Secure and Energy-Efficient Geocasting Protocol for Hierarchical Wireless Sensor Networks

Vianney Kengne Tchendji*, Blaise Paho Nana*, A. Yvan Guifo Fodjo*

*Department of Mathematics and Computer Science
Faculty of Science
University of Dschang
PO Box 67, Dschang-Cameroon
vianneykengne@yahoo.fr, blaisepaho@gmail.com, yvanguifo@gmail.com

ABSTRACT. Wireless sensor networks are now used in many applications (medical, agricultural, military, fire detection, the Internet of Things, etc.) needing a high security level and better energy saving. In this paper, we present a geocasting protocol, secured by the use of elliptic curves as a generator of secret keys. For this purpose, we begin by presenting a technique of grouping sensors in clusters or zones, then in cliques. This clustering creates a virtual architecture that facilitates the routing, the management of the network, and reduces the energy expenditure. We then propose for this architecture, our secure geocasting protocol, energy efficient and with low memory needs, which makes good use of the built architecture. The geocasting uses the skills of the sink to quickly and more efficiently reach geocast regions. The security aspect of our protocol, based on elliptic curves, offers a good level of security that can also limit damages in case of eventual attacks, facilitates the distribution and keys refreshment (a recurrent problem) without ever having to pass these keys in the network, and remains effective against several types of classic attacks such as passive or active listening, identity theft, black hole, data replication, etc .; but remains vulnerable to the problem of frequency jamming.

RÉSUMÉ. Les réseaux de capteurs sans fil sont de nos jours utilisés dans plusieurs domaines (médicale, agricole, militaire, surveillance, Internet des objets, etc.) à fortes exigences en matière de sécurité et d'économie d'énergie. Dans ce document, nous présentons un protocole de géocasting, sécurisé grâce à l'utilisation des courbes elliptiques comme générateur de clefs secrètes. Pour ce faire, nous commençons par présenter une technique de clustérisation de la région d'intérêt en zones ou clusters, puis en cliques. Cette clustérisation met en place une architecture virtuelle qui facilite le routage, permet une meilleure gestion du réseau, et réduit la consommation énergétique. Nous proposons ensuite pour cette architecture, notre protocole de géocasting, économe en énergie et en espace mémoire, qui utilise à bon escient l'architecture sous-jacente. Le géocasting utilise les atouts du sink pour atteindre plus rapidement et efficacement les régions de géocast. L'aspect sécuritaire de notre protocole, basé sur les courbes elliptiques, offre un bon niveau de sécurité qui permet de limiter les dégâts en cas d'éventuelles attaques, facilite la distribution et le rafraîchissement des clefs (un problème récurrent) sans jamais avoir à faire transiter ces clefs dans le réseau. Il reste efficace face à plusieurs types d'attaques classique tels que l'écoute passive ou active, l'usurpation d'identité, le trou noir, la réplication de données, etc.; mais reste vulnérable au problème de brouillage de fréquence.

KEYWORDS : Wireless sensor networks, geocasting, hierarchical clustering, virtual architecture, clique, cluster, elliptic curve, security.

MOTS-CLÉS : Réseaux de capteurs sans fil, géocasting, clustérisation hiérarchique, architecture virtuelle, clique, cluster, courbes elliptiques, sécurité.

1. Introduction

Several progress have been achieved this recent years in the fields of microelectronics, micro technology, and wireless communication technologies. Theses advances enable the production of sensors of small sizes, at low cost and at the forefront of the technology. Yet, they are still subject to several constraints: limited energy, low computing power, small memory storage, etc. Massively deployed in a given area (usually for monitoring reasons) which is most of the times hostile to humans, they are able to organize and self-configure into a wireless network called Wireless Sensor Network (WSN). This type of network requires hundreds or even thousands of units that are mass-produced in an environment where testing is a luxury. Each unit is usually equipped with a single use battery, irreplaceable and non-rechargeable for various reasons (cost of batteries replacement, hostility of the area to be monitored, etc.).

Many WSN-based applications make use of geocasting to send messages to sets of receivers in a well-defined geographic area [5, 7, 14]. This technique is of a particular interest especially for civil security. During a sinister or a natural disaster for example, police forces and firefighters may need a geocasting mechanism to join any other actor in an area of the disaster. This technique is also used in agriculture when watering a specific area, or even in military applications, where information must often be provided to all soldiers located in a given area. In addition, it can provide a commercial use, especially to allow anyone passing near a store to immediately receive advertising information.

Promoted for their ease of deployment, WSNs faces many challenges, some of the most important are related to the energy consumption, security and reliability of information circulating in such a network. We are presently seeing WNS-based applications flourishing and moving towards the Internet of Things (IoT) [5, 15]. Thus, for military or medical applications, the need to provide a reliable security solution seems important or even compulsory [3]. Even if the context has evolved, from the machine not connected to wired and wireless networks, the goal of security has always been the same overall, namely to provide basic security services such as authentication, control and security access, confidentiality, integrity, availability, etc. However, because of the characteristics of WSNs (lack of preset infrastructure, dynamic topology, large number of sensors, limited physical security, modes and deployment areas offering multiple possibilities of attacks, etc.) coupled to the inherent constraints of sensor nodes, securing sensor networks is nowadays the source of many scientific and technical researches [3, 4, 11, 12]. Earlier research has shown that the security solutions offered for wireless networks (mobile and ad hoc), especially those based on the use of public cryptography key are very heavy for WSNs [8]. It is therefore important that the security solutions implemented should be the least expensive in terms of resource consumption and aim to reduce delays, number of communications, and the bandwidth occupancy. In other words, these solutions should provide maximum security while preserving the lifetime of the sensors.

In this paper, we are interested in the geocasting and the security problems in WSN. The geocasting is energy-efficient, fast and with less flooding thanks to the use of the Wadaa et al.'s virtual architecture protocol [1] and the Sun et al.'s secure clique formation protocol [12]. We also seek to define inexpensive energy mechanisms and solutions for wireless sensor networks that take into account the relative weaknesses of defense of an autonomous network. For this purpose, we apply energy-saving symmetric cryptography solutions based on elliptic curve-based asymmetric cryptography mechanisms to secure the geocasting protocol that we propose.

The rest of this paper is organized as follows: section 2 presents the construction of the virtual architecture; section 3 presents our geocasting protocol; we secure the whole protocol in section 4; and we conclude after the simulation results of section 5.

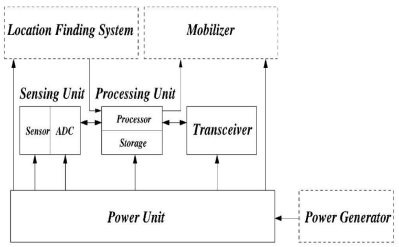
2. Setting up the virtual architecture of the network

2.1. Hypotheses

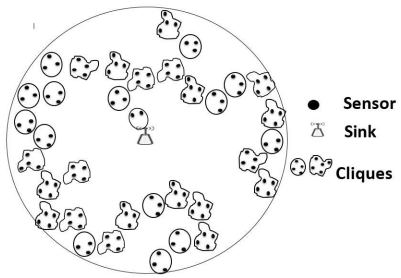
The work done in this paper is subject to the following assumptions: each sensor of the network is static, has an unique *ID*, a GPS (or ASP [9]) and is loaded with an elliptic curve that will help secure data in transit; it is also able to estimate its residual energy; its internal clock is synchronized with that of the base station (sink) so that it can wake up at defined time intervals to collect informations and remain idle the rest of the time; all the deployed sensors are connected and are aware of the number "c" of coronas and the number "s" of angular sections of the architecture; Adding or removing a sensor is allowed but it is a rare event. The base station has the possibility to broadcast messages in the network either at different radii coverage or at different angles and constitutes the only reliable and incorruptible entity of the network; Time is slotted and each message sent by a sensor is received by the sensors in its vicinity within a slot.

2.2. Sensor's model

The sensor is the basic element of any WSN. It must have a sensing unit, a radio module to receive or transmit data, a storage and computing module, all powered by a small battery. This equipment can be completed according to the requirements of the application. Figure 1a shows an example of a sensor with other optional equipment (framed with dotted lines).



(a) Generic model of a wireless sensor.



(b) Deployment of sensors and formation of disjointed cliques.

Figure 1: Deployment of sensors and cliques' formation.

2.3. Formation of cliques

After a massive deployment around the base station also called the sink (figure 1b), it is time for sensors to self-configure, collect environmental data and route them to the sink. To do this, we start by grouping the sensors into small disjoined groups using Sun et

al. protocol [12]. This technique presents a network partitioning protocol using the cluster first (CF) approach. This approach requires the fact that each node accepts membership in a group before the leader's election. This is how the network is partitioned into cliques in which each sensor is at a single hop from any other sensor of the same clique. This protocol has the following properties: it is essentially distributed and each node calculates its clique's membership by sending messages to its direct neighbors; when the partitioning algorithm terminates, the participating nodes that do not follow the specifications of the protocol (sending superfluous or conflicting messages) are systematically identified and removed from each clique; at the end of the partitioning algorithm, the network consists of disjoint cliques and each node has a clear view of the member nodes of its membership clique. After the cliques' formation as on figure 1b, the formation of the zones follows.

2.4. Formation of zones

Consider here that the sink is a special node capable of performing omnidirectional transmissions with certain radii for the formation of concentric discs (or coronas) and directional transmissions at certain angles for the formation of angular sections. Once deployed in the supervised zone, the sensors, each having a unique identifier, are grouped in clusters or zones (as described in [1]) according to the angular sectors and coronas (figure 2). In this way, the intersection of a corona c and an angular sector s constitutes the zone (c, s). In addition, since the network is parse, it will be important to identify empty zones or clusters (section 2.5). This will allow the sink node to have an overview of the areas actually covered by the sensors. Note that when a clique is shared between several zones, after the CH1 (clusterhead of a clique) election, the sensors of this clique will behave as part of the zone where their CH1 is located. Now, let's build the architecture and proceed to the CH1 and CH2 (clusterhead of a zone) elections.

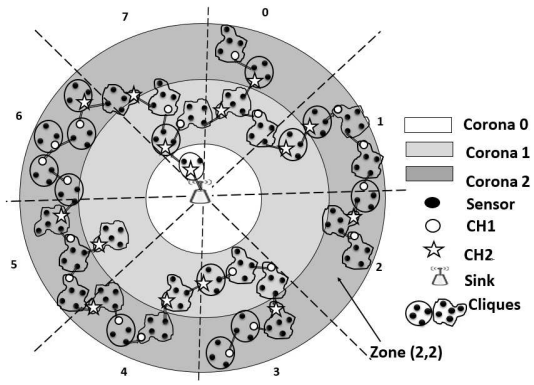


Figure 2: Formation of zones.

2.5. Hierarchical structure, detection of empty zones and election of clusterheads

The empty zones discovery process is similar to the one presented in one of our previous work [13] with some differences and inspired by [10]. In this process, there's a part executed by the sink, and a part performed by the sensors.

Base Station Algorithm: Before running the algorithm, the sink node periodically broadcasts an alert message across the entire monitored area, specifying to the sensors the date the discovery algorithm will begin. All the sensors are awake and the sink node initiates the detection of the empty clusters by spraying in the first corona a $Detect(-1,-1)$ message containing a variable $NumberOfHops$ initialized to 0. This variable allows each sensor to evaluate its distance (in terms of the number of hops) from the base station; It will also serve as a selection criteria for the different clique (CH1) and cluster (CH2) leaders. Initially, each sensor initializes its $NumberOfHops$ variable to $+\infty$. The network being connected, it is sure that at least one sensor of the first corona will receive the message $Detect$. During the algorithm, the sink listens to answers coming from the area of interest. At each message reception, the sink node maintains two routing tables: h and $relay$. The cells of table h initialized to 0 contains a bit 1 in the cell (i,j) if the sink has received a message from the cluster (i,j) or a 0 if not; and the relay table contains in its cell (i,j) the coordinates of the relay cluster (or relay zone) of the zone (i,j), i.e. the closest zone to the sink in the vicinity of cluster (i,j) through which its data passes to reach the sink. This allows the base station to have an overview of the areas covered in the network.

Algorithm of a sensor: When the $Detect$ message arrives especially in a cluster of the first corona and in general in any other cluster, the sensors execute approximately the same algorithm. This algorithm is inspired by the one proposed in [13]. When a sensor receives such a message, it checks that it comes from a neighboring cluster or a neighboring clique. In the case it is the first to receive this message (i.e. it has not yet received a $DetectTimer$ message), he broadcasts a message $DetectTimer$ containing the election's beginning date of the leader of its clique, and retransmits the $Detect$ message to allow the discovery of other zones once it has incremented the $NumberOfHops$ variable.

Election of leaders: Within a clique, the election of CH1s takes place in two steps requiring only two message transmissions (Head1 and Head2). Each sensor having a residual energy greater than the threshold energy Es calculates and starts the countdown of a timer that lasts $(1 - \frac{1}{NumberOfHops} + \frac{e^{-ID}}{\Lambda})$ slots. When this timer expires, it broadcasts a Head1 message to propose itself as leader of the clique if it has not yet received one. The sender and receivers of the previous Head1 message calculate and arm a second timer (which lasts $(\frac{1}{Er} + \frac{e^{-ID}}{\Lambda})$ slots) at the end of which, only the CH1 will be able to broadcast the message Head2 preventing at the same time any sensor that receives this message to diffuse another. These timers are calculated such that the priority should be given to the sensor having the most energy, the one closest to the base station, and the *ID* is used to decide in case of concurrent access. Note that after the $Detect$ message is broadcast by a sensor, it specifies the election start date within its clique. And when the election begins within its clique, it is after a slot that it will start in the clique which just received the message $Detect$. After more than 2 slots, if within a clique there is no longer message transmission, then the election of CH1 is over and we can start the election of the CH2 (leader of a given zone).

When the election of CH1s ends within a zone, there is an immediate broadcast of a Head3 message from the elected CH1. At each reception of a Head3 message, the elected CH1s memorize the parameters of the transmitter and make updates to keep each time the best candidate for the CH2's post i.e. the one closest to the sink. Note that with the CH1 already in place, the communication between two direct neighbors needs at most 2 slots. A CH1 knows that the election is over if after more than 2 slots it no longer receives a Head3 message. This mechanism allows the various elected CH1 and CH2 to know the number

of CH1 in their area. After the election of the CH2 of the zone, the latter broadcasts a message Head4; The neighboring CH1s will receive the Head4 message in one slot and will immediately rebroadcast the Head4 message. Any other CH1 that receives the message Head4 keeps the sending clique as its relay clique i.e. the one through which its messages will pass to reach the sink.

Hierarchical structure and packet routing: We have thus constructed an hierarchical structure with several levels. The bottom of this structure consists of ordinary sensors grouped into cliques. Each clique is driven by a leader of the first level: the CH1. The CH1s are managed by super clusterheads (CH2) which also report to the sink. In a given clique, when data are collected at the level of the ordinary sensors, these data are transmitted to their CH1 which will then be responsible for retransmitting them to their relay clique. This message can be received either directly by the CH1 of the relay clique at the best of the cases, or by a member of this clique in which case it transmits it to its leader (the CH1 of the relay clique). When the CH1 of the relay clique is also the CH2 of the zone, the message is transmitted to the relay zone either to a CH1 of a clique of the relay zone (at best), or to an ordinary sensor of a clique of the relay zone. This is how messages are routed from cliques to zones and zones after zone until they reach the sink.

3. Scenario of geocasting

Here we implement a procedure that allows any sensor in the network to communicate with sensors of other areas. Note that sensors have no way of keeping a global view of the network because of their limited means. On the other hand, the communication between two zones, though neighboring or geographically close zones, is not always obvious because of the empty zones bypassing during the discovery. We assume that sensors know only their direct neighboring cliques (good use of memory) and that a sensor of clique A has a data D that it wants to send to the sensors of a region B defined by a set of GPS coordinates. For this purpose, the sensor begins by transmitting this data to its leader. The CH1 in procession of this data will be in several situations that we describe below:

The CH1 can directly reach the sensors of the B region: since the CH1 has a restricted view of its neighbors, it can know if it can reach the CH1 in charge of the region B or at least one sensor of this region. In this case, it broadcasts the message toward this region. Any sensor in this region that receives the message informs its leader, which will also inform all sensors in the given region.

The CH1 cannot directly reach the sensors of region B: In this case, the CH1 processes the geocast message like any other normal data message collected i.e. this message must go from cliques to cliques and from zone to zone until it reaches the sink as described in the above method. If along the road, a CH1 can reach a sensor of the region B, this brings us back to the first case and the preceding algorithm is executed. Otherwise, the message will necessarily reach the base station. This leads us to the third case.

The message reaches the base station: In this case, the sink will use its directional transmitting antenna to directly send the information in the region B. For this, it performs directional transmissions to each CH2 in the region B by modulating the transmission power so that the beam reaches the target region and also by modulating the transmission angle to not divulge the message to sensors that are not concerned. This is illustrated by figure 3. The CH2s will send the message to the CH1s in the geocast region which will also transmit it to the concerned sensors.

Because the transmission is not secure, sensors on the passage of the beam can listen to

the message which is not intended for them and try to send a useless message back to the sink or to a neighboring clique. To avoid this, any sensor that receives the message from the base station verifies that it is in the geocast region to know if this message is also intended for it.

In case of multi-geocasting i.e. for a message intended for several regions, this protocol is easily adaptable by replacing for example the geocast region with a list of regions.

Up to here, our protocol has no security mechanism. The purpose of the next section is to secure all the exchanged messages of this protocol from the clustering stage.

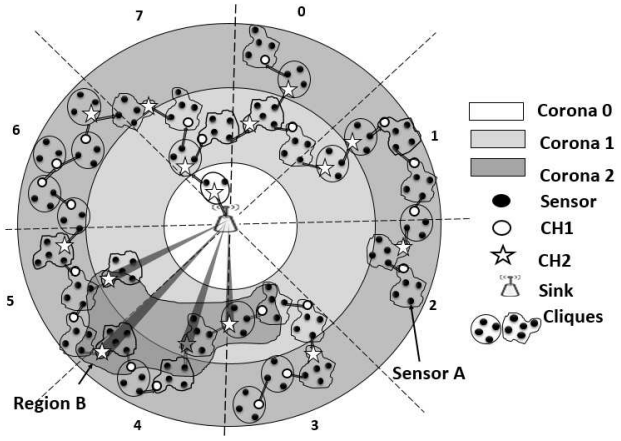


Figure 3: Multizones geocasting.

4. Securing the protocol

4.1. Elliptic Curve Integrated Encryption Scheme (ECIES)

The Elgamal protocol is rarely used directly with elliptic curves. Before encrypting a message, it must first be converted to a point on the elliptical curve used. There are different techniques that exist, but the conversion requires more calculation. In several researches, elliptic curves are commonly used to establish a shared key between the 2 parts of a conversation, after which, a symmetric cryptography algorithm is used to secure the communication between them [4, 8].

The ECIES protocol is indeed a standardized version of Elgamal. Suppose that Alice wants to send a message M to Bob in a secure way, they must first have all the following information: Key Derivation Function (KDF: A key derivation function that allows to generate several keys from a secret reference value); Message Authentication Code (MAC: Code transmitted with the data in order to ensure the integrity of the data); a symmetric encryption algorithm (SYM); an elliptic curve $E(Fp)$ used with the generator point G with $Ord_p(G) = n$; the public key of Bob $K_B = k_B.G$ where $k_B \in [1, n-1]$ is his private key. In this protocol, the critical value is k with which Bob can compute $Z = k.K_B$, and generate the pair $(k1, k2)$ that is used to decrypt and authenticate the message. Due to the difficulty in solving the discrete logarithm problem, Alice can send $R = k.G$ without any problem.

4.2. Security integration

To return to our goal of securing our protocol, here we describe the steps of our secure geocast protocol. We combine symmetric cryptography with the techniques offered by elliptic curves to generate secret keys. Before the deployment, the BS calculates the initial parameters (ID, KDF_{initBS}, MAC_{initBS}, SYM_{initBS}, $E(F_p)$ and $K_{initBS} = k_{initBS}.G$ where $k_{initBS} \in [1, n-1]$) that will be used to execute the symmetrical cryptography algorithm, and charge them in each sensor. The BS also puts the cryptographic material so that once deployed, the nodes can communicate securely and build the network topology. For this purpose, the sink builds a secrete key *k* that will be the same for all the sensors, an elliptic curve *E* which will allow the sensors to know the point of the curve used to secure communications and an initial point *P* (belonging to the first cyclic group *Fp*).

Before the election of the CH2 in a zone (period of confidence), the messages exchanged are secured by the initial parameters loaded in each sensor before the deployment. When the CH2s are all elected, the sink can easily change the keys used by the CH2s (keys of zones) at any time to make it more difficult to an attacker to penetrate the network. Likewise, the technique can be at will repeated by each CH2 to refresh the keys of its CH1s, and by each CH1 to refresh those of its common sensors (keys of cliques). The keys refreshment mechanism is done here in a secure way and with little effort. For this purpose, after a time arbitrarily chosen by the sink, the latter randomly generates a number *x* belonging to *Fp*, it encrypts it with the current key and broadcasts it in the entire network to the attention of the CH2s. When the CH2s receive and decipher the message with their current key, they understand that they must change the current key by jumping *x* points from the current position *P* of the elliptic curve. With this new position, each CH2 easily calculates the new encryption parameters. We are sure that through this strategy, a malicious sensor that has also received information from the sink, according to which the sensors have to change their point on the curve, will not be able to interfere with the operation of the network because they do not have any elements allowing it to determine the points of the elliptic curve.

With this security mechanism, we note that the keys are refreshed without ever having to circulate in the network. The phase of formation of the cliques and zones, election of CH1 and CH2 is entirely secured by the key initially generated and loaded in each sensor. All further exchanges are secure thanks to the random and localized key refreshment performed by the sink, the CH2s and the CH1s. Each time a CH1 changes the key used in its clique (by generating a number *x*), it also notifies the cliques linked to it clique, so that they can recalculate the encryption and decryption key corresponding to each neighboring cluster. Note that two neighboring cliques or two neighboring areas do not necessarily use the same key which would help to circumscribe the damages in case of possible attacks.

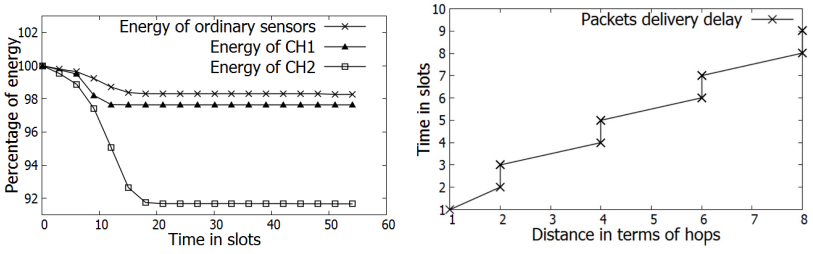
To pass information from a sensor to the sink, the sensor encrypts the message with the key of the clique and sends it to its CH1. The CH1 deciphers the message and encrypts it with the key of the relay clique. The message received by any sensor of the relay clique is then transmitted to the CH1 of the clique which can be at the same time the CH2 of the zone. This is how the collected data are securely routed during date collection and geocasting. When the sink wants to transmit a secure information in a zone, it encrypts it with the key currently used by the CH2 of the zone before transmitting it with its directional antenna. This allows to not divulge the carried messages. In the next section, we perform simulations to study the behaviors of this protocol.

5. Simulation and analysis of the results

Simulations were performed on a laptop (Intel Core i3 CPU M350 @ 2.27GHz $\times$ 4, 8GB RAM, Ubuntu 16.04) with the J-Sim simulator[6]; We deployed 500 sensors each equipped with a battery of 100 joules, a radius of transmissions of 20m. Each transmission requires 35.28×10^{-3} joule and each reception costs 31.32×10^{-3} joule [2]. A slot lasts $78\mu s$. The virtual architecture covers a radius of 400m and has 8×8 zones. The simulation includes the partitioning into cliques, creating the virtual architecture, routing the data to the sink and sending data to the geocast regions. Curves are made with gnuplot 5.0.

Energy consumption: The figure 4a shows the energy consumption for both ordinary sensors and clusterheads of level 1 and 2. This allows us to observe that the energy consumption is more accentuated respectively at the CH2 level, followed by the level of CH1 and finally at the level of ordinary sensors. This is explained by the fact that the CH2 are responsible for the management of the entire zones.

Messages delivery delay: The axis of distance of figure 4b denotes the distance separating the sender sensor from the geocast region. The figure shows that the delivery delay is quite proportional to this distance. Due to the lack of an aggregation technique, the geocast message could be duplicated in some cliques on the way to the geocast region. Nevertheless, these late messages reaches the destination not more than one slot after the expected time. That is why this curve has stairs' shape.



(a) Energy consumption. (b) Packets delivery delay.

Figure 4: Simulations results.

6. Conclusion

In this paper, we have set up a hierarchical clustering protocol to build a layered virtual architecture. We have defined mechanisms to achieve geocasting for one region and for several regions while remaining energy efficient. Subsequently we secured this protocol using cryptographic methods based on elliptic curves. Elliptic curves are used here to generate secret key. Keys are randomly refreshed after a random time without having to circulate in the network. The tests conducted gives acceptable results showing that the proposed solution works and is valid. In addition, the security integrated in our contribution avoids many types of attacks and is equipped with a control mechanism at different levels, which makes it a robust solution. In our next work, we intend to increase a lit-

tle more the challenges by addressing quality of service, fault tolerance mechanisms and including mobile sensors.

7. References

[1] A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, K. Jones, "Training a wireless sensor network", *Mobile Networks and Applications*, Vol. 10, Num. 1-2, p. 151–168, 2005.

[2] D. Wei and S. Kaplan and H. A. Chan, "Energy Efficient Clustering Algorithms for Wireless", *in: Sensor Networks, Proceedings of IEEE Conference on Communications, Beijing*, IEEE, p. 236–240, 2008.

[3] Das, Ashok Kumar, "A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor", *International Journal of Communication Systems, Wiley Online Library*, Vol. 30, Num. 1, 2017.

[4] Dou, Yunqi, Weng, Jiang, Ma, Chuangui, Wei, Fushan, "Secure and efficient ECC speeding up algorithms for wireless sensor networks", *Soft Computing, Springer*, Vol. 21, Num. 19, p. 5665–5673, 2017.

[5] Khan Imran, Belqasmi Fatna, Glitho Roch, Crespi Noel, Morrow Monique, Polakos Paul, "Wireless sensor network virtualization: A survey", *IEEE Communications Surveys & Tutorials, IEEE*, Vol. 18, Num. 1, p. 553–576, 2016.

[6] A discrete event network simulator, J-Sim : *https://sites.google.com/site/jsimofficial/*, 2016.

[7] Panta, Rajesh Krishna, Auzins, Joshua Marc, Fernandez, Maria F, Hall, Robert J, "Geocast protocol for wireless sensor network", *Google Patents, US Patent 9,210,589* , dec 2015.

[8] Saqib, Najmus, Iqbal, Ummer, "Security in wireless sensor networks using ECC, *Advances in Computer Applications (ICACA), IEEE International Conference on, IEEE*, p. 270–274, 2016.

[9] S. Faye, C. Chaudet, I. Demeure, "A Distributed Algorithm for Adaptive Traffic Lights Control", *15th International IEEE Annual Conference on Intelligent Transportation Systems, Anchorage, USA*,September, 2012.

[10] Sébastien Faye, Jean-Frédéric Myoupo, "Deployment and Management of Sparse Sensor-Actuator Network in a Virtual Architecture", *International Journal of Advanced Computer Science*, Vol. 2, Num. 12, December, 2012.

[11] Sébastien Faye, Jean-Frédéric Myoupo, "An Ultra Hierarchical Clustering-Based Secure Aggregation Protocol for Wireless Sensor Networks", *AISS: Advances in Information Sciences and Service Sciences*, Vol. 3, Num. 9, p. 309 – 319, 2011.

[12] Sun Kun, Peng Pai, Ning Peng, Wang Cliff, "Secure distributed cluster formation in wireless sensor networks", *Computer Security Applications Conference. ACSAC'06. 22nd Annual, IEEE*, p. 131–140, 2006.

[13] Vianney Kengne Tchendji, Blaise Paho nana, "Management of Low-density Sensor-Actuator Network in a Virtual Architecture", *ARIMA Journal*, Vol. 27, p. 75–100, 2018.

[14] Wang, Neng-Chung, Wong, Shih-Hsun, "Agrid-Based Geocasting Protocol for wireless sensor networks", *Machine Learning and Cybernetics (ICMLC), 2016 International Conference on, IEEE*, Vol. 2, p. 530–534, 2016.

[15] Yassen Muneer Bani, Aljawaerneh Shadi, Abdulraziq Reema, "Secure low energy adaptive clustering hierarchal based on internet of things for wireless sensor network (WSN): Survey", *Engineering & MIS (ICEMIS), International Conference on, IEEE*, p. 1–9, 2016.