordantly validate or invalidate the nodes. In order to limit the scope for the key management, which is quite large, we do not address the issue of intrusion detection. For more information about the research efforts on the latter topic, kindly refer to [23].

3.2. Attack model and security requirements

We suppose that the system doesn't include compromised sensor nodes at the initialization step and the attacks are performed after the network deployment. An attacker could be static or mobile and could perform active or passive attacks by considering internal and external attacks. An attacker could perform

an external attack by compromising a sensor node and/or extracting its secret cryptographic parameters. An internal attacker could decrypt and/or modify the encrypted messages with the keys that it holds. An internal attacker could also drop messages and overhear the communication between the sensor nodes being in its range.

The main objective of our framework is threefold, namely compromising resistance, cloning resistance, and past and future secrecy. In the fact that a sensor node be compromised, it must not affect the security of the stocked information in the other sensor nodes or reveal their keys. An adversary may launch an attack of cloning, by extracting the keys of a sensor node it compromised and installs them into a clone nodes, which injects in the network. The system must ensure the transmission of the secret key in order to counter a sensor node to use an old key in decrypting the new messages. It must also ensure the past secrecy in order to counter a sensor node with its new key to decrypt the message, which it eventually stocked in the past.

3.3. Overview of the proposed solution

One of the most challenging issues in WSNs is the secure communication establishment, which requires a sous-jacent key management system that should be efficient and robust. The key pre-distribution, rekeying and hybrid encryption are the main building-block of a such system. In this work, we propose a key management system based on hybrid encryption for dynamic WSNs. Our proposal manages two types of key, namely the key pair (public and private keys) and the cluster-keys. Before the network deployment, the BS generates an ECC-based key pair for each sensor node, which are then installed in its local memory. These keys are used to encrypt the communication between the sensor node and its corresponding CH or between the latter and the BS. All the sensor nodes belonging to a cluster share a cluster-key. This key is used to disseminate with a secure manner the alert messages in the cluster whenever a compromised node is detected. The CH of a cluster is the only node which updates this key when a sensor node joins or leaves the cluster. The proposed key management framework operates over five main operations, namely the system bootstrapping, the cluster formation, the key updating, the node movement and the revocation. In Table 1, we present the main used notations.

4. Efficient and energy-aware key management approach

In this section, we present the detailed description of the proposed solution.

Symbol	Signification
ϑ_i	Sensor node i's identity
CH_j	Cluster-head j's identity
Ω	System parameters
P_i	Sensor node i's public-key
S_i	Sensor node i's private-key
K_j	Cluster j's key
C	List of the legitimate network sensor nodes
C_j	List of the legitimate cluster j member nodes
R	List of the revoked sensor nodes
R_j	List of the revoked cluster j nodes
T_i	Timestamp generated by the entity i
DD_i	Departure date of the sensor node ϑ_i
$(M)_K$	Message M encrypted with the key K

Table 1: Notations

4.1. System bootstrapping

This phase is executed in offline before the network deployment. The sensor nodes are deployed after the initial private-key pre-distribution. The BS generates the system parameters, and then, the sensor nodes themselves generate their key pairs. The BS outputs the system parameters $\Omega = \langle E/F_q, F_q, P\rangle$, where q is a large prime number, E/F_q is an elliptic curve over q, F_q a set of values defined for the variables of the elliptic curve, and P is a point of the curve. Then, the BS publishes Ω for all the sensor nodes. Each sensor node ϑ_i chooses its private-key $S_i \in Z_q^*$ and computes the corresponding public-key $P_i = S_i \cdot P$, which is made available to the BS. With the same process, the CHs generate their key pairs. The BS generates a list of members C, where it registers the sensor nodes. This list contains the identifiers, public-keys and CHs which belong all the sensor nodes. It initializes a revocation list R that contains the revoked sensor nodes during the network life. The BS installs in each CH, a certificate giving the privilege of CH and the latter forms its cluster based on that.

4.2. Cluster formation

After the network deployment, each CH gets the identifiers of its neighboring sensor nodes using the balise messages which it diffuses. Then, it proceeds to their authentication through the BS. In this way, the CH_j diffuses the message $\langle P_j, CH_j, (CH_j, \text{"CH"})_{S_{BS}}\rangle$ in order to discover its neighbors and authenticates itself to them regarding the role of CH. The sensor node ϑ_i, upon receiving the several offerts, it selects the CH with the best signal intensity, and authenticates the CH's role and identity. If it holds, it sends a positive acknowledgment $\langle \text{"PACK"}, \vartheta_i, P_i, CH_j\rangle$, and a negative acknowledgment for the other offers. Finally, each CH_j constitutes its list C_j and sends it to the BS. Then, the BS verifies the identity of the CH_j and the validity of the sensor nodes belonging to the list C_j. If the CH_j is legitimate and all the

sensor nodes are valid, the BS responds with a positive acknowledgment to the CH_j. Otherwise, it forms a list R_j and sends it to the CH_j. Upon validating the list C_j, the CH_j generates the cluster-key K_j, then diffuses $\langle \text{"PACK"}, T_j, (T_j)_{S_j}, (K_j)_{P_1}, (K_j)_{P_2}, \cdots, (K_j)_{P_{|C_j|}} \rangle$ to the validated sensor nodes. Each sensor node $\vartheta_i \in C_j$ receiving this message, verifies the identity of the CH, and then decrypts the cluster-key with its public-key. Algorithm 1 gives in detail the different steps of the cluster formation. The asymptotic time complexity of this algorithm is of order $O(n)$ for the worst case.

Algorithm 1 Cluster formation

for each CH_j **do**
  CH_j diffuses to the neighboring nodes $\langle P_j, T_j, (T_j, \text{"CH"})_{S_{BS}} \rangle$;
  for each ϑ_i **do**
    ϑ_i selects the nearest cluster-head $\overline{CH}$ regarding the signal intensity;
    ϑ_i sends to $\overline{CH}$ a positive acknowledgment $\langle \text{"PACK"}, \vartheta_i, P_i, \overline{CH} \rangle$;
    ϑ_i sends to the other CH_j a negative acknowledgment $\langle \text{"NACK"}, \vartheta_i, CH_j \rangle$;
  end for
  CH_j constitutes the cluster member list C_j;
  CH_j sends to the BS: $\langle T_j, (T_j, C_j)_{S_j} \rangle$;
  $R_j \leftarrow \varnothing$;
  for each $\vartheta_i \in C_j$ **do**
    BS verifies the legitimacy of ϑ_i;
    if ϑ_i is not valid **then**
      $C_j \leftarrow C_j - \{\vartheta_i\}$;
      $R_j \leftarrow R_j \cup \{\vartheta_i\}$;
      $R \leftarrow R \cup R_j$;
      $C \leftarrow C - R_j$;
    end if
  end for
  BS sends to CH_j: $\langle (C_j, R_j)_{P_j} \rangle$;
  CH_j diffuses a negative acknowledgment $\langle \text{"NACK"}, CH_j \rangle$ to $\vartheta_i \in R_j$;
  CH_j generates the cluster-key K_j;
  CH_j diffuses positive acknowledgment $\langle \text{"PACK"}, T_j, (T_j)_{S_j}, (K_j)_{P_1}, (K_j)_{P_2}, \cdots, (K_j)_{P_{|C_j|}} \rangle$ to $\vartheta_i \in C_j$;
  for each $\vartheta_i \in C_j$ **do**
    ϑ_i authenticates CH_j using $\langle T_j, (T_j)_{S_j} \rangle$;
    if CH_j is not valid **then**
      ϑ_i alerts the BS;
    else
      ϑ_i decrypts its part $(K_j)_{P_i}$ and saves the cluster-key K_j;
    end if
  end for
end for

4.3. Nodes departure and joining

The sensor node movement phase can happen in several cases. A sensor node can leave voluntarily (proactive departure) its cluster, leave involuntarily (reactive departure) its cluster, joins its old cluster or a new one. A sensor node can leave its cluster in case of failure, physical displacement or in case of

communication failure with the CH. In the proactive case, a sensor node ϑ_i leaves actively its cluster. For example, if the power of the CH_j's signal is weakened when it moves away from CH_j. At this moment, the sensor node ϑ_i marks its departure date DD_i and sends to the CH_j: $\langle DD_i, \vartheta_i, (DD_i, \vartheta_i)_{S_i} \rangle$. Upon receiving the message, the CH_j saves $(DD_i, \vartheta_i)_{S_i}$, which will be used later to authenticate the sensor node in question when it comes back to the cluster. The CH_j notifies then the BS in order to change the status of the leaving sensor node. The BS responds with an acknowledgment, then the CH_j removes ϑ_i from the list C_j. In the reactive case, a sensor node ϑ_i leaves suddenly the cluster without preventing the CH. This may happen in the depletion of the battery, and could be detected by the CH through the beacon messages that the latter exchanges periodically with the cluster members. In this case, the CH_j alerts the BS, which revokes the sensor node in question, and finally updates the cluster-key K_j. When a sensor node displaces, it may rejoin an old cluster to which was a member in the past or join a new one. In the case of a sensor node ϑ_i coming back to its old cluster, it sends to the CH_j: $\langle \text{"Rejoin\_request"}, (DD_i, \vartheta_i)_{S_i} \rangle$. Upon receiving the message, the CH_j decrypts it using the ϑ_i's public-key and verifies if the content of the decrypted message corresponds to the already stored one. If it holds, the authentication succeeds and then the CH_j requests the BS in order to validate the sensor node. If it holds, the BS sends a positive acknowledgment to the corresponding CH_j and, finally, the sensor node rejoins the cluster. Otherwise, the CH_j ignores the sensor node request. In the case of a sensor node ϑ_i joining a new cluster, it sends to the corresponding CH_j the message $\langle \text{"Join\_request"}, \vartheta_i, P_i \rangle$. Upon receiving the message, the CH_j requests the BS in order to authenticate and validate the new sensor node. If it holds, it responds with a positive acknowledgment to the CH_j and the sensor node ϑ_i can then join the cluster. Finally, the CH_j proceeds to the update of the cluster-key. Algorithm 2 gives in detail the different steps of the nodes departure and joining operations. The asymptotic time complexity of this algorithm is of order $O(n)$ for the worst case.

4.4. Revocation and key updating

A compromised sensor node can be detected through the IDS and its compromising mustn't affect the cluster-key. If its key pair is compromised, this does no affect the other sensor nodes. However, if the cluster-key is compromised, this affects all the other sensor nodes belonging to the same cluster. In this case, the cluster-key must be updated. In order to guarantee the past and present secrecy, the rekeying will be performed for two cases, namely when a sensor node is compromised and at each cluster configuration changing (e.g., a sensor node joins or leaves the cluster). The rekeying process is the same in the two cases.

Algorithm 2 Nodes joining and departure

if proactive departure of ϑ_i **then**
  ϑ_i sends $\langle DD_i, \vartheta_i, (DD_i, \vartheta_i)_{S_i} \rangle$ to CH_j;
  CH_j saves $\langle (DD_i, \vartheta_i)_{S_i} \rangle$;
  CH_j notifies the BS;
else
  if reactive departure of ϑ_i **then**
    CH_j alerts the BS;
    BS revokes ϑ_i ($R_j \leftarrow R_j \cup \{\vartheta_i\}$ and $R \leftarrow R \cup \{\vartheta_i\}$);
  else
    if joining of ϑ_i **then**
      if the case of an old cluster **then**
        ϑ_i sends $\langle$"Rejoin_request", $(DD_i, \vartheta_i)_{S_i} \rangle$ to CH_j;
        CH_j verifies $\langle (DD_i, \vartheta_i)_{S_i} \rangle$ and ignores the request if the node is unregistered;
      else
        ϑ_i sends $\langle$"Join_request", $\vartheta_i, P_i \rangle$ to CH_j;
      end if
      CH_j requests the BS to authenticate ϑ_i;
      BS verifies the legitimacy of ϑ_i;
      if ϑ_i is not valid **then**
        BS sends to CH_j a negative acknowledgment;
        $R_j \leftarrow R_j \cup \{\vartheta_i\}$;
        $R \leftarrow R \cup R_j$;
        CH_j ignores the ϑ_i's request;
      else
        $C_j \leftarrow C_j \cup \{\vartheta_i\}$;
        $C \leftarrow C \cup \{\vartheta_i\}$;
        BS sends to CH_j a positive acknowledgment with the list C_j;
      end if
    end if
  end if
end if
CH_j updates the list C_j;
CH_j generates a new cluster-key K_j^*;
CH_j diffuses a $\langle$"NKEY", $T_j, (T_j)_{S_j}, (K_j^*)_{P_1}, (K_j^*)_{P_2}, \cdots, (K_j^*)_{P_{|C_j|}} \rangle$ to $\vartheta_i \in C_j$;
for each $\vartheta_i \in C_j$ **do**
  ϑ_i authenticates CH_j using $\langle T_j, (T_j)_{S_j} \rangle$;
  if CH_j is not valid **then**
    ϑ_i alerts the BS;
  else
    ϑ_i decrypts its part $(K_j^*)_{P_i}$ and saves the cluster-key K_j^*;
  end if
end for

If a sensor node ϑ_k is compromised, the BS extracts from the list C, the CH_j supervising ϑ_k and alerts it by sending the message $\langle$"Compromised_node", $(\vartheta_k)_{P_j}\rangle$. Then, the CH_j removes the stored ϑ_k's public-key and launches the rekeying process of the cluster-key. In this context, the CH_j encrypts a fake key $(\overline{K}_j)_{P_k}$ in order to disconcert the compromised sensor node and do not establish communication links with it. If the BS detects the CH_j be compromised, it alerts the sensor nodes of the cluster supervised by that CH through the message $\langle$"Compromised_CH", $(CH_j)_{P_i}\rangle$. Finally, each sensor node removes the stored CH_j's public-key, the cluster-key, and tries to join another cluster. Algorithm 3 gives in detail the different steps of the nodes revocation. The asymptotic time complexity of this algorithm is of order $O(n)$ for the worst case.

Algorithm 3 Nodes revocation

1: **if** BS detects a compromised node ϑ_k **then**
2: BS sends $\langle$"Compromised_node", $(\vartheta_k)_{P_j}\rangle$ to CH_j;
3: CH_j generates a new cluster-key K_j^*;
4: CH_j generates a fake cluster-key $\overline{K}_j$;
5: CH_j diffuses $\langle$"NKEY", $(K_j^*)_{P_1}, \cdots, (\overline{K}_j)_{P_k}, \cdots, (K_j^*)_{P_{|C_j|}}\rangle$ to all the cluster members;
6: **end if**
7: **if** BS detects a compromised CH_j **then**
8: BS diffuses $\langle$"Compromised_CH", $(CH_j)_{P_i}\rangle$ to each cluster member ϑ_i;
9: Each $\vartheta_i \in C_j$ removes the CH_j's public-key;
10: Each $\vartheta_i \in C_j$ removes the cluster-key K_j;
11: Each $\vartheta_i \in C_j$ joins another CH;
12: **end if**

We suppose an adversary, which captures a sensor node belonging to a given cluster. This adversary may extract all the secret sensor node keys, namely its private-key and the cluster-key. However, the sensor node key pairs are independent and unique. Getting access to the sensor node private-key has no impact on the other node keys. Moreover, the cluster-key is unique and independent to the other cluster-keys. Compromising this key has no impact on the other clusters. The cluster in question will be compromised only in a certain window of vulnerability until the compromised node be detected and revoked by the IDS. In addition, the network mobility reduces the window of vulnerability, where the cluster rekeys at each joining or departure of a sensor node. Hence, the cluster will regain rapidly its correct state after updating the cluster-key.

5. Performance evaluation

In this section, we evaluate the performances of the proposed scheme by simulations. We compare it with other protocols to underline the efficiency and the suitability of our solution.

5.1. Simulation parameters

The simulations are developed using Matlab programming environment [1]. We consider a randomly deployed network of a set of sensor nodes in an area of $1km^2$. The CH number represents 25% of the network size. The sensor nodes have the same hardware characteristics and are equipped with wireless communication interfaces with the same power of communication range. The simulator considers whether a radio link exists between any pair of sensor nodes according to the distance which separates them. We suppose that the data is with a size of 2 kbit. We use the energy consumption model of wireless communication hardware proposed in [24]. If the sensor node transmits a k-bit packet over a distance d, it consumes in Joule $k \cdot (\alpha + \beta \cdot d^2)$, where α denotes the electrical energy, which represents the energy per bit consumed by the transmitter electronics, and β denotes the empirical energy, which represents the energy dissipated in the transmission amplifier. When receiving a k-bit packet, it consumes in Joule $k \cdot \alpha$. In the simulations, the empirical energy is set to 10 nJoule and the electrical energy to 50 nJoule. For processing, we have adopted the computational energy model developed by Meulenaer et al. [25]. In the latter, a practical estimation of the energy cost of cryptographic operations is performed in case of both symmetrical and asymmetrical encryptions. They have developed experimentations for two systems, namely, AES-128 in case of symmetrical encryption and ECDSA-160 in case of asymmetrical encryption. In our simulations, we have considered a processing capacity of 4 MHz, and as resulted in [25], a sensor node consumes 40 mJoule, 9 µJoule and 19 mJoule for hash operation, symmetrical and asymmetrical encryption, respectively.

The performances of our protocol are compared to the protocols CL-EKM [22] and CRKPH [21], which are described in Section 2. Unlike the nature of the other classes of solutions, namely "static key distribution protocols" and "dynamic key distribution protocols for static WSNs", the proposed solution belongs to the category of "dynamic key distribution protocols for dynamic WSNs", in which the protocols CL-EKM and CRKPH are the most representative solutions for comparison. The evaluated criteria are the storage overhead and the energy consumption in both communication and computation. The storage overhead represents the number of memory units used in order to save the identifiers and the required keys by the sensor nodes. The energy consumption is due to the communication and computation. The energy consumption in terms of communication is estimated regarding the exchanged messages among the sensor nodes. The energy consumption in terms of computation is estimated regarding each individual cryptographic operation performed by the sensor nodes. The impacts studied are the network size, the

[1] Matlab does not include predisposed modules for network performance evaluation. In this part of the work, Matlab is used as a programming language tool with which is developed the simulator and the compared protocols.

network partitioning and the topology change degree. The variation of the network size allows evaluating the scalability of the compared schemes. This metric has been varied between 50 and 1000 sensor nodes. The network partitioning has been evaluated by varying the sensor node communication range in order to evaluate the impact of the availability and network mobility on the performances of the compared schemes. This metric is varied between 20m and 200m. The variation of the topology change degree allows evaluating the impact of the node mobility intensity on the performances of the compared schemes. The frequency of topology change has been varied between 20s and 60s.

We have not provided a real implementation and/or suite layers development. We believe that the simulation study is more appropriate regarding the evaluated metrics (scalability, mobility frequency, etc.). The environment setting would be hard to realize by real experiments when evaluating such metrics, due to the high size of network and the high number of randomly network deployment topologies. The compared key management systems are operating at the application layer regarding the protocol stack. For this purpose, all the security messages that are exchanged between the sensor nodes and the BS, and between the sensor nodes themselves are generated by the application layer. These messages can be encapsulated by any communication standard. The choice of the communication standard (TCP/IP, ZigBee, etc.) has no impact on the compared schemes functioning. Thus, in the developed simulator, an abstraction is made about the lower layers, where no exigence is posed for specific protocols in the network, data link and physical layers. In this context, each message generated by the application layer is considered as a message transmitted by the sensor node. Over the network topology view, the simulator computes the shortest routing paths connecting the communicating sensor nodes. Then, the simulator estimates the energy consumption of the participating intermediate sensor nodes in the exchange. Likewise, the application layer is independent to the MAC layer, and the simulator does not implement the MAC layer protocols. We summarize in Table 2 the simulation parameters and in what follows, we discuss the obtained results.

5.2. Scalability

Figure 3 plots the storage overhead of the compared schemes in function of the network size. Regarding the obtained results, for the concurrent schemes (especially for CRKPH), when the sensor node number increases, the storage cost increases considerably compared to our scheme. This is due to the fact that our scheme uses the key pairs to secure the communication. However, the other protocols use other types of key, such as the pairwise-keys in CL-EKM and the base keys in CRKPH. Figure 4 plots the energy consumption in terms of communication of the compared schemes in function of the network size.

Parameter	Value
Area of deployment	1km^2
Network size	50–1000
CH number	25%
Communication range	20m–200m
Data size	2kbits
Frequency of topology change	20s–60s
Empirical energy	10 nJoule
Electrical energy	50 nJoule
Symmetrical encryption energy	9 µJoule
Asymmetrical encryption energy	19 mJoule
Hash operation energy	40 mJoule

Table 2: Parameters of simulation

Regarding the obtained results, for the concurrent schemes, when the sensor node number increases, the energy consumption increases considerably compared to our scheme. This result is due to the lightweight load of exchanged messages in case of the proposed scheme compared to the others. Figure 5 plots the total energy consumption of the compared schemes in function of the network size. Again, the obtained results demonstrate that the gain in terms communication overhead has covered the cost of computation. We note that in the protocols CL-EKM and CRKPH, although the cost of computation is quite reduced, the energy overhead generated by the communication is high compared to our scheme.

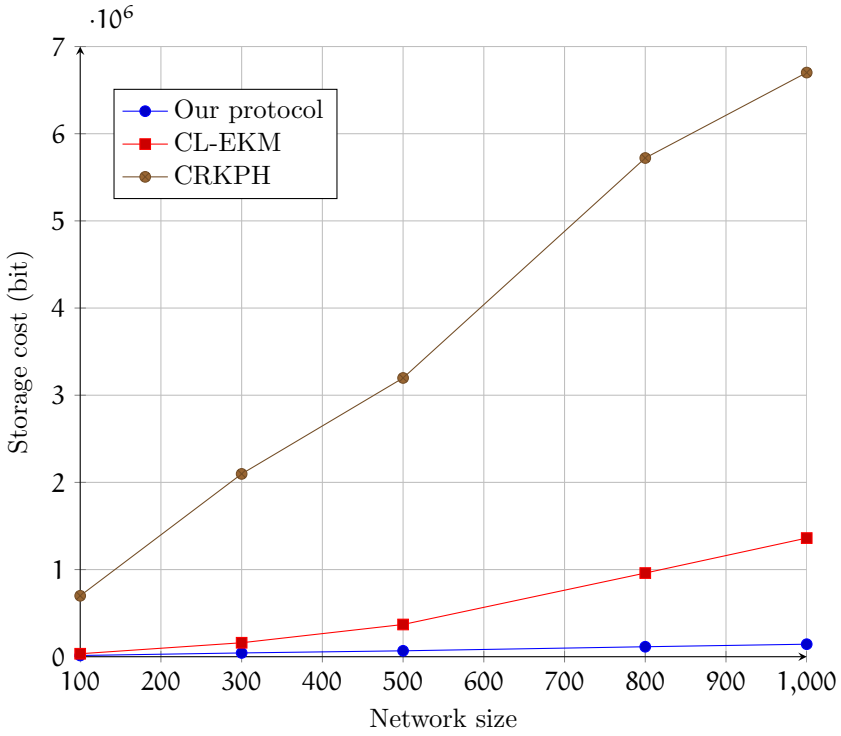

Figure 3: Storage in function of the network size

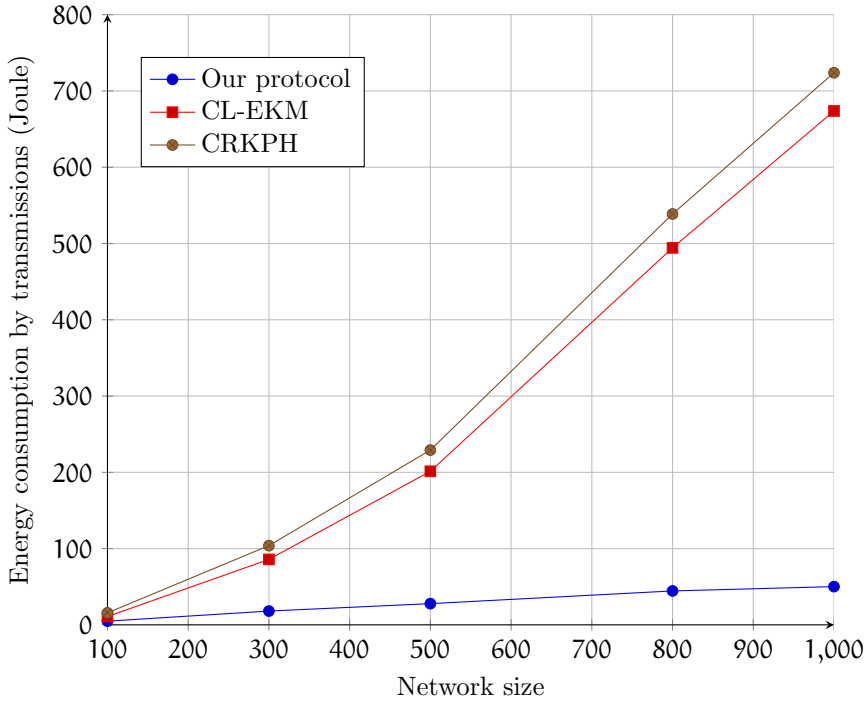

Figure 4: Energy consumption by transmissions in function of the network size

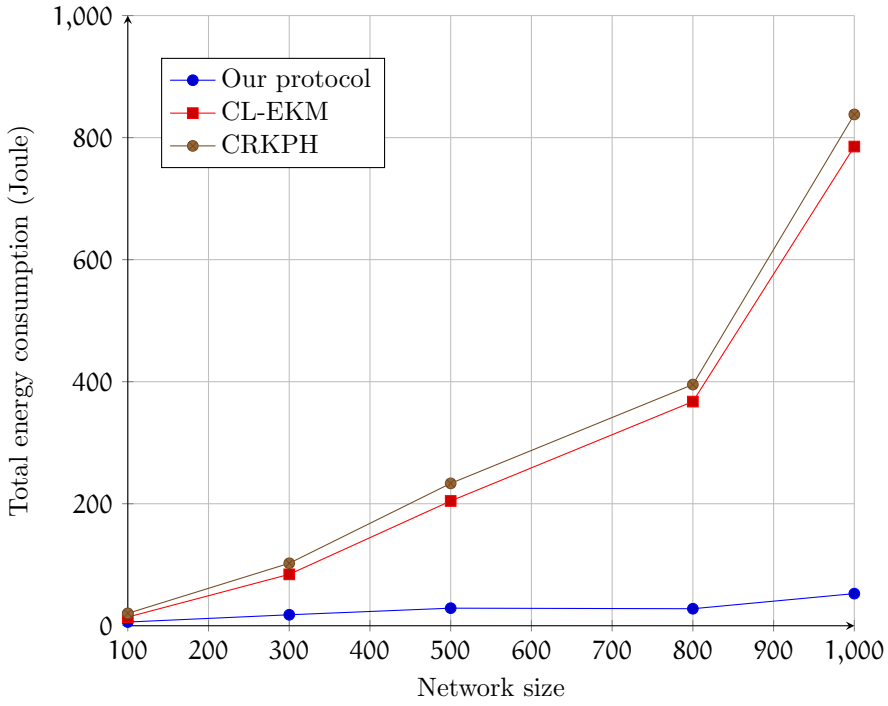

Figure 5: Total energy consumption in function of the network size

5.3. Impact of network partitioning

Figure 6 plots the storage overhead of the compared schemes in function of the sensor node communication range. Regarding the obtained results, in the protocol CL-EKM, more the communication range increases, more the sensor nodes have an important number of neighboring nodes, and hence more keys to store. In the protocol CRKPH, more the communication range increases, less are the isolated sensor nodes, where each one holds 20 basis keys and the corresponding derived keys. However, in our scheme, the communication range augmentation reduces the isolated sensor node number, where each one holds 4 keys. Figure 7 plots the energy consumption in terms of communication of the compared schemes in function of the sensor node communication range. Regarding the obtained results, until 50m of communication range, the compared protocols are equal in terms of energy consumption. Indeed, with a communication range of 50m regarding the considered deployment surface provides a relatively high number of isolated sensor nodes. The latter nodes haven't the ability to communicate with the CH, which provides a low degree of energy consumption. However, beyond 50m, the energy consumption following the protocols CL-EKM and CRKPH increases significantly due to the high load of exchanged messages in order to establish all the required keys. Figure 8 plots the total consumed energy of the compared schemes in function of the sensor node communication range. We constate that the obtained results approximate the results illustrated in Figure 7. This is due to the energy generated by the computation overhead, which is negligible compared to the energy consumption in terms of communication.

5.4. Mobility

Figure 9 plots the storage overhead of the compared schemes in function of the network topology change frequency. Regarding the obtained results, the storage cost following the protocol CRKPH is high, which increases when the intensity of mobility increases. This is due to the high number of keys which holds, where it frequently leaves and joins the clusters. In the other protocols, the storage overhead is less initially, but later the increase is considerable in the case of the protocol CL-EKM. This is due to the established shared keys at each topology change. Our scheme provides the best results, where the key number is limited regardless of the intensity of sensor nodes mobility. Figure 11 plots the energy consumption in terms of communication of the compared schemes in function of the topology change frequency. Compared to our scheme, in the protocols CL-EKM and CRKPH, more the mobility is intense, more the energy consumption increases considerably. This is due to the high load of the exchanged messages in order to renew the cluster-keys and to establish the shared keys between the neighboring sensor nodes. Figure 11 plots the total energy consumption of the compared schemes in

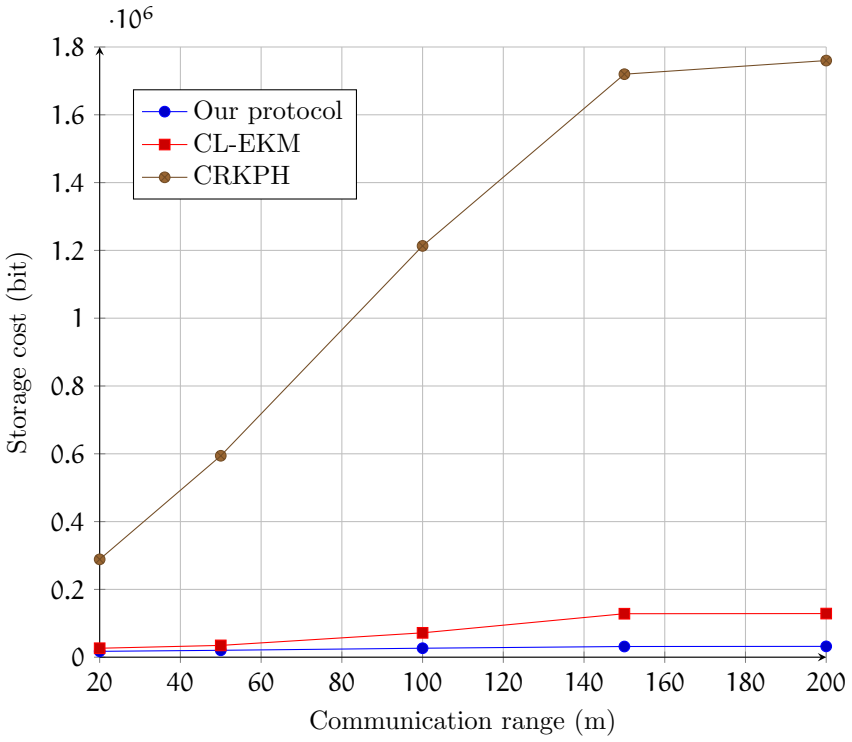

Figure 6: Storage in function of the communication range

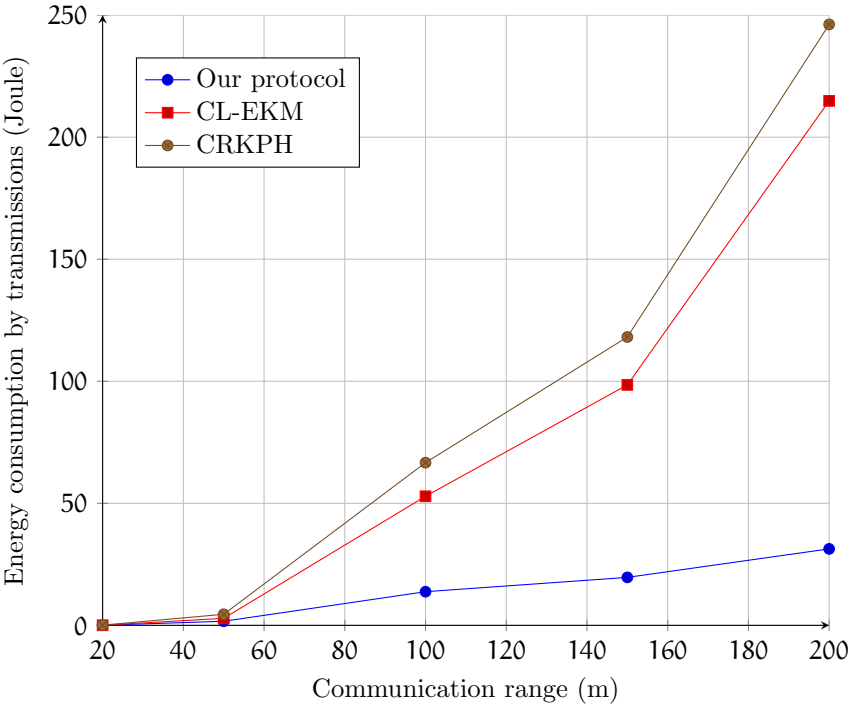

Figure 7: Energy consumption by transmissions in function of the communication range

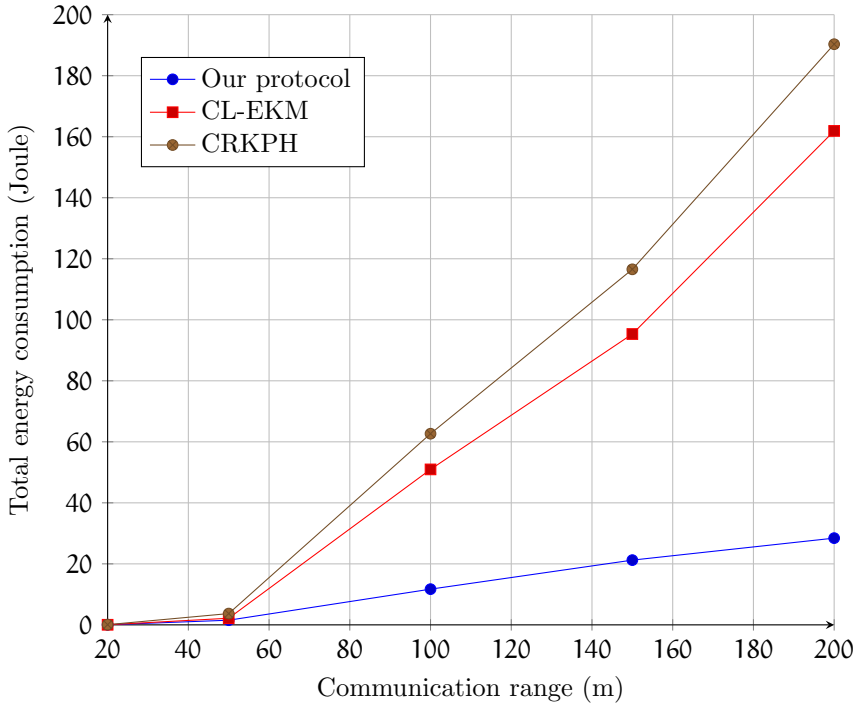

Figure 8: Total energy consumption in function of the communication range

function of the topology change frequency, where the results approximate the results illustrated in Figure 10.

6. Conclusion

The wide-ranging spectrum of wireless sensor networks is very observable in nowadays applications, especially in the emergent communication technologies such as the Internet of things and smart cities. We have investigated one of its most vital functionalities, namely the key management. The latter is a primordial service for any security operation. Without establishing and distributing keys, the communicating sensor nodes cannot guarantee the other security services, such as authentication, confidentiality, integrity, non-repudiation, etc. We have first reviewed the literature in the light of the recent and relevant key management systems proposed in the framework of wireless sensor networks, and we have classified them with respect of the mobility criterion. Afterwards, we have proposed a new key management scheme, which deals with the mobility issue in such networks. The proposed key management scheme operates over five main components, namely the system bootstrapping, cluster formation, key updating, node movement and revocation. With the aim to check the performances of the proposed scheme, we have conducted intensive simulations. We have compared the proposed scheme to concurrent ones from

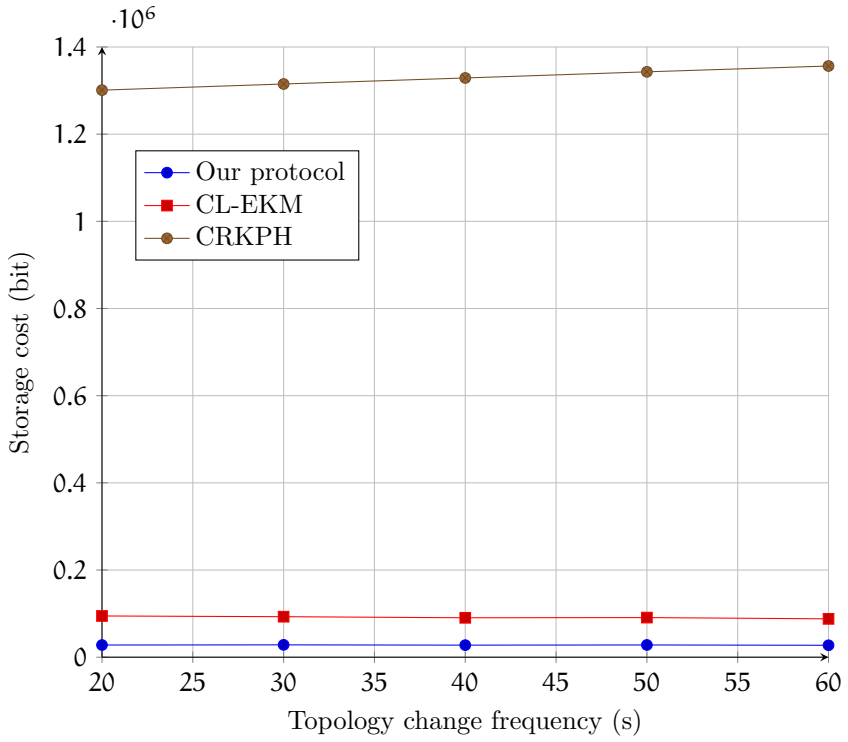

Figure 9: Storage in function of the topology change frequency

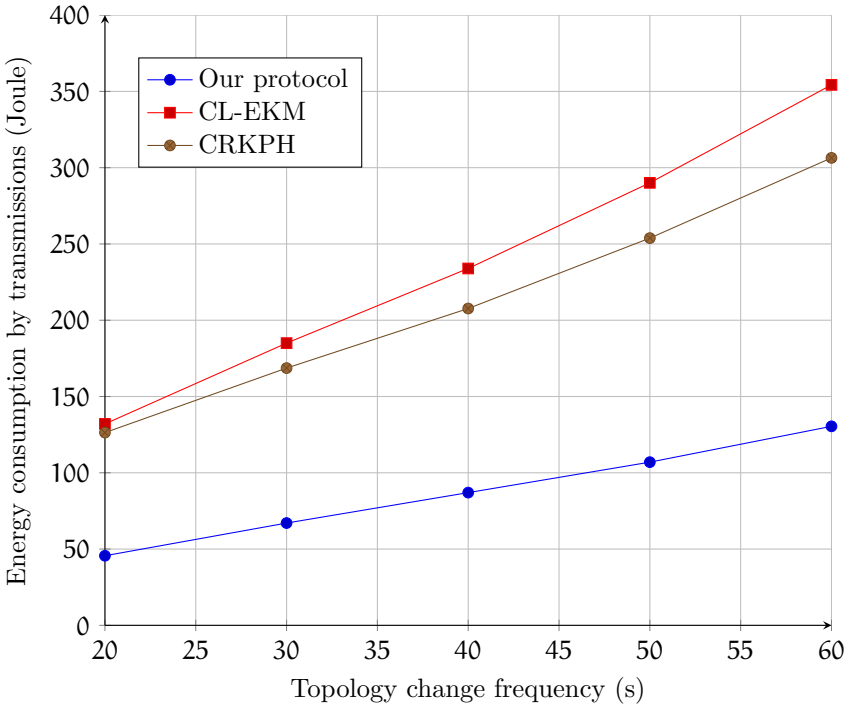

Figure 10: Energy consumption by transmissions in function of the topology change frequency

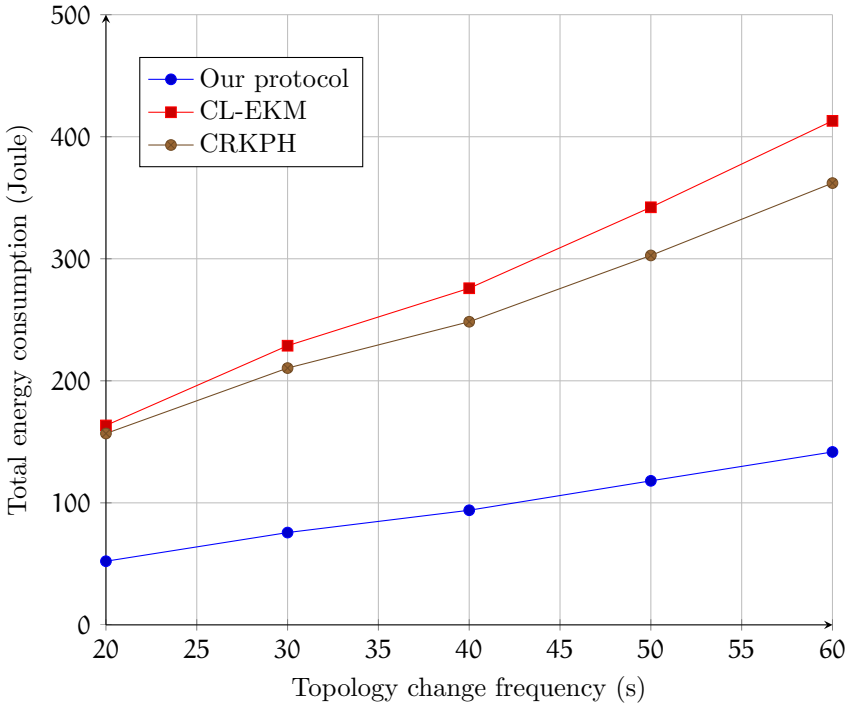

Figure 11: Total energy consumption in function of the topology change frequency

the literature with respect to the network partitioning, the topology change degree and the scalability. Under these inherent settings, the proposed scheme obtains the best results. It optimizes the resource usage regarding the important constraints of such network, while resisting against the compromised and cloned sensor nodes, and preserving the past and future secrecy.

The present work opens up for several medium and long-term perspectives. One of the aspects that can improve the proposed scheme is the integration of an access control service. This will allow the management of the cohabitation of several heterogeneous networks and the integration of other services. Another perspective that we consider important is the implementation of the proposed scheme on a real platform of sensor network.

Acknowledgment

This work was carried out in the framework of the research activities of the laboratory LIMED, which is affiliated to the Faculty of Exact Sciences of the University of Bejaia. The authors are very grateful and greatly thank the editors and anonymous reviewers for the time and effort they devote to the examination of our work. The authors thank also Akilal Abdellah for its technical assistance.

References

[1] D-L. Curiac. Towards wireless sensor, actuator and robot networks: Conceptual framework, challenges and perspectives. Journal of Network and Computer Applications, Vol. 63, Pages 14–23, 2016.

[2] A-J. AL-Mousawi, H-K. AL-Hassani. A survey in wireless sensor network for explosives detection. Computers & Electrical Engineering, DOI: 10.1016/j.compeleceng.2017.11.013, 2017.

[3] W-H. Nam, T. Kim, E-M. Hong, J-Y. Choi, J-T. Kim. A Wireless Sensor Network (WSN) application for irrigation facilities management based on Information and Communication Technologies (ICTs). Computers and Electronics in Agriculture, Vol. 143, Pages 185–192, 2017.

[4] M-F. Othman, K. Shazali. Wireless Sensor Network Applications: A Study in Environment Monitoring System. Procedia Engineering, Vol. 41, Pages 1204–1210, 2012.

[5] O. Cheikhrouhou. Secure Group Communication in Wireless Sensor Networks: A survey. Journal of Network and Computer Applications, Vol. 61, Pages 115–132, 2016.

[6] G-S. Oreku, T. Pazynyuk. Security in Wireless Sensor Networks. Chapter Book: Introduction and Overview. Part of the series Risk Engineering, Springer, Pages 1–23, 2015.

[7] M-L. Messai, H. Seba. A survey of key management schemes in multi-phase wireless sensor networks. Journal of Computer Networks, Vol. 105, Pages 60–74, 2016.

[8] Nisha and M. Dave. Storage as a parameter for classifying dynamic key management schemes proposed for WSNs. Proceedings of the International Conference on Computational Techniques in Information and Communication Technologies, Pages 51–56, India, 2016.

[9] X. He, M. Niedermeier, H. de-Meer. Dynamic key management in wireless sensor networks: A survey. Journal of Network and Computer Applications, Vol. 36, Num. 2, Pages 611–622, 2013.

[10] J. Zhang, V. Varadharajan. Wireless sensor network key management survey and taxonomy. Journal of Network and Computer Applications, Vol. 33, Num. 2, Pages 63–75, 2010.

[11] S. Zhu, S. Setia, S. Jajodia. LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks. Proceedings of the 10th ACM conference on Computer and communications security, Pages 62–72, 2003.

[12] H. Chan, A. Perrig, D. Song. Random key predistribution schemes for sensor networks. Proceedings of the symposium on security and privacy, Pages 197–213, 2003.

[13] M. Kun, L. Li. An Efficient Pairwise Key Predistribution Scheme for Wireless Sensor Networks. Journal of Networks, Vol. 9, Num. 2, Pages 277–282, 2014.

[14] P. Hazra, D. Giri, A-K. Das. Key Chain-Based Key Predistribution Protocols for Securing Wireless Sensor Networks. The series Springer Proceedings in Mathematics and Statistics, Vol. 139, Pages 135–154, 2015.

[15] A. Perrig, R. Szewczyk, J-D. Tygar, V. Wen, D-E. Culler. SPINS: security protocols for sensor networks. Journal of Wireless Networks, Vol. 8, Num. 5, Pages 521–534, 2002.

[16] H. Chan, A. Perrig. PIKE: peer intermediaries for key establishment in sensor networks. Proceedings of the 24th annual joint conference of the IEEE computer and communications societies, Pages 524–535, 2005.

[17] R. Vaid, V. Katiyar. VLKM: Virtual Location-Based Key Management Scheme for Wireless Sensor Networks. Proceedings of the International Conference on Parallel, Distributed and Grid Computing, Pages 53–61, 2014.

[18] Y. Zhang, J. Pengfei. An Efficient and Hybrid Key Management for Heterogeneous Wireless Sensor Networks. Proceedings of the 26th Chinese Control and Decision Conference, Pages 1881–1885, 2014.

[19] N. Suganthi, S. Vembu. Energy Efficient Key Management Scheme for Wireless Sensor Networks. International Journal of Computers Communications and Control, Vol. 9, Num. 1, Pages 71–78, 2014.

[20] X. Zhang, J. He, Q. Wei. EDDK: Energy-Efficient Distributed Deterministic Key Management for Wireless Sensor Networks. EURASIP Journal on Wireless Communications and Networking, Article Num. 12, 2010.

[21] S. Banihashemian, A-G. Bafghi, M-H. Yaghmaee Moghaddam. Centralized Key Management Scheme in Wireless Sensor Networks. Wireless Personal Communications, Vol. 60, Pages 463–474, 2011.

[22] S-H. Seo, J. Won, S. Sultana, E. Bertino. Effective key management in dynamic wireless sensor networks. IEEE Transactions on Information Forensics and Security, Vol. 10, Num. 2, Pages 371–383, 2015.

[23] R. Mitchell, I-R. Chen. A survey of intrusion detection in wireless network applications. Computer Communications, Vol. 42, Pages 1–23, 2014.

[24] W. Heizelman, A. Chandrakasan, H. Balakrishnan. Energy-efficient communication protocol for wireless sensor networks. Proceedings of the Hawaii international conference on systems, 2000.

[25] G. de-Meulenaer, F. Gosset, F-X. Standaert, O. Pereira. On the energy cost of communication and cryptography in wireless sensor networks. Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Pages 580–585, 2008.

Vitae

Mawloud Omar is assistant professor in the university of Bejaia (Algeria), where he is a member of the laboratory LIMED in the "Networking for Healthcare Applications" group. He got his PhD and Magister degrees in computer science from the university of Bejaia in 2011 and 2007, respectively. He got his engineer diploma in computer science from the university of Chlef (Algeria) in 2004.

Imene Belalouache received the master degree in computer science from the university of Bejaia (Algeria) in 2016. Her research report as a master student concerns security and energy consumption in wireless sensor networks. She got her licence diploma from the university of Bejaia in 2014.

Samia Amrane got her master degree in computer science in 2016 from the university of Bejaia (Algeria). Her research interests concern security in wireless sensor networks. She graduated at the university of Bejaia and got the diploma of licence in 2014.

Bournane Abbache is assistant professor in the university of Bejaia (Algeria). He received the Engineering degree and the Magister degree in computer science from the university of Bejaia (Algeria), in 2006 and 2010 respectively, where he is currently working toward the Ph.D. degree. He is a member of the laboratory LIMED in the university of Bejaia.