View metadata, citation and similar papers at core.ac.uk
brought to you by CORE

POWERFUL MOBILE NODES FOR ENHANCING WIRELESS SENSOR NETWORKS' SECURITY AND LIFETIME

Med Saïd Salah[1*] – Abderrahim Maizate[2] – Mohamed Ouzzif[1] – Mohamed Toumi[1]

[1]RITM-ESTC/CED-ENSEM, University Hassan II, Casablanca - Morocco
[2]STIC Laboratory, Chouaib Doukkali University, El Jadida - Morocco

ARTICLE INFO

***Article history*:**
Received: 06.11.2016.
Received in revised form: 10.07.2017.
Accepted: 11.07.2017.

Keywords:
Wireless sensor networks
Mobile nodes
Key management
Elliptic curve cryptography

DOI: *http://doi.org/10.30765/er.39.1.7*

Abstract:

To maintain the proper functioning of critical applications based on Wireless Sensor Networks, we must provide an acceptable level of security while taking into account limited capabilities of the sensors. In this paper we proposed a mobile approach to secure data exchanged by structured nodes in cluster. The approach is based on mobile nodes with significant calculation and energy resources that allow cryptographic key management and periodic rekeying. However, mobility in wireless sensor networks aims to increase the security and lifetime of the entire network. The technical methods used in this paper are based on cryptography elliptic curves and key management through a balanced binary tree. To compare the performance of the proposed approach with other mobile algorithms, we focused on the following metrics: the energy consumed by normal sensors and cluster heads, the number of packets exchanged during key installation, time to generate and distribute cryptographic keys, and the memory used by the different sensors to store keys.

1 Introduction

A wireless sensor network (WSN) is composed of a set of nodes sensors. Each of these nodes has the capacity to collect and transfer data to a base station connected to the user.

Security plays a central role in critical applications of the Wireless sensor network [1] such as smart cities applications, public lighting management, military applications, security solutions, etc. During normal operation of the network, data can be threatened by external events- something that should not happen. Security insurance [2]-[3] is a serious challenge, especially when nodes are made up of electronic devices [4] with limited material capacities.

Two key management approaches are available: symmetric and asymmetric approach [5]-[6]. The first focuses on the effectiveness of establishing cryptographic keys after network deployment. However, asymmetric cryptography offers better resistance against compromise nodes, but requires additional costs on the software and hardware nodes. Recently, experts in cryptography have made use of techniques based on elliptic curves [7] that can replace integer calculations by the calculations in

* Corresponding author. Tel.: +212 666 37 43 30 fax: +212 522 25 22 45
E-mail address: salahmedsaid@gmail.com

groups associated with elliptic curve, which turns the cryptography based on ECC stronger.
Some approaches based on the elliptic curve cryptography [8]-[9] have been proposed by the scientific community, yet the limited capacity of captors considerably slow down the life span of the entire network and does not allow any complex calculations [10]. To maintain the proper functioning of critical applications, the key management and rekeying must be periodic after network deployment. Each node must have all necessary keys to communicate with neighbors or with nodes in the same cluster.
The remainder of this paper is organized as following: in section II, we presented applications security in Casablanca Smart City. Then, we explained how mobility may affect the safety and life of the WSN. In section III, we presented some algorithms used mobility in WSN. In section IV, we discussed the technical methods used in this paper, especially the cryptography based on ECC and the AVL tree [11]. Thereafter, we described the proposed approach and its added value by algorithms and explanations of each phase. In the last section, we analyzed the different algorithms and compared the performance of each approach [12]- [13] by looking at: the average energy consumed per node, the memory used by node for storing ECC keys, the number of packets exchanged when installing keys and ECC keys computing time and rekeying.

2 Applications security in Casablanca smart city

The Casablanca has launched the study to develop a master plan for digital processing and information systems. This study has several phases, including analysis, up growth of digital development plan, as well as the pane of intelligent applications. Then it launched concrete projects and started their achievements.
Profits of applications based on the wireless sensor networks [14] may enhance the management of road traffic; inter-modality between different means of transport, through the traceability of cleaning operations, energy efficiency, etc. The objective is to put the applications for the benefit of the citizens [15]. However, attacks can have negative impacts on all applications in the city, thus minimizing security in those solutions. So, it is important to secure these applications in order to maintain their effectiveness.
The applications will be used daily by the citizens of the city and will help to manage the city and save energy resources much better [16]. In order to secure these applications, we proposed nodes attached to the 6 existing tramway lines shown in Fig. 3, to ensure coverage for the entire city (a diameter of 42 Km). Each line contains 8 tramways with a 10 minutes passage frequency. Hence, the rekeying frequency remains an organizational choice and depends on the risks to which the sensors are exposed.
Mobile nodes collaborate with each other to ensure full coverage of the city and the rekeying of all applications. Each mobile node, by communicating with a cluster head, asks whether the rekeying period has expired. If the period has expired the mobile node starts rekeying, if not the mobile node moves to the next cluster.

3 Algorithms using mobility in the WSN

3.1 A mobile agent approach for sensor networks

Cai, Chen, Hara, Shu and Kwon [17] proposed a multi-agent route planning algorithm based on the GA-MIP (genetic algorithm based multi-agent itinerary planning), to solve problems related to the latency of data collection and the overall imbalance of energy consumption in the energy network in wireless sensor networks.

The proposed GA-MIP algorithm is based on the geographical distance and the size of the collected data that impact the energy consumption of the sensor nodes. With these two factors, GA-MIP provides a new approach to determining the number of mobile agents and the clustering of source nodes. Once the number of the mobile agents and the node groups are defined, GA-MIP identifies the itinerary to be visited by each mobile agent using the GRASP heuristic algorithm (Greedy randomized adaptive search procedure). The simulations carried out showed the performance and efficiency of GA-MIP to reduce the time of data collection and to minimize the energy consumed.

3.2 A Kalman framework based mobile node localization in rough environment

Given the importance of precise location of nodes in RCSFs, a mobile method has been proposed by Chu and Cheng-dong [18] for RCSFs in LOS / NLOS environments. The measurements used in the proposed method are time difference by arrival (TDOA) and received signal strength (RSS) [19].

Firstly, this approach adopts the general likelihood ratio method to identify the propagation condition in Kalman framework. According to the condition identified, the method VB-AKF (variation of Bayesian approximation adaptive Kalman filtering) is used to filter the NLOS error in the NLOS condition, and the classic Kalman update is used in the LOS condition. Thereafter, the maximum likelihood method is used to estimate the position of mobile node. Compared with Kalman filter and $H\infty$ filter, the proposed method by simulation has the highest localization accuracy with varying parameters.

4 Technical methods used in the proposed approach

4.1 Key management based on the AVL tree

After deployment, sensors need to establish cryptographic keys with their neighbors to communicate securely. However, static management is not appropriate after a long time.

In this paper, we proposed a management of cryptographic keys ECC based on the tree AVL [20] (the name comes from the last name initials of those who created this structure: Adelson, Velskii and Landis). In Fig. 1, the cryptographic keys are represented by circles and the sensor nodes by squares.
In the AVL tree, each sensor must have all the keys corresponding to the nodes of the AVL tree. In addition, the sensors do not know the keys they do not need. For the balancing factor, each node of the tree, the difference in height of its sub-tree is a maximum 1.

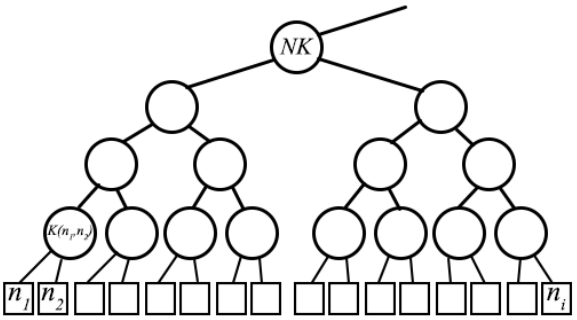

Figure 1. Keys and sensors in the AVL tree.

$h(T_L)$: the height of the subtree to the left
$h(T_R)$: the height of the subtree to the right

$$|h(T_L) - h(T_R)| \leq 1 \quad (1)$$

This obviously leads to the property that any sub-tree of a balanced binary tree under the AVL tree is balanced [20] in the AVL sense.

The main question is to determine the maximum height of the balanced AVL binary tree. To find out, it is necessary to determine the minimum number (NB_h^{min}) of nodes of a balanced binary tree AVL using (of) a height (h).

$$NB_0^{min}=0,\ NB_1^{min}=1,\ NB_2^{min}=2 \quad (2)$$
$$NB_h^{min}=1+ NB_{h-1}^{min}+NB_{h-2}^{min} \quad for\ h\geq 2 \quad (3)$$

This recurrence has as a solution Fibonacci number (F) to order h + 2 minus 1. So:

$$NB_h^{min}=F_{h+2}-1 \quad (4)$$
$$NB_h^{min} =1/\sqrt{5}*(((1+\sqrt{5})/2)^{h+2}- ((1+\sqrt{5})/2)^{h+2})) -1 \quad (5)$$
$$NB_h^{min} \approx 1/\sqrt{5}*(((1+\sqrt{5})/2)^{h+2}) -1 \quad (6)$$

So conversely, the maximum height h of an AVL tree containing *n* elements is such that:

$$1/\sqrt{5}*(((1+\sqrt{5})/2)^{h+2}) -1 \leq n \quad (7)$$
$$h\leq log_a((n+1)*\sqrt{5})-2 \quad with \quad a=(1+\sqrt{5})/2 \quad (8)$$
$$h\leq log_a(2)* log_2 (n+1)- log_a(\sqrt{5})-2 \quad (9)$$
$$h\leq 1,44* log_2 (n+1)- 0,328 \quad (10)$$

The worst case of complexity of balanced binary trees under the AVL is *log(n)*.

4.2 Cryptography based on the elliptic curves

Cryptographic systems based on the elliptic curves allow to obtain a gain in efficiency in key management. In fact, such cryptosystems require a small size [22] of keys (for example, a 160-bit key when RSA use a 1024 bit key, with an equivalent level of security) this represents a significant advantage for wireless sensor networks [23] whose memory space is very limited. In addition, the algorithms of calculations related to elliptic curves are faster, and therefore have an important debit to generate and exchange the keys [24].

The elliptic curve (E) is an algebraic curve [16] that can be represented by the Weierstrass equation:

$$E:\ y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (11)$$

We suppose that the curve is defined in a field (K) and the parameters $a1, a2, a3, a4, a6 \in K$. In order to obtain a smooth curve with no reversal point, it is necessary for the discriminant of the curve to be $\Delta \# 0$. The complete definition of Δ is represented in the formula (12).

$$\Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 - 9d_2d_4d_6 \quad (12)$$

Where: $d_2 = a_1^2 + 4a_2$

$d_4 = 2a_4 + a_1a_3$

$d_6 = a_3^2 + 4a_6$

$d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2$

The equation of the elliptic curve can be simplified if the curve is defined on a first field (Fp). The Weierstrass equation can be transformed into a simple equation as the following:

$$E:\ y^2 = x^3 + ax + b \quad (13)$$

Where: $a, b \in Fp$.

The discriminant of the curve:

$$\Delta = -16\,(4a^3 + 27b^2) \quad \text{and} \quad \Delta \# 0 \quad (14)$$

Actually, this is the curve's form used in this paper.

5 The proposed method

In this paper we propose architecture based on mobile nodes with significant capacities of calculations and no energetic constraints as they are fixed to movable empowered structures. The mobile nodes will be responsible for ensuring a good level of security by generating the ECC key [25], storing them in the AVL tree, and making periodic rekeying. The operations performed by the mobile nodes can minimize the cost of communication between the nodes and the base station which increases the lifetime of the entire network. Moreover, the results estimated how many messages we minimize during installation of the keys and how fast generating and storing of the ECC keys occur. In section 6, we also compared the result with the following implementations; specially RECC [12] and AVL-Headers/AVL-KDC [13]. The obtained results proved that the proposed algorithm can outperform other approaches based on the ECC.

After deployment, the mobile nodes cooperate to secure all clusters that are on their ways. In order to maintain security of the entire network [26], it is necessary to establish a strategy for coordination between mobile nodes and estimate the time of intervals between rekeying passages. To evaluate the performance, the mobile approach is implemented as a prototype system on the WSNs.

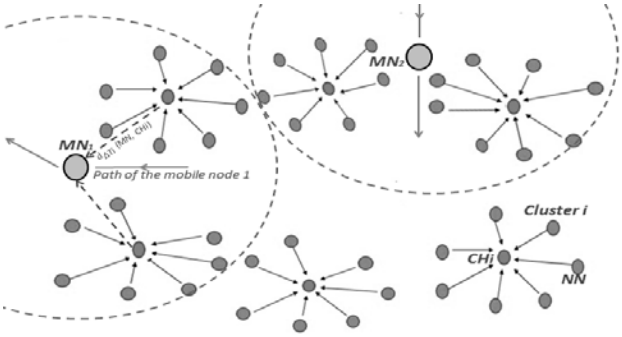

Figure 2. Coverage by two mobile nodes.

The power of the mobile nodes enables them to have a significant coverage radius, and nodes located within that radius can exchange directly with the mobile node. However, nodes outside the cover (out of the radio coverage) may receive the messages from its CH, not directly from MN.

5.1 Assumptions and requirements

This so-called approach is based on the following assumptions and requirements:

- The mobile nodes are powerful as a workstation capable of sending the wide-range radio signals.
- Each MN has no constraints on energy, computing, and storage capacity.
- Normal nodes have the same capacity in processing, energy, and storage.
- The compromise of a node means that all information stored in its memory are known by the attacker.
- Communication channels are bidirectional, i.e., if a node 'x' can receive such a message from another node 'y,' it is possible likewise.
- An attacker can listen to all traffic; return old messages, or inject their own messages.

5.2 The main phases of the mobile approach

Three phases compose the proposed approach. During the deployment phase, all nodes are preloaded by the same network key (NK) which will later be the network broadcast key, as well as it the root of the AVL tree. After the clusters' formation in the setting phase, the mobile nodes calculate their distances with cluster heads and establish the membership list. Thereafter, each cluster head forms its member's list and sends it to the mobile node to which it is attached. Each mobile node generates and stores the ECC keys of its clusters in the AVL tree afterwards; it sends the ECC key for each node of the selected cluster. During the rekeying phase, each mobile node calculates its passage cycle based on the number of clusters and nodes per cluster. When an area is covered by several mobile nodes, the CH sends in the arrival of each MN the time elapsed since the last rekeying. If the time is longer than the rekeying time, the MN starts rekeying operations, if not the MN moves to the next cluster.

- ***Deployment phase:***

We proposed that all nodes are randomly deployed and are preloaded with a network key NK to communicate with the CH and neighboring nodes. Then, each node broadcasts a "Hello" message with its ID to all neighbors within its listening range. The nodes in a node's neighborhood respond by sending their own ID. Each node prepares a list of its neighbors $N(n_i)$ from distances $d_{i,j}$ between them. The rest of this phase is described in the following algorithms:

a- Initialization:

n_i ➔ broadcasts "Hello" message with its ID;
n_i ← messages from neighbors;
IF no message received THEN n_i is ORPHAN;
ELSE n_i receives messages with ID and:
n_i : collects IDs for received messages;
the neighbor set $N(n_i)$;
n_i computes d_{ij}, for all neighbors $n_j \in N(n_j)$;
n_i : computes $e(n_i)$ for node to become CH;
n_i ➔ $(d_{ij}, N(n_i), e(n_i))$ to all $n_j \in N(n_j)$;
computes weight $w(n_i)$;
n_i ➔ $w(n_i)$ to all $n_j \in N(n_j)$;
collects $Nw(n_i)$ weights of distance 1 nodes;

b- Cluster Formulation:

IF $(e(n_i) = 1)$ and $(Nw(n_i) \neq 0)$ THEN
IF $(w(n_i) \geq w(n_j))$ for all $n_j \in N(n_i)$ THEN
n_i : initiates a cluster; ELSE
n_i : joins a cluster formed by $n_j \in N(n_i)$;
ELSE each node computes $|N(n_j)|$ for all $n_j \in N(n_i)$;
IF $|N(n_j)| \geq |N(n_i)|$ for all $n_j \in N(n_i)$ THEN
n_i : initiates a cluster; ELSE
n_i : joins cluster created by n_j;
n_i ➔ broadcasts invitation ;
n_i ← the responses to form $S_i = \{n_i, n_{i1}, n_{i2}, \ldots, n_{il}\}$;
n_i ➔ S_i to all neighbors;

c- Cluster Reduction:

IF $(w(n_i) = 0)$ and $(f_s * |S_i| \leq |N(n_j)|)$ THEN
Enquire: n_i asks members of S_i of their alternate cluster choices and receives responses;
Reduce: If > $(f_r * |S_i|)$ members have choice to join other cluster then send message to accept invitation;

d- Cluster Head Selection:

IF $|S_i| \,||\, N(n_j)|$ THEN
n_i : is confirmed as a CH; ELSE IF
n_{ij} : can be reduced THEN
create connectivity by reducing n_{ij};
ELSE IF $e(n_{ij}) = 1$ and $(S_i - n_{ij}) \in N(n_j)$ THEN
n_{ij} : is confirmed as CH_i;

- ***Configuration phase:***

During the first pass, the mobile nodes establish the membership list from the calculated distances between them and the CH. After each MN_i they select the nearest cluster head and ask it to send its list of members. Once received by the MN, then the cryptographic keys begin to generate depending on the number of node by cluster. When the key generating ends, the MN_i stores the keys in the AVL sub-tree of the cluster. At this moment the MN_i sends the tree branch that contains all ECC keys allowing each node to communicate with its neighbors and the CH. When sending of the key ends, the CH is responsible for sending the ECC keys required for nodes belonging to the cluster, which are outside the coverage of all MN, hereinafter, the algorithm of this phase.

a- Collecting Information:

MN ➔ message "Hello" to CH_i with its ID;
MN ➔ messages from CH_i;
IF no message received THEN MN falls asleep for t_w and sends the message "Hello";
ELSE MN receives messages with ID and an estimate of the distance based on the received signal power.
MN : collects (ID, $d_{CHi,MN}$) for received messages;
the CH_i set $N(CH_i)$;

b- Generating ECC Keys:

MN : computes for each cluster all public and private keys requiredfor all nodes, following this algorithm:
k_0=1 and k ← (k_{n-1},…,k_1,k_0)2
Set P_1 ← P, P_2=2P
For i from i-2 down to 0 do
If k_i=1 then
Set P_1 ← P_1 + P_2, P_2 ← $2P_2$.
Else
Set P_2 ← P_2 + P_1, P_1 ← $2P_1$.
end for
RETURN (Q = P_1)

c- Insert the keys in the AVL tree:

MN : inserts the keys generated in the AVL tree, T denotes the tree
pointer and x denotes ECC keys:
If T=null THEN
T:=new tree; T.data:= x; height:=0;
case
T.data = x : return ;
T.data > x : return Insert(T.left, x)
if ((height (T.left) – height (T.right)) = 2) {
if (T.left.data > x) THEN
T = RotateFromLeft (T);
else T = DoubleRotateFromLeft (T);}
T.data < x : return Insert (T.right, x)
Similar to the left case
Endcase

d- Send the necessary keys:

MN: computes the neighbors of each node based on its position in the AVL tree.
MN ➔ all keys to communicate with them in the selected cluster.

- ***Rekeying phase:***

The need for security is different from one application to another and the risks to which the nodes are exposed should be studied in advance. When an application requests a high level of security, it is recommended to enforce the periodic rekeying T (for example every hour), in order to limit the impact of key compromise on data flows. One of the advantages of the proposed approach is that a mobile node renews the cryptographic keys of all clusters during each pass Ti, greatly increasing the security of the entire network. During the ΔTi interval, the mobile node compares the nodes list during the last pass and the list sent by the CH to check if a node has left or has just arrived in the cluster. Hereinafter, a chronology of the rekeying cycles ΔTi for each cluster and periods of passing Ti for a mobile node.

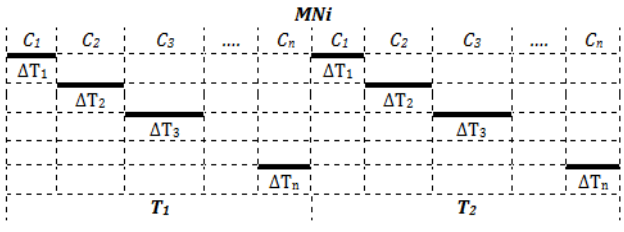

Figure 3. Example of rekeying by a mobile node.

6 Simulation

6.1 Methods based on the elliptic curves

A- Routing Driven Key Management (RECC)
The *RECC* method [12] is based on routing protocol for heterogeneous sensor networks. The main idea is that a node can communicate only with a few neighbors. Before installing the keys, the protocol calculates the itinerary tree using a service to evaluate the location of each node, which requires several exchanges of messages between the nodes. All these communications are not protected, allowing an attacker to capture and edit these messages. However, the routing protocol does not have the ability to add a new node or to update keys.

B- AVL-Headers and AVL-KDC
Network contains a large number of nodes that form clusters around the node elected Cluster-Head (*CH*). They receive and then transmit nodes messages to the base station. The *ECC* keys generation and storage in the *AVL* tree are done by the base station while distribution is done either by a *KDC* server or by *CH* [13].
In *AVL-KDC* approach, when a node arrives or leaves the network, the *KDC* server starts updating public keys located throughout the node path to the base of the *AVL* tree and reconfigures the *AVL* tree. Then, it sends an update message to all Headers. Thereafter, each Header broadcasts the message to all nodes in the cluster.
In *AVL*-Headers, *CH* starts updating the *ECC* cryptographic keys on the branch of the node that has just arrived or left the *AVL* tree; it reconfigures and rebalances the *AVL* tree. Finally, it broadcasts the updates to all nodes in the cluster that depletes its energy resources.

6.2 Simulation parameters and metrics

The different algorithms are implemented with the software TOSSIM (TinyOS Simulator) and we use IEEE 802.15.4 as a communication model for the network.
In this part of the paper, we evaluated the performance of the proposed mobile algorithm for distribution and periodic MADR rekeying by comparing simulation with both methods RECC and AVL-KDC / AVL-Headers, hereinafter, the simulation parameters for a normal node.

Table 1. Simulation parameters of a normal sensor

Parameters	Values
Deployment Surface	(0,0) X (10000m, 10000m)
Eelec	50nJ/bit
Eamp	10pJ/bit/m2
Initial Energy of a normal sensor	10J
Active mode	0,073J/s
Sleep mode	0,005J/s
Data traffic	CBR

In this section we presented the results that correspond exactly to the values obtained by simulations of the studied methods. In order to evaluate each method performance, we will focus on the following four metrics:

- Energy consumption per node/cluster head
- ECC Keys computing time and rekeying
- Number of stored keys
- Number of exchanged messages

6.3 Comparison of metrics

A- Comparison of Energy consumption by sensor

The lifetime of a network depends heavily on the energy consumed by the sensors that compose it. Several studies showed that a large part of energy consumed in the WSN is due to wireless communications and the calculations performed by the processing units.
In the following simulation we focused on the operations performed by the mobile nodes that have no energetic constraint to ensure proper security and increase the lifetime of the entire network. The following graph shows the energy consumed by each method during the distribution of the ECC keys.
The results show that the RECC method consumes more energy because of the large number of packets exchanged during distribution and the ECC key installation. By against, the power consumed by normal nodes in both approaches AVL-KDC and AVL-Headers do not depend with the size of the network.
We can see after this comparison that the mobile

nodes and their permanent energy can drastically save energy consumed and that this is the strong point of the proposed approach.

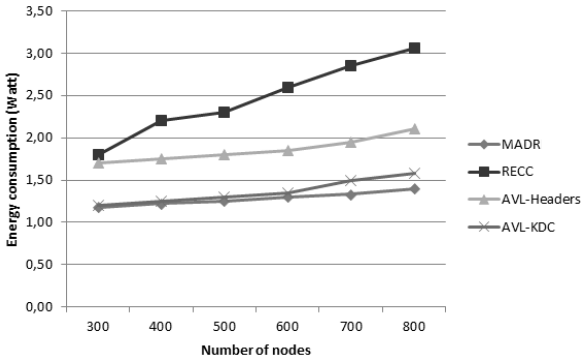

Figure 4. The energy consumption per node.

B- The ECC Keys computing time and rekeying

More time do regeneration and distribution of cryptographic keys take, more vulnerable the network becomes. When the number of cluster per node is large, the generation will take more time because of the complex calculations in order to generate the ECC keys. After deployment of the RECC approach, the nodes evaluate their positions in the network and calculate the routing table. All these operations require several message exchanges and more time before installing the keys.

A header from the AVL-Headers takes at least 15 minutes to calculate and manage ECC keys because of its limited computing capabilities, which puts the entire network at risk. In the MADR approach, the diffusion and rekeying of cryptographic keys are ensured by the mobiles nodes. The significant power calculation allows the mobile node to manage keys in a cluster of 400 nodes in 752 seconds and rekey the same cluster every 380 minutes. At the end of each pass Ti, each sensor node must have new pairwise keys shared with the members of the neighbors' list $N(n_i)$.

It should be noted that when the rekeying period is short, the risk of compromising is also weak. We notice in Fig. 5 that the energy consumed in the rekeying phase depends on the duration of the rekeying period. However, for the rekeying period higher than 500min, the performance of the sensors becomes linear. We can consider this value as optimal for periodic rekeying in the MADR approach.

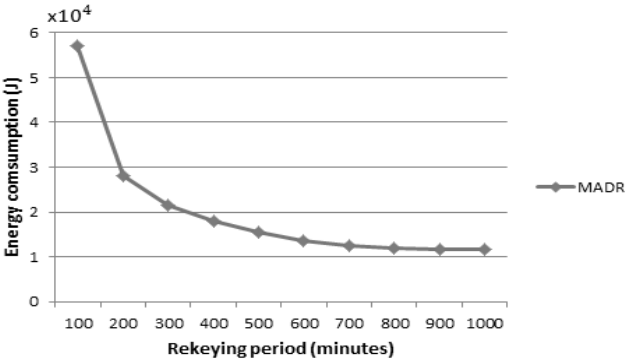

Figure 5. Variation of energy consumption in function of the rekeying period.

C- Number of Stored Keys

The memory used by the nodes and cluster heads to store the cryptographic keys has an important role in the proper functioning of the network. When the used size is large, it requires more treatment and more energy is consumed. According to the size of the network, Fig. 6 shows memory used by the normal node to save the ECC keys.

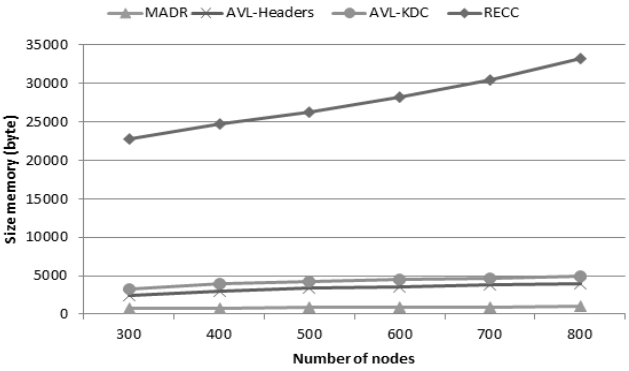

Figure 6. The size of memory used to store the keys by normal node.

The cluster head is preloaded with all of its cluster keys (use almost 24000 bytes); which makes the RECC approach the most gourmand in terms of memory. The AVL-Headers use memory space that varies between 1200 and 2500 bytes. AVL-KDC and MN use a very limited memory space, because the CH is not loaded with the keys of the nodes in the cluster. The server KDC and the MN are responsible for the

AVL tree, which offers to the Cluster Head an economy of storage memory.
The Fig. 6 shows a management method based on the AVL tree. The number of keys stored in the normal node depends only on the height of the AVL tree. Each node has the keys placed in a way to communicate with its neighbors and its Cluster Head. However, the number of keys stored in a normal node in the RECC approach depends on the size of the network, because each node has all the network keys.

D- The Number of exchanged packets

The last metric to discuss in this paper is the number of packets exchanged during distribution and Installation of the ECC keys. A large number of packets exchanged implies more energy consumption and more risk of capturing the exchanged packets.

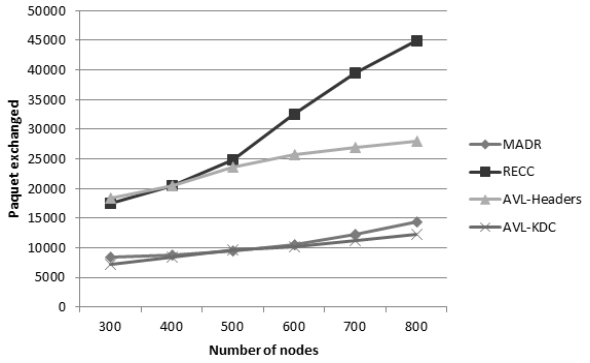

Figure 7. Number of exchanged packets.

In Fig. 7 we compared the number of exchanged packets by the nodes during distribution and rekeying. The RECC approach has no renewal mechanism of cryptographic keys which explain its small number in comparison to other methods. However, the rekeying in the *AVL-KDC* and *AVL-Headers* methods is performed after the outbreak of an external event. The *MADR* approach offers a periodic rekeying for all clusters. For a cycle of 5,400 seconds, the number of exchanged messages during distribution and rekeying is smaller than the remaining methods.

6.4 The security analysis

Time to generate and send cryptographic keys for both methods AVL-Headers and AVL-KDC is very long. Thus, the energy cost of exchanges and calculation has a great impact on the lifetime of the entire network.
A normal node in the RECC approach communicates with all the nodes to be located in the network, after it calculated its routing table. All these operations take a long time before the generation and distribution of the cryptographic keys, which makes applications based on the WSN vulnerable to attacks.
We found that the MADR approach has three strong points. The first deals with the lifetime of the network, mobile nodes that have no energy constraints ensuring all complex calculations [27] which can deplete the energy of a normal sensor. The second is security; the rekeying is done periodically which significantly reduces the risk of attacks [28]. The last point is regarding the network coverage. The significant power of the mobile nodes allows a large radius of coverage and even the nodes that are outside the coverage of mobile nodes to be managed by their Clusters heads.

7 The conclusion and perspectives

To maintain the effectiveness of critical applications based on the wireless sensor networks, we must ensure a good level of nodes security taking into account their limited energy and computing.
In this paper, we proposed an approach based on mobile nodes for the ECC keys generation, distribution, and periodic rekeying of all clusters. The generation is done by the mobile node which stores subsequently the generated ECC keys in the AVL tree before sending the keys to each node, It needs to communicate with its neighbors and its cluster head.
The collaboration between powerful mobile nodes provides better coverage and a good key management. Due to the significant capabilities of the mobile nodes we can use them to secure critical applications, at the same time if needed in the applications which require difficult operations.

References

[1] Othman, M., Shazali, K.: *Wireless sensor network applications: A study in environment monitoring system*, Procedia Engineering, vol. 41(2012), p. 1204-1210.

[2] Jain, A., Kant, K., Tripathy, M.: *Security solutions for wireless sensor networks.* Second International Conference on Advanced Computing & Communication Technologies, 2012, IEEE, pp. 430-433.

[3] Pandey, A., Tripathi, R.: *A survey on wireless sensor networks security*. International Journal of Computer Applications, vol. 3(2010), no 2, p. 43-49.

[4] Adnan, A., Hanapi, Z.: *Geographic Routing Protocols for Wireless Sensor Networks: Design and Security Perspectives*, International Journal on Communications Antenna and Propagation (IRECAP), vol. 5(2015), no 4, p. 197-211.

[5] Chandra, S., Paira, S., Alam, S.: *A comparative survey of symmetric and asymmetric key cryptography.* International Conference on Electronics, Communication and Computational Engineering (ICECCE), IEEE, 2014, p. 83-93.

[6] Sasi, S., Dixon, D., Wilson, J,: *A general comparison of symmetric and asymmetric cryptosystems for WSNs and an overview of location based encryption technique for improving security*. IOSR Journal of Engineering, vol. 4 (2014), no 3, p. 1.

[7] Kandasamy, R., Krishnan, S.: *Enhanced Energy Efficient Method for WSN to Prevent Far-Zone*, International Journal on Communications Antenna and Propagation (IRECAP), vol.4 (2015), no 4, p. 137-142.

[8] Munivel, E., Ajit, M.: *Efficient Public Key Infrastructure Implementation in Wireless Sensor Networks*, International conference on Wireless Communication and Sensor Computing, February 2010, p. 1-6.

[9] Zhang, Y., W. Yang, W., K. Kim, K., M. Park, M.: *An A VL Tree-Based Dynamic Key Management in Hierarchical Wireless Sensor Network*, International Conference on Intelligent Information Hiding and Multimedia Signal Processing, August 2008, p. 298-303.

[10] Setiadi, I., Kistijantoro, A., Miyaji, A.: *Elliptic curve cryptography: Algorithms and implementation analysis over coordinate systems.* 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), IEEE, 2015, p. 1-6.

[11] Qin, Z., Zhang, X., Feng, K.: *An efficient key management scheme based on ECC and AVL tree for large scale wireless sensor networks.* International Journal of Distributed Sensor Networks, vol. 2(2015), p. 198.

[12] Du, X., Guizani, M., Xiao, Y., Chen, H.: *A Routing-Driven Elliptic Curve Cryptography Based Key Management Scheme for Heterogeneous Sensor Networks*, IEEE International Conference on Communications, ICC 2007, Glasgow, August 2007.

[13] Boumerzoug, H., Boucif, A.: *A lightweight key management scheme based on an AVL tree and ECC cryptography for wireless sensor networks.* Concurrency and Computation: Practice and Experience; vol. 28(2016), no 6, p. 1831–1847.

[14] Schleicher, J., Vögler, M., Dustdar, S., Inzinger, C.: *Enabling a smart city application ecosystem: requirements and architectural aspects.* IEEE Internet Computing 2016, vol. 20(2016), no 2, p. 58-65.

[15] Walravens, N.: *Mobile city applications for Brussels citizens: Smart City trends, challenges and a reality check.* Telematics and Informatics 2015, vol. 32(2015), no 2, p. 282-299.

[16] Jalali, R., El-Khatib, K., C. McGregor, C.: *Smart city architecture for community level services through the internet of things.* International Conference In Intelligence in Next Generation Networks (ICIN), 2015, p. 108-113.

[17] Cai, W., Chen, M., Hara, T., Shu, L., Kwon, T.: *A genetic algorithm approach to multi-agent itinerary planning in wireless sensor networks.* Mobile Networks and Applications, vol. 16(2016), no 6, p. 782-793.

[18] CHU, H., Cheng-dong, W.: *A kalman framework based mobile node localization in rough environment using wireless sensor network.* International Journal of Distributed Sensor Networks, 2015, vol. 2015, p. 73.

[19] Naraghi-Pour, M. and Rojas, G. C., "*A novel algorithm for distributed localization in wireless sensor networks*", ACM Transactions on Sensor Networks, vol. 11, no. 1, pp. 1–25, 2014.

[20] Anderson, R., Seitz, S.: *CSE 326: Data Structures AVL Trees,* 2014.
[21] Salah, S., Maizate, A., Ouzzif, M.: *Security approaches based on elliptic curve cryptography in wireless sensor networks.* 27th International Conference on Microelectronics (ICM). IEEE, 2015, p. 35-38.
[22] Gura, N., Patel, A., Wander, A.: *Comparing elliptic curve cryptography and RSA on 8-bit CPUs*. International Workshop on Cryptographic Hardware and Embedded Systems, Springer Berlin Heidelberg, 2004, p. 119-132.
[23] Enge, A.: *Elliptic curves and their applications to cryptography: an introduction.* Springer Science & Business Media, 2012.
[24] Anitha, R., Nawaz, G.: *Development of a Secure, Energy Efficient and Reliable Routing Protocol for Mobile Wireless Sensor Networks,* (2014) International Review on Computers and Software (IRECOS), vol. 9(2014), no 3, p. 487-494.
[25] Avdaković, S., Jusić, A.: *Dynamic response of a group of synchronous generators following disturbances in distribution grid.* Engineering Review, vol. 36(2016), no 2, p. 181-186.
[26] Kazienko, J., Moraes, I., Albuquerque, C. V.: C.: *On the performance of a secure storage mechanism for key distribution architectures in wireless sensor networks.* International Journal of Distributed Sensor Networks, 2015, p. 44.
[27] Tsou, T., Lu, C., Kuo, S.: (2013). *MoteSec-aware: a practical secure mechanism for wireless sensor networks.* IEEE Transactions on Wireless Communications, vol. 12(2013), no 6, p. 2817-2829.
[28] Geetha, R., Kannan, E.: *Secure Communication Against Framing Attack in Wireless Sensor Network,* (2015) International Review on Computers and Software (IRECOS), vol. 10(2015), no 4, p. 393-398.

Nomenclature

- *AVL* denotes a binary tree (authors names are: Adelson, Velskij and Landis).
- CH_i denotes a cluster head.
- d_{ij} denotes distance measure from node n_i to n_j.
- $d_{CHi,MN}$ denotes distance between the *CH* and *MN*.
- *ECC* denotes the cryptography based on Elliptic Curves.
- $e(n_i)$ denotes the readiness of node n_i to become a *CH*.
- f_s small fraction.
- f_r fraction of reduction.
- *ID* denotes a node identifier.
- k random number of the order of n (*1* to *n-1*)
- *KDC* denotes a keys distribution center.
- *MADR Mobile Algorithm for Key Distribution and Periodic Rekeying.*
- *MN* denotes the mobile node.
- n_i or n_j denotes a node.
- $N(CH_i)$ denotes the *CH* list.
- $N(n_i)$ denotes the neighbors of node n_i.
- $Nw(n_i)$ is weight of neighbors.
- *ORPHAN* denotes that the node has no neighbor.
- P denotes a point of the elliptic curve E.
- S_i is cluster with $I + 1$ member nodes.
- t_w denotes waiting time.
- $w(n_i)$ denotes weight of nodes.
- *WSN denotes a wireless sensor networks.*
- ➔ send a message.
- ← receive a message.