A Survey on Cluster-Based Group Key Agreement Protocols for WSNs

Eleni Klaoudatou, Elisavet Konstantinou, Georgios Kambourakis, and Stefanos Gritzalis

Abstract—**The scope of this survey is to examine and thoroughly evaluate the cluster-based Group Key Agreement (GKA) protocols for Wireless Sensor Networks (WSNs). Towards this goal, we have grouped the WSNs application environments into two major categories (i.e., infrastructure-based and infrastructureless) and have examined: a) which of the cluster-based Group Key Agreement (GKA) protocols that appear in the literature are applicable to each category, and b) to which degree these protocols will impact the systems' performance and energy consumption. In order to answer these questions we have calculated the complexity of each protocol and the energy cost it will add to the system. The evaluation of all discussed protocols is presented in a generalized way and can therefore serve as a reference point for future evaluations and for the design of new, improved GKA protocols.**

Index Terms—**Clustering, Group Key Agreement Protocols, Wireless Sensor Networks, Cryptography.**

I. Introduction

DURING the last years we have witnessed a wide spread in the use of wireless sensors for various applications. Usually a number of sensors are employed to form a Wireless Sensor Network (WSN). These sensors can be placed in or scattered over a specific area of interest (a house, a forest, a battlefield etc.) in order to monitor critical parameters of their application environment. They can also be implanted in a person's body (e.g., a patient) to track vital signs, personal performance data etc. The information collected by the sensors is periodically transmitted to a central Base Station (BS), either via direct (one-hop) communication or via multi-hop routing, where data is forwarded through intermediate nodes towards the BS or a gateway. The small size of the wireless sensors and low manufacturing cost enables the use of multiple sensors which leads to more accurate data. The common characteristic among all these applications is the fact that all of them need to confront the limitations imposed by the WSNs. These limitations are caused by: a) the inherent limited computation capabilities and the short life of battery-powered sensors, and b) the large number of the nodes employed in WSNs, the lack of a specific network architecture or infrastructure and the frequent topology changes due to nodes mobility. To overcome these issues, a great emphasis is placed on scalability and efficient resource management.

Currently, one of the proposed architectures for efficient resource management of wireless networks is clustering. Clustering is ideal for large-scale environments and time-critical applications compared to the multi-hop model and can be particularly efficient in one-to-many, many-to-one, one-to-any and one-to-all fashioned communication. The use of cluster-based approaches optimizes network bandwidth and service discovery while addressing the needs for scalability at the same time [1].

Another issue that affects the use of WSNs is security. By nature WSNs are vulnerable to a number of threats already identified in several works, e.g. [2]-[4]. Every security solution adopted for the protection against these threats requires the employment and management of cryptographic keys. This implies that effective key management mechanisms must be employed and that efficient ways for the distribution and management of these keys should be established. Work in [5] outlines the basic characteristics of the various secret key schemes, in relation to the basic security requirements for WSNs. It also reviews and evaluates five representative protocols for each keying model, in terms of some general performance requirements, like simplicity, scalability, robustness and storage efficiency. However, without considering the specific requirements of the application or the environment that WSN technology is used, we can only be led to general conclusions for the evaluation of the protocols. The authors of [5] also state that beyond the fact that the use of a group key is necessary for multicast communication, cluster-based group keying schemes are more robust than network-wide keys because the compromise of a node, will lead to the compromise of the cluster, but not the entire group. Moreover, these schemes are more scalable since additions and removals of nodes are managed in a more efficient way.

Work in [6] proposes two authenticated cluster-based group key agreement protocols and reviews the main characteristics of only four well known cluster-based GKA protocols. However, the performance of all these protocols is not evaluated. The authors in [7] present a survey of key management schemes in WSNs. This paper also outlines that key distribution techniques are not suitable for many WSN applications. The paper describes the basic key management schemes and overviews the main characteristics as well as the limitations and security threats of each scheme. Work in [8] analyses the security issues and requirements of WSNs. Authors describe the most important WSN applications and scenarios and emphasize on the specific requirements for each application. The paper also proposes a security framework and architecture to integrate existing technologies with WSNs in order to provide secure and private communications to its users.

The authors are with the Laboratory of Information and Communication Systems Security, Department of Information and Communication Systems Engineering, University of the Aegean, Karlovassi, GR-83200 Samos, Greece (e-mail: eklad, ekonstantinou, gkamb, sgritz@aegean.gr).

The basic conclusion from our research in the literature is that the evaluation of cluster-based GKA protocols is application dependent. In general, WSNs can be deployed in environments in which a basic infrastructure can be expected, or ad hoc in infrastructureless environments. Based on this fact, we can group the WSNs applications into two major categories: infrastructure-based and infrastructureless. A very representative case of such environments and one of the most promising contributions of WSNs is their adoption in the healthcare sector. In particular, the motivation behind this survey came from this area of interest.

Our Motivation (Applications of WSNs in the Healthcare Sector)

WSNs can be deployed in several medical environments, like intra-hospital or medical emergencies and their use can significantly improve the quality of medical care provided and facilitate patients' every day living. However, the adoption of WSNs in healthcare realms also introduces many security issues and challenges mostly because medical services and the associated to them information are considered particularly sensitive. Without doubt, every information system deployed in medical premises must comply with the following security requirements: confidentiality, integrity, availability, authentication, privacy, non-repudiation, authorization and accountability. The mechanisms employed in medical WSNs should also consider patients' mobility without compromising the needs for security and efficiency. Here, also, the use of cluster-based architecture for both the network topology and the key establishment protocols can address these needs.

For this reason, in our previous work [9] we have proposed a general framework for cluster-based medical environments. The framework is comprised of two different scenarios based on the nature of the medical environment. Scenario I copes with medical environments which have a fixed infrastructure while scenario II considers infrastructureless environments. This framework is able to serve almost every need for the provision of modern sensor-oriented medical services. This means that it can fully cover the varying needs of both intra-hospital environments and environments formed ad hoc for medical emergencies but it can be realized for hybrid scenarios as well. In both cases, a number of wireless sensors are implanted on every patient's body in order to collect and transfer real time vital sign data to a central database.

For the first scenario we consider a hierarchical network with Cluster-Heads (CH). This scenario is more suitable for environments where one can afford some powerful nodes, which can play the role of CH, like intra-hospital environments. We can then assume that CHs are fixed and that energy consumption is not a key issue for them. The hospital sensor network can be decomposed to several clusters, based on their geographical location. For example, we can realize one cluster per one or more neighboring patient rooms and one or more clusters for the external area of the hospital. This grouping scheme minimizes frequent topology changes each time a patient roams within the boundaries of her cluster. Clusters' number and size may vary according to the size of the hospital premises, the different units and the number of sensors as well as the number of fixed-nodes or CHs available and their level of wireless coverage. For this scenario we assume that Cluster Members (CM) communicate with their CHs every time they need to transfer data and that communication between each node and the CH is typically one-hop. As a result, the sensors used in this scenario do not need to have special processing capabilities and can be very cheap. The CH collects medical data from all nodes and forwards them towards the central database. Additionally, if necessary, the CH can perform aggregation and filtering of the collected data. This method eliminates even more the amount of data in transit improving resource utilization and conserving network bandwidth.

In our second scenario we discuss the case of networks formed ad hoc, due to the lack of basic infrastructure. We therefore, assume that we cannot rely on any powerful nodes to act as CHs. This architecture is more suitable for medical environments where there is no full coverage or no fixed infrastructure at all, as in the case of a medical emergency. According to this scenario, sensors can be dynamically grouped into overlapping or non-overlapping clusters. Every time a node needs to transmit data, the node closer to the gateway (best path) is selected as the Cluster Leader (CL). The CL can either forward the data directly to the gateway or forward the data via the CLs of adjacent clusters located near the gateway. In order to realize such a scheme the CL must implement a multi-hop routing protocol. Communication between each node and the CL might also be multi-hop. Having in mind that the sensors' location may change very often, the CL responsibility will be automatically assigned to the node located closest to the gateway or to the CLs of neighboring clusters located near the gateway. This means that all nodes should be able to potentially become CLs. As a result, nodes should be more expensive than the ones employed in the first scenario. A detailed description of the proposed framework can be found in [9].

Our Contribution

In this paper, we borrow the main idea of the cluster-based medical framework and specify two basic scenarios for the use of WSNs in various realms. Scenario I embraces the functionality and requirements of applications that need to rely on some type of infrastructure (i.e. wireless access points, gateways, etc). We, therefore, assume that the sensors for this scenario can be organised in a hierarchical structure of the network. Scenario II includes the functionality and requirements of infrastructureless applications of WSNs. In such cases, WSNs are formed ad hoc and thus they cannot always rely on the presence of gateways and routers or any other infrastructure. We then examine the most important cluster-based GKA protocols and discuss which of them can be custom-tailored and thus be profitable for each scenario. Our analysis is driven by two basic aims: a) identify the protocols that fit best in each scenario, and b) assess and measure the efficiency of each protocol in terms of its energy consumption. In particular, we have calculated the energy cost of every protocol, based on its computation and communication complexity.

The challenging part of this study is that each protocol is based on a specific structure and thus its complexity has been

calculated for this structure per se. Nevertheless, we have managed to produce generalized results for every protocol even though we often had to re-calculate their complexity according to some general assumptions. We therefore believe that this evaluation can serve as a reference point for selecting the suitable mechanism for each application and for the design of new, improved GKA protocols. To the best of our knowledge no similar survey for cluster-based GKA protocols exists in the literature so far. The rest of this paper is organized as follows: Section II analyzes the existing cluster-based GKA mechanisms and describes how they can be applied in the two main architectures described as 'scenario I' and 'scenario II'. Section III presents the performance evaluation of these protocols in a comparative way while Section IV concludes the paper.

II. CLUSTER-BASED GROUP KEY AGREEMENT PROTOCOLS

As already mentioned in the previous section, a group key management scheme is usually needed in applications where a number of intermediate nodes participate in the data path, for secure routing and packet forwarding. GKA protocols are considered to be more efficient than pairwise key establishment schemes for WSNs because devices do not waste energy every time they wish to communicate with another device by establishing a new shared secret key. Most of the traditional group key agreement protocols reported in the literature cannot cope with the dynamic nature and limitations of wireless ad hoc networks. In particular, the well known protocols appeared in [10], [11], [12] are efficient for wired networks but they cannot be directly applied to ad hoc wireless networks and especially the highly dynamic ones. However, by organizing the nodes of the network hierarchically based on their relative proximity to one another and allowing the formation of small subgroups, this situation can change. It has been proved in several works, such as [13], [14] that clustering can improve the performance of traditional GKA protocols.

The majority of cluster-based key agreement schemes [6], [13]-[22], assume a specific hierarchical structure of the clusters or some tree-structure and then apply a general key agreement protocol like the two-party Diffie-Hellman protocol [23] or the GKA protocol of Burmester and Desmedt (BD) [10], or a variation of them. The GKA protocol is first applied locally in every cluster and then, the clusters' keys are used from the same or another key agreement mechanism to form the final group key. Upon that, this group-key can be used from all the members of the network to provide confidentiality. An exception to this approach are the protocols described in [15] and [22] which do not end up with the creation of a secret group key for the whole group. This is done for communication efficiency and memory storage purposes. These two protocols are completed as soon as all the nodes have created a cluster key. In this case, for the inter-cluster communication between two nodes, the corresponding CHs which share common keys with other CHs must decrypt and re-encrypt the messages they relay using the corresponding cluster-keys.

From the aforementioned cluster-based GKA protocols, only [6], [13], [18] and [19] provide authentication. Authentication ensures that only valid group members participate in the key setup phase and therefore provides a way for protection against man-in-the-middle attacks during the key agreement phase. In works [15], [16], [17] and [20] the authors propose a way for turning their protocol into an authenticated one, but they do not specifically analyze the additional communication and computation cost introduced in the protocol by authenticating every message. Finally, protocols [14] and [22] do not consider a specific authentication mechanism at all. However, these protocols can be modified to provide authentication with the use of either an authenticated GKA protocol or a special compiler like the one described in [24].

In the next subsections we examine the main features of the aforementioned cluster-based GKA protocols. Moreover, we analyze the way they can be applied in WSNs environments that fall into one of the two major architectures (infrastructure-based and infrastructureless). We also evaluate the complexity of each solution in a comparative way. We are particularly interested in authenticated GKA protocols. However, as already pointed out, many of the protocols do not provide authentication in their primitive form. In order to be able to include these protocols in our evaluation we present two tables for the complexity analysis, i.e., one for authenticated and one for unauthenticated protocols.

The complexity analysis of the protocols comprises the calculation of the total number of computations performed by every protocol and the number of messages exchanged by each of them. A detailed complexity analysis can also include the message size and the memory requirements (i.e., the keys that need to be stored in every node). In our study we will focus on the computation and communication complexity of each protocol. In particular, for the computation cost we will only take into consideration the number of modular exponentiations, scalar multiplications and the pairings performed in each case. We assume that other calculations like hash functions, signatures and verification are much less energy consuming tasks compared to heavy public key calculations. Moreover, for each GKA protocol we use its elliptic curve analog.

An elliptic curve defined over a finite field F_p, where $p > 3$ and prime, is the set of points $(x, y) \in F_p$ (represented by affine coordinates) which satisfy the equation $y^2 = x^3 + ax + b$ and $(a, b) \in F_p$ are such that $4a^3 + 27b^2 \neq 0$. The set of solutions (x, y) together with a point O, called the point at infinity, and a special addition operation define an Abelian group. The point O acts as the identity element (for details on how the addition is defined see [25], [26]. Besides the addition, another basic operation on the elliptic curve group is scalar multiplication. Scalar multiplication is the multiplication of an elliptic curve point with an integer. The result is also a point on the elliptic curve. The interested reader may find additional information on the theory of elliptic curves in [25], [26].

The well known Discrete Logarithm Problem (DLP) defined over F_p can also be applied on the elliptic curve group. The main difference is that the exponentiation in F_p is replaced by a scalar multiplication on the elliptic curve group. Thus, when we say that we use the elliptic curve analog of a group key agreement scheme, we simply replace the modular exponentiations in the protocol with scalar multiplications.

The communication cost refers to the number of messages transmitted and received by each entity of the group. For

the calculation of the broadcast messages, we assume that each broadcast message corresponds to the transmission of one message. The difference is that broadcast messages are received by the whole group, which is either the local cluster, or the whole group, depending on the protocol. In most cases, the complexity of the GKA protocol is already given by its authors. However, quite often, we had to do some extra calculations, especially if the complexity analysis was not complete or in accordance with our assumptions. For example, authors often calculate the communication cost based on the number of rounds required by the protocol, but in our analysis we are interested on the number of transmitted and received messages.

Finally, every protocol includes the key establishment phase, usually referred as Initial Key Agreement phase and the key maintenance phase usually referred as Group Key Maintenance phase, for the management of group membership changes. We will focus our analysis on the key establishment phase. Actually, this phase follows the cluster-setup phase, where the creation of clusters takes place. All cluster-based GKA protocols assume a specific cluster structure; some of them describe in detail how the cluster formation is done [6], [17], [19]-[21], while others [13], [14], [16], [18] rely on well-known clustering algorithms. We shall assume that the cluster-structure has already been formed (number of layers, grouping into clusters, election of CHs) and thus we will not calculate the overhead introduced by the cluster-setup phase.

A. GKA protocols for Infrastructure-based WSNs (Scenario I)

In order to examine which GKA protocols are suitable for WSNs applications that have the characteristics of scenario I, we should take into consideration that the clusters in this case form a hierarchical structure in the network. GKA protocols that are best suited for hierarchical networks are [6], [14], [17], [18] and [22]. Here we present the main features of each protocol followed by a complexity analysis, in terms of computational and communicational costs.

In order to have a comparative evaluation of the GKA protocols for scenario I, we can make the following assumptions: We first consider a WSN comprised by: One controller, N nodes and n_c clusters. A Controller (C) denotes the entity in the upper-layer of the hierarchical structure. Moving downwards to the previous layers of the structure, various sub-controllers exist which are actually the CHs. The lowest layer of the structure is comprised by the CMs which are grouped in several clusters. The number of different layers of the hierarchical structure is actually the height h of the structure, whereas the current layer is noted as υ. Finally, we assume that the hierarchical network has only three layers, which means that the height of the structure is equal to 3. Then, the controller is in level $\upsilon = 0$, the cluster heads CH_j are in level $\upsilon = 1$ and the cluster members $CM_{i,j}$ are in level $\upsilon = 2$. The symbol n_j signifies the number of members in the *j-th* cluster (or subgroup) and symbols CH_j and $CM_{i,j}$ signify the CH of the *j-th* cluster and the *i-th* CM of *j-th* cluster, accordingly. In our study, we consider clusters with equal size.

It is obvious that the number of layers as well as some restrictions in the structure of the clusters play a significant role in the complexity cost of the protocol. Consequently, if we change our assumptions for the group structure, the communication and computational cost of the protocols will be parameterized by the number of layers and the cluster size. However, as it was stated in [17] and [19], the best performance can be achieved when the hierarchical structure becomes fully balanced, i.e., all the clusters have equal size. Moreover, the authors of [17] claimed that the efficiency of the whole protocol is improved when the number of levels is small (e.g., equal to 3). Fig. 1 illustrates a typical model for the hierarchical structure of GKA protocols for scenario I.

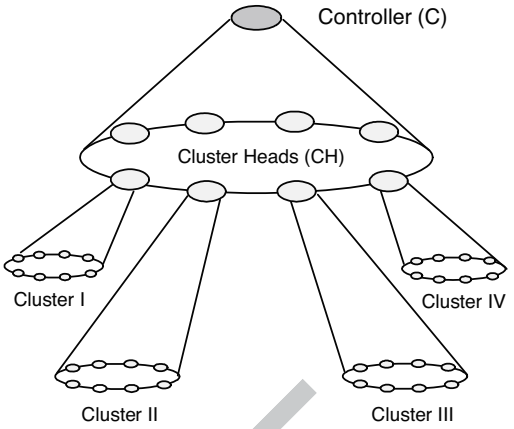

Fig. 1. A typical model of the hierarchical structure of scenario I

It should be noted that in this section we calculate the computation and communication cost for each entity of the group and not the total number of these metrics performed by the whole group.

HKAP Protocol

The Hierarchical Key Agreement Protocol (HKAP) proposed by Yao et al. in 2003 [14] uses a cluster-based hierarchical structure of mobile nodes and then applies some well known GKA protocols in every cluster. The protocol first groups the nodes into clusters and applies an existing GKA protocol to the members of each cluster in order to generate a cluster key. Then, a GKA protocol is applied to all CHs to generate the group key. Finally, the group key is distributed to all the group members with the use of a key distribution protocol. The protocol assumes a hierarchical structure of the network where all nodes are grouped into one-hop clusters. The CHs of each cluster can communicate with each other using a more powerful radio transmission, and thus it is assumed that they form a backbone network. Multilevel backbone networks can be formed recursively in the same way. The protocol comprises of three main phases. In phase 1, the nodes are organized in a hierarchical cluster-based structure consisted of *h*-levels. It should be noted that clustering is based on geographical relationship between the mobile nodes. In phase 2, each member chooses a secret key. Then, all members agree on the use of a GKA protocol and execute it to establish the cluster key. In the last phase of the protocol, each CH broadcasts the computed upper keys to all the members of its clusters. The HKAP protocol assumes that nodes' mobility will only affect the group key if there is a

change in the logical topology of the network. The protocol is well suited for scenario I of our framework mainly for two reasons. The most important reason is that the CHs employed by this protocol should be more powerful devices and have the ability to transmit in higher levels links. Secondly, the design of the protocol permits the geographical reorganization of the nodes without triggering a key-refresh procedure every time a node leaves the boundaries of its cluster and joins a new one. Moreover, it is flexible since any GKA protocol can be applied in every cluster. On the other hand, the protocol does not provide authentication mechanisms in its original form.

The computation cost of HKAP consists of the number of scalar multiplications executed by every entity of the group. In particular, every CM executes $(5n_j - 6)/n_j$, the CHs $2(5n_j - 6)/n_j$ and the Controller $3(5n_j - 6)/n_j$ scalar multiplications. The communication cost is calculated by the number of messages that each entity of the group sends and receives for the creation of the group key. More specifically, every CM has to send $(2n_j - 1)/n_j$ messages and receive $1 + (4n_j - 3)/n_j$. The CHs send $1 + 2(2n_j - 1)/n_j$ messages and receive $1 + 2(4n_j - 3)/n_j$ and the Controller sends $1 + 3(2n_j - 1)/n_j$ messages and receives $3(4n_j - 3)/n_j$.

GKA-CH Protocol

The Group Key Agreement protocol for Circular Hierarchical group model (GKA-CH) [17] proposed by Teo and Tan in 2005, applies the Burmester-Desmedt [10] GKA protocol in every layer of a circular-hierarchical group structure. More specifically, the whole group is arranged in h hierarchical layers with each layer having one or more subgroups. Every subgroup is organised in a circle, contains an equal number of members and is managed by a subgroup controller. The subgroup members in each layer, from layer L_{h-2} to L_0, are also subgroup controllers of the next layer. Since this protocol assumes an equal number of members in each cluster, it can only be applied in scenario I of our framework. The protocol comprises of four phases. The first three phases specify the procedures for the calculation of the subgroup key for every subgroup, starting from the lowest layer (phase 1) and ending in the highest layer (phase 3). The subgroup key of the upper level is actually the group key K. During the last phase of the protocol, the group key K is encrypted and broadcasted to the lower levels using symmetric key cryptography. Specifically, for the four phases of the protocol we have:

1) In the lowest level, L_{h-1}, the subgroup key of every subgroup is calculated using the Burmester-Desmedt (BD) GKA protocol [10].
2) In the second phase, in the upper layers (layers $L_{(h-2)}$ to L_1), the subgroup key produced by the first phase is used as the random number for the execution of the BD-GKA protocol for the upper layers. In the end of this phase, every subgroup until the first layer has calculated the subgroup key.
3) In the third phase, all the subgroup members of the highest layer L_0, use the subgroup keys produced during phase 2 in order to calculate the subgroup key K, which is actually the final group key. Every member of the subgroup broadcasts the group key K to its members encrypted with its subgroup key.
4) In the last phase of the protocol, every subgroup member decrypts the message from phase 3 to finally get the group key K. The subgroup controllers will have to first decrypt the message and then re-encrypt it using its subgroup key and finally broadcast the key to its subgroup members. This process continues until the members of the lowest layer have obtained the final group key K.

In a three layer structure, every CM has to perform $n_j + 1$ scalar multiplications, transmit 2 messages and receive $n_j + 2$ messages. Each CH has to perform $2(n_j + 1)$ scalar multiplications, transmit 5 messages and receive $2n_j + 3$. Finally, the Controllers will perform $3(n_j + 1)$[1] scalar multiplications, transmit 8 messages and receive $3(n_j + 1)$ messages.

The protocol assumes that a signature scheme is used for authentication but authors do not specify how this can be done. However, this protocol can be easily enhanced to support authentication by applying an authenticated BD protocol instead of the unauthenticated one. However, this process will increase the complexity of the protocol and the size of the exchanged messages.

Alternatively, an easy way to enhance the unauthenticated protocols [14] and [17] to support authentication is to use a special compiler like the one proposed by Katz and Yung in [24]. The compiler actually requires each user to sign every message it sends and verify all the messages it receives. More specifically, the application of this compiler will add to the protocols an extra round and also a number of signature and verification calculations that is equal to the number of the messages exchanged by each protocol. The compiler also introduces an extra overhead in the storage requirements of each node, since every one must store a (very large in size) nonce for all the nodes of the group.

PB-GKA-HGM Protocol

Another protocol, proposed in 2007 also by Teo and Tan, is a password-based GKA protocol for hierarchical group models (PB-GKA-HGM) [18]. This protocol creates a hierarchical structure based on three main entities: the main controller C in highest layer, various subgroup controllers (S_i) and several members (M) in every subgroup. The establishment of the common group key is then performed in three phases. During Phase 1, each S_i interacts with the subgroup members to compute the subgroup key K_i. In Phase 2, S_i interacts with the controller C to obtain the final group key K. Finally, in Phase 3, the group key K is sent downward securely by the controller to the subgroup controllers which in turn are responsible to securely forward the K to their members. Key confirmation messages are also sent along to verify and confirm the subgroup key K_i and final group key K. The protocol is password-based, which means that for the computation of the subgroup key K_i a password and a pairwise secret key shared between the subgroup members and the subgroup controller is used. Likewise, for the computation of the group key a password and a pairwise secret key shared between the subgroup controller and the main controller is used. The protocol assumes that both the pairwise secret keys and the passwords are securely

[1]Protocol GKA-CH assumes an equal number of members, so $n_c = n_j$

pre-loaded into the devices. Clearly, this mechanism can fit into scenario I of the proposed framework, if we consider a hierarchical WSN with several nodes grouped into hierarchical clusters, with the CHs acting as the subgroup controllers and the BS acting as the main controller.

Concerning the complexity of the protocol, we have calculated that in a three layer structure every CM has to perform 3 scalar multiplications, transmit 1 message and receive 2 messages. Each CH has to perform $2n_j + 5$ scalar multiplications, transmit 3 messages and receive $n_j + 2$. Finally, the Controller will perform $2n_c+2$ scalar multiplications, transmit 2 messages and receive n_c messages.

AP-1 and AP-2 Protocols

Dutta and Dowling proposed in 2009 two cluster-based GKA protocols, AP-1 and AP-2 [6]. AP-1 is based on the constant round multi-party dynamic key agreement protocol DB [27] whereas AP-2 uses the pairing-based group key agreement protocol DBS [28] and assumes that the CHs are arranged in a tree-structure.

Both protocols assume that a group of nodes is organized in a number of clusters according to their relative proximity to one another and perform a GKA protocol to generate a cluster key. Then, a sponsor is elected from every cluster to participate in the generation of the Session Key which is actually the secret group key (e.g., a sponsor in this protocol is actually the CH). The authors assume that sponsors are able to communicate with each other in a single hop. This assumption can only be realized in the case of WSNs with infrastructure, thus these protocols are suitable for Scenario I.

The protocols are comprised by two main phases: the Initialization Key Agreement phase (IKA) which specifies the procedures for the establishment of the common group key and the Group Key Maintenance Phase (GKM) which specifies the procedures for membership changes (like JOIN, LEAVE, etc). The IKA phase of the AP-1 protocol is performed in two steps. In Step 1, the DB protocol is executed in every cluster and a secret cluster is constructed. In Step 2, the DB [27] protocol is executed by the Sponsors of every cluster. The result of this step is the group key. The DB protocol is a variation of the Burmester-Desmedt (BD) protocol [10] and requires 2 rounds and 2 broadcast messages for its completion. The authors assume that each message broadcasted by the sponsors is received by all the other sponsors and also by their respective CMs. Thus, by the end of the IKA phase, every CM has all the necessary information for the calculation of the group key. Since the DB protocol is invoked twice, the complexity of AP-1 is twice the complexity of the DB protocol. This means that every CM has to perform 3 scalar multiplications, send 2 broadcasts and receive $2(n_j - 1)$ messages. The nodes that are elected as sponsors will perform additionally 3 scalar multiplications, send 2 more broadcasts and receive in total $2(n_j - 1) + 2(n_c - 1)$ messages.

AP-2 is a variation of AP-1. Specifically, AP-2 assumes that sponsors are arranged in a tree structure, which enables efficient handling of dynamic membership changes (such as join/leave procedures). AP-2 invokes the DB protocol for Step 1 of the IKA phase, i.e., the creation of the cluster key for each cluster, but for the tree-structure of the sponsors in Step 2 the DBS protocol is invoked. The fact that the GKA protocol applied in AP-2 is pairing-based[2] increases the computation cost and the total complexity of the protocol. We will therefore include only AP-1 protocol in the performance analysis.

For the authenticated versions of the two protocols, AP-1 and AP-2, a variation of the compiler proposed by Katz and Yung [24] is used.

Other Protocols

The protocol described in [22] groups the nodes into non-overlapping clusters and then applies symmetric key encryption for the intra-cluster communication and asymmetric encryption for the inter-cluster communication. The calculation of the shared keys is based on intermediate keys called partial keys, which are created during the initialization phase with the use of the GDH.2 [12] protocol. Each member in a cluster can use these partial keys to compute the symmetric key it shares with other nodes in the same cluster. Note that the symmetric shared keys are not explicitly exchanged between a pair of nodes. For the asymmetric encryption, all nodes create a public/private key pair based on the RSA algorithm. A Certification Authority (CA) is also required for the asymmetric encryption. The protocol proposes a decentralized CA based on threshold cryptography. It should be noted that the use of this protocol does not lead to the creation of a common group key. Clearly, this approach introduces a great restriction to the applicability of this protocol in our scenarios. However, taking into consideration that inter-cluster communication usually occurs in infrastructureless environments, where nodes use multi-hoping mechanisms to relay data towards the gateway or the BS (like the ones described in scenario II), we can assume that the protocol can be applied in special cases of scenario I of our framework. Nevertheless, because of this particularity, it is not included in our evaluation.

As stated earlier, protocol HKAP [14] supports any GKA protocol that the group members agree upon. This is actually the main difference between HKAP [14] and GKA-CH [17]. The first is more generic, whereas the latter is based on the BD protocol. Moreover, the GKA-CH protocol assumes an equal number of nodes in every cluster. Here, for our evaluation we will present the complexity analysis of HKAP with the use of the GDH.3 protocol [12].

Table I shows a comparative analysis of the four aforementioned protocols, in terms of computational and communication costs. The analysis shows that protocols PB-GKA-HGM [18] and AP-1 [6] are more lightweight for all CMs. This can be very important if we consider that in networks with the characteristics of scenario I, CMs are the only nodes that may have power limitations. Protocols GKA-CH [17] and HKAP [14] have similar functionalities. Nevertheless, if we apply a different GKA scheme for HKAP than the BD which is also used by GKA-CH, their performance will vary. Table I shows the complexity of protocol HKAP based on the use of the GDH.3 [12] protocol. Table V in the Appendix Section summarizes the main features of the examined GKA protocols for this Scenario.

[2]The interested reader may refer to [29] and [30] for more details on pairing-based cryptography.

TABLE I
COMPLEXITY ANALYSIS OF PROTOCOLS FOR SCENARIO I

Protocol		Scalar Multiplications (SM)	Messages Sent	Messages Received
HKAP [14]	C	$3\frac{5n_j-6}{n_j}$	$3\frac{2n_j-1}{n_j}+1$	$3\frac{4n_j-3}{n_j}$
	CH_j	$2\frac{5n_j-6}{n_j}$	$2\frac{2n_j-1}{n_j}+1$	$2\frac{4n_j-3}{n_j}+1$
	$CM_{i,j}$	$\frac{5n_j-6}{n_j}$	$\frac{2n_j-1}{n_j}$	$\frac{4n_j-3}{n_j}+1$
GKA-CH [17]	C	$3(n_j+1)$	8	$3(n_j+1)$
	CH_j	$2(n_j+1)$	5	$2n_j+3$
	$CM_{i,j}$	n_j+1	2	n_j+2
PB-GKA-HGM [18]	C	$2n_c+2$	2	n_c
	CH_j	$2n_j+5$	3	n_j+2
	$CM_{i,j}$	3	1	2
AP-1 [6]	CH_j	6	4	$2(n_j-1)+2(n_c-1)$
	$CM_{i,j}$	3	2	$2(n_j-1)$

B. GKA protocols for Infrastructureless WSNs (Scenario II)

The protocols that seem to be well suited for our second scenario are those described in [13], [15], [16], [19], [20] and [21]. The reason is that all these protocols rely on some tree-structure among the CHs which enables them to easily build dynamic networks and manage join and leave events. Here, we present the main features and a complexity analysis of these protocols.

In order to evaluate the protocols, we need to make a number of assumptions. So, without loss of generality, for all the protocols we assume a well balanced tree structure, where clusters have the same cluster size (i.e., number of cluster members) and that communication between cluster members is one-hop. The symbols N and n_j signify the number of nodes of the whole group and the number of members in the *j-th* cluster (or subgroup), accordingly. Fig. 2 illustrates a typical model for the structure of GKA protocols for scenario II for non-overlapping and overlapping clusters.

CEKA-ACEKA Protocol

The protocol proposed by Shi et al. in 2006 [13] is a hierarchical GKA protocol that is based on Diffie-Hellman [23] and on Joux's tripartite GKA [32] scheme. Authentication is provided with an ID-based signature scheme based on bilinear pairings to authenticate communication messages. The paper proposes two different versions of GKA, the Communication Efficient Key Agreement (CEKA) and ACEKA which is the authenticated version of CEKA. A group of nodes is divided in several cluster key trees (where a node in the same subgroup is put on the same level of the cluster key tree) and the backbone key tree which combines all the cluster key trees. In a cluster key tree, the root node represents the cluster key. The protocol uses virtual nodes to successfully construct the tree structure. The protocol assumes that a hierarchical routing protocol is employed and that the nodes in the same cluster are at one-hop distance with the CH. In this case, each pair of nodes in the same cluster is able to communicate in two hops. The protocol comprises of 2 phases, the Initial Key Agreement

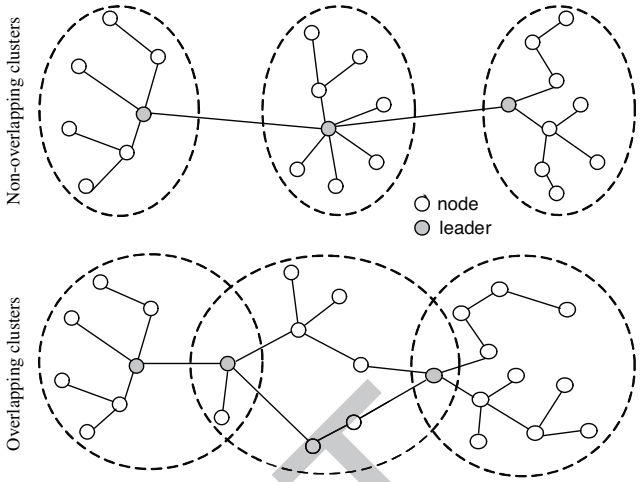

Fig. 2. Typical model of GKA protocols structure for scenario II (non-overlapping and overlapping clusters)

(IKA) phase for the establishment of the initial group session key and the Group Key Maintenance (GKM) phase for the management of dynamic events like member join, leave and refresh. The protocol assumes that if the nodes' mobility does not change the cluster tree structure, there is no need to update the group key. We will only examine the procedures for the establishment of the shared group key, so we will focus on the description of the IKA phase. In this phase, we can distinguish two different procedures for the establishment of the group key. The first procedure, namely IKA-ClusterKeyAgreement specifies the algorithm for the establishment of a cluster key in every cluster. The second procedure, namely Group Key Agreement, specifies the algorithm for the establishment of the final group key K. First off, all nodes are grouped into subgroups of 2 or 3 members each. Then, all nodes construct in parallel their cluster key according to the IKA-ClusterKeyAgreement procedure. The algorithm used for the cluster key agreement is the Diffie-Hellman key exchange protocol if the subgroup members are 2 or the Joux's tripartite key agreement protocol if the subgroup members are 3. By the end of the IKA-ClusterKeyAgreement procedure, a cluster key is generated in every cluster.

During the Group Key Agreement procedure each cluster selects a sponsor. The sponsors become members of a virtual cluster. In this way, every cluster is regarded as a virtual network node and the whole network is regarded as a virtual cluster. Then the IKA-ClusterKeyAgreement procedure is executed in the virtual cluster. The result of this procedure is the cluster key of the virtual cluster which is actually the group key K.

The authenticated version of the protocol referred to as ACEKA, is based on an ID-based signature scheme. Usually, every node has its unique identification code, such as IP or MAC address which can serve as node identity. An alternative approach is to use universally unique identities while manufacturing each node. Then, an ID-based signature scheme can be used to digitally sign the messages exchanged by the protocol.

For the complexity cost of the protocol we can assume that every node who executes Joux's protocol sends a broadcast message and receives 2 messages. According to the unauthenticated version of the protocol, the total number of

computation includes $3(N/2)+3$ scalar multiplications and an equal number of pairing computations. For the communication cost of the protocol, the nodes send $3(N/2)+3$ messages and receive in total $3N+6$. The authenticated version of the protocol introduces a variation only in the computation cost. More specifically, the computation cost includes $15(N/2)+18$ scalar multiplications and $21(N/2+1)$ pairing computations.

CGDH Protocol

The cluster-based GKA protocol proposed by Abdel-Hafez et al. in 2006 [21] (CGDH) groups the nodes into clusters of equal sizes in a hierarchical and well balanced structure of l levels. The value of l depends on the size of the clusters and on the total of number of nodes in the group. The number of clusters is reduced systematically from the base level to the top level which means that the top level will have only one cluster. Each level has a cluster-representative, which are then grouped together to form a new cluster in the upper level. According to the functionality of the protocol, the group key and the intermediate clusters' keys are not transmitted to the network. Therefore, for the communication between the cluster members, the representatives will have to create and broadcast another key to the members or to apply a strong hash function to every cluster key. The protocol comprises of two main phases: the initial members distribution phase and the initial key generation one. After the initial member distribution phase has been completed, we assume that the nodes are grouped into equal-sized clusters in a structure that the protocol describes. During the initial session key generation phase, every cluster uses the GDH.2 [12] protocol to create the cluster-session-key. This procedure is performed by all clusters in each level, starting from the lowest level of the structure. Once the cluster key of the lowest level is created, the same procedure is performed in the upper level. Only then the cluster members will send their public share to the next member in the cluster using the lower-level-cluster key, instead of using a random value. Additionally, the cluster representative of every cluster broadcasts a value to the members of the lower level which they will use in order to calculate the group key. The creation of each cluster-session key takes n_j rounds whereas the total duration of the protocol is ln_j (equals the time it takes for the creation of a cluster session key multiplied with the l levels of the structure).

For the complexity of this protocol we assume that the nodes are grouped into 3 binary trees and that the size of each cluster is 3. This means that the height of the tree will be $log_2(N/3)$ and the number of the layers l will be equal to $log_2(N/3)+1$. For the number of clusters we have $n_c = N-2/3$. Based on this assumption, the computation cost of the protocol is: $\frac{n_j^{l-1}}{n_j-1}[n_j\frac{n_j+1}{2}+1]$.

The total number of transmitted messages is $n_j * n_c$, where $n_c = (N-1)/2$ and the total number of received messages is the total number of transmitted messages plus $(l-1)N-n_c+1$.

A-DTGKA Protocol

The protocol in [19], proposed in 2007 by Abdel-Hafez et al., is an authenticated group-key agreement protocol for ad hoc networks (A-DTGKA) and it is based on hierarchical authentication. The protocol is ID-based and uses pairwise keys for entity authentication. It comprises of two phases: (1) organization of the nodes into clusters, and (2) generation of the group session key. For the needs or our study we will focus on the latter phase. The protocol assumes that each node is equipped with a secret Group Identity Key (KIG), a one-way hash function H and a local identifier (ID). Also, every node is able to compute its weight. This is a number that expresses the node's current status in terms of node's mobility, battery power level, distance from the other nodes, and values related to the surrounding environment (terrain, temperature, battery power etc). Furthermore, the protocol assumes that node identities are publicly available. A Trusted Authority is also needed during the setup phase to generate the private keys for every node. We can distinguish 3 different phases of the protocol aiming to construct and distribute the final group key.

During the first phase, which is also the setup phase, every node computes a pairwise shared secret key using the secret key and the hash of the ID of the other node. This pairwise secret key is used by each member during the authentication phase. Therefore, this phase does not require communication between members. In the second phase, a mutual authentication procedure takes place between members that belong to the same cluster (inner-cluster authentication). The result of this phase is that every node within the cluster is authenticated with the others and with the CL, and that each cluster holds a cluster session key. During the third phase, the protocol is repeated with the upper level clusters, considering that the CLs that participated in the previous phase are now the children for their one level hierarchically higher CL. The CL in the lower level has to decrypt every message it receives from upper levels to first check the identity of the transmitter and then forward the message to its children (re-)encrypted using the cluster session key it shares with them. This procedure is repeated until the root level is reached. This mechanism blends with scenario II of our framework, if we consider that every node authenticates its neighbor and the neighbor its next node and so on until they reach the CL. This is useful for our scenario since communication between the CM and the CL might not always be one-hop. Due to the infrastructureless form of the WSN in scenario II, the frequent changes of the cluster structure do not introduce significant additional cost, since every node that joins another cluster has only to obtain from the CL the local parameters of the new cluster and the cluster session key encrypted with the global group key. On the other hand, the number of rounds increases along with the number of hierarchical levels.

For the complexity of this protocol, we assume that each cluster consists of 4 nodes (3 CM and 1 CL). This is an assumption the authors also make, while describing their scheme. In that case, each node will perform an average of 3 scalar multiplications and 3 pairings. The computation cost of the protocol includes $5N+1$ scalar multiplications and an equal number of pairing computations. The total number of transmitted messages is $(13N+7)/3$ and the total number of received messages is $(19N+2)/3$.

CBGKA - ACBGKA Protocol

The protocol proposed by Konstantinou in 2008 [20] is a cluster-based GKA protocol (CBGKA) based on Joux's

tripartite key agreement protocol [32] and comes in two versions, namely contributory and non-contributory. A protocol is contributory when each member contributes its part to the global group key; otherwise it is referred as non-contributory. Although in the current version of the protocol nodes are not authenticated during the key agreement phase, authors state that the protocol can easily provide members' authentication, by substituting Joux's tripartite key agreement protocol with an authenticated version of it, like those described in [33] or [34]. For the needs of our study we will examine an authenticated version of this protocol, using the ID-based tripartite authenticated key agreement protocol from [34]. The protocol assumes clusters consisted of either two or three members each. The authenticated version of the protocol also assumes that a Key Generation Center (KGC) exists and participates in the generation of the public/private keys for each member. However, this operation is held once, during the setup phase and thus can be considered as offline. The protocol is consisted of 3 main phases and a setup phase. During the setup phase every node has to obtain a long term private key from the KGC. In order to do so, every node sends its long-term public key to the KGC. This key is calculated based on node's identity. The KGC calculates and sends back the private key. During the first phase of the authentication procedure every member calculates two scalar multiplications and sends the resulting points towards the other members of its cluster. Nodes in the lower levels belong to only one cluster, but nodes in upper levels belong to two clusters and therefore will have to send these points to a greater number of nodes (actually this number depends on the size of the clusters the nodes belong to). During the second phase an AuthCreateClusterKey procedure is executed simultaneously in every cluster. Every member first verifies the other members of the same cluster and if verification succeeds, each member calculates the common secret key $K_{cluster}$. By the end of this phase, every leaf shares a secret key with the nodes of its cluster and the rest of the group members (intermediate nodes) share two secret keys: one key with the nodes of the upper level cluster and one with the nodes of the lower level cluster. Finally, in the third phase of the protocol, the root key (K_{root}) is sent downwards to the lower members, one level at a time, by encrypting the key with the session key of every level. This phase takes as many steps as the height of the tallest branch of the cluster-based structure.

The two versions of the protocol, namely authenticated and unauthenticated, have the same communication cost and different computation cost. In particular, the group will send and receive $4N$ messages. The authenticated version of the protocol includes $5N$ scalar multiplications and $(11/2)N$ pairing computations. Finally, for the unauthenticated version the group has to perform N scalar multiplications and $(3/2)N$ pairing computations.

C-AT-GDH Protocol

The protocol proposed by Hietalahti in 2008 [16] is a clique-based GKA protocol (C-AT-GDH) since it assumes that nodes in a cluster are at a one-hop distance from each other. For this network structure authors state that the most efficient GKA protocol is the BD [10]. As a result, the protocol uses the BD protocol for the establishment of the cluster key and the AT-GDH [35] protocol for the establishment of the group key. The protocol comprises of 3 main phases. The nodes are first grouped into cliques (e.g., clusters where each member is able to communicate with every other node in one-hop). Then, the BD protocol is applied in every cluster to finally form a common secret key (which we will call the cluster key). After that, the AT-GDH protocol is applied among the CHs organized in a tree-structure for the establishment of the group key. Once the group key has been formed, the CHs broadcast the last message of the AT-GDH protocol so that every node can calculate the group key. The AT-GDH protocol employs a spanning tree which contains only the one-hop links used in initial key agreement. In the first phase of the protocol every node (in the lowest level of the tree) selects a random secret exponent and sends it to its parent. As soon as every node has received the blinded keys from its children, they select their exponents and form DH type keys with their children repeatedly using the resulting key as the new exponent. The nodes do not send these keys to the children yet. This process continues until the root is reached and the final group key is formed. The protocol does not provide authentication. However, authors believe that the best way to add authentication is to use authentication with ID-based crypto, like ICPK [36] public keys, with key confirmation messages.

The complexity cost of the protocol is the complexity cost of the DB protocol for the leaf nodes (CMs) plus the complexity cost of the AT-TGDH protocol performed by the intermediate nodes (i.e., the number of clusters $n_c = N/n_j$). We also assume that every broadcast message sent in the lowest level is received by (n_j-1) nodes, whereas every broadcast message sent by an intermediate node is also received by (n_j-1) nodes (i.e., the CH sends a message to his CMs). The height of the tree is $h = log_2 n_c$ for a well balanced tree. The computation cost is calculated by the total number of scalar multiplications which is $n_j n_c(n_j+1)+n_c h$. The total number of messages transmitted by the group is $2N + n_c + h^2 - h$, and the total number of messages received by the group is $2N(n_j - 1) + (n_c + h^2 - h)(n_c - 1)$.

The complexity analysis for every protocol described in the previous section is presented in Tables II and III. Protocols [13] and [20] come in two versions, namely authenticated and unauthenticated. Table II shows the complexity cost for the three protocols that are authenticated. The unauthenticated versions of [13] and [20] are presented in Table III with the other unauthenticated protocols. Moreover, the main features of the

TABLE II
COMPLEXITY ANALYSIS OF PROTOCOLS FOR SCENARIO II (AUTHENTICATED)

Protocol	Scalar Multiplications (SM)	Pairings (P)	Messages Sent	Messages Received
ACEKA [13]	$15(N/2)+18$	$21(N/2+1)$	$3(N/2)+3$	$3N+6$
A-DTGKA [19]	$5N+1$	$5N+1$	(13N – 7)/3	$(19N+2)/3$
ACBGKA [20]	$5N$	$(11/2)N$	$4N$	$4N$

TABLE III
COMPLEXITY ANALYSIS OF PROTOCOLS FOR SCENARIO II (UNAUTHENTICATED)

Protocol	Scalar Multiplications (SM)	Pairings (P)	Messages Sent	Messages Received
CEKA [13]	$3(N/2)+3$	$3(N/2)+3$	$3(N/2)+3$	$3N+6$
CGDH [21]	$\frac{n_j^{l-1}}{n_j-1}[n_j\frac{n_j+1}{2}+1]$	-	$n_j * n_c$	$n_j * n_c + (l-1)N - n_c + 1$
CBGKA [20]	N	$(3/2)N$	$4N$	$4N$
C-AT-GDH [16]	$n_j n_c(n_j+1)+n_c h$	-	$2N+n_c+h^2-h$	$2N(n_j-1)+(n_c+h^2-h)(n_c-1)$

TABLE IV
ENERGY COSTS FOR COMPUTATION AND COMMUNICATION

Computation cost of Scalar Mult.	8.8 mJ
Computation cost of Pairings	47.0 mJ
Communication cost for transmitting a message	3.46 mJ
Communication Cost for receiving a message	2.40 mJ

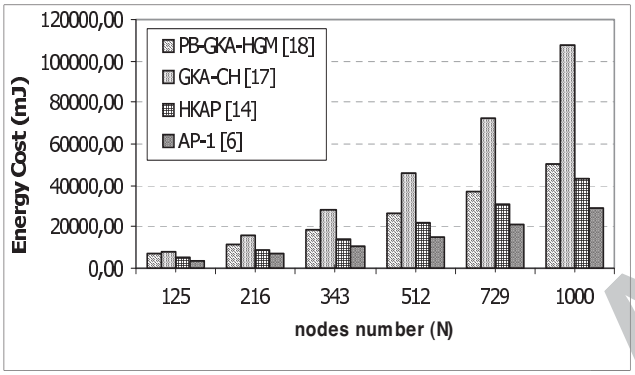

Fig. 3. Computation cost for protocols of Scenario I

examined GKA protocols for this scenario are summarized in Table VI, in the Appendix Section.

III. Performance Evaluation

In the previous sections we analysed several cluster-based GKA protocols for WSNs that appear in the literature and proposed taxonomy based on the underlying network infrastructure. The main issue here is whether these protocols can be implemented by devices with limited-resources such as sensor nodes. A very recent work of Szczechowiak et al. [37] showed that pairings and elliptic curves not only are viable in sensor nodes but in fact attractive. The same conclusion is also derived from [38] where it is shown that non-interactive identity-based key agreement protocols based on pairings provide the best solution in Underwater Wireless Sensor Networks. We therefore believe that all previously mentioned GKA protocols are also viable for WSNs since every node is able to perform ECC scalar multiplications, exponentiations and pairings.

In order to obtain a better estimation for the energy cost of each protocol presented in the previous section, we have calculated its energy consumption for a specific sensor. More specifically, we consider a sensor network comprised by Tmote Sky devices by Texas Instruments with a maximum data rate of 100 Kbps. According to [31] a sensor node built upon the 133MHz StrongARM microprocessor consumes

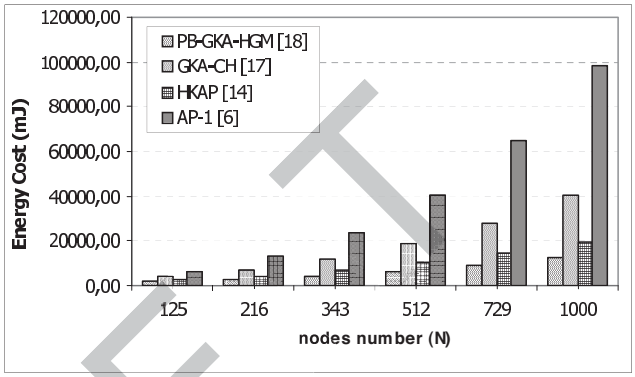

Fig. 4. Communication cost for protocols of Scenario I

8.8mJ for a scalar multiplication and 47.0mJ for a pairing. As for the communication cost, a 100 kbps radio transceiver module consumes 10.8μJ and 7.51μJ for the transmission and reception of one bit of information, respectively. For each GKA protocol we use its elliptic curve analog and thus we assume that the exchanged messages have the size of an elliptic curve point. If we use a 160-bit elliptic curve, the size of its points (x, y) will be 320 bits. We can then calculate the cost for the transmission and reception of a single point by multiplying its size in bits with the energy cost for the transmission and reception of a single bit. Table IV summarizes the energy costs for a scalar multiplication, a pairing computation and a transmission and reception of a message using the particular device (Tmote Sky) and the 100 kbps radio transceiver module.

Based on these metrics as well as on the complexity cost of each protocol already presented in Tables I, II and III we calculate the amount of energy that every protocol needs in order to perform all the needed computation (i.e., scalar multiplications and pairings) and to transmit and receive all the messages exchanged until a group key is created and distributed to all the members of the group. We have also calculated the total amount of energy needed for each protocol (i.e., computation cost + communication cost). These calculations refer to the energy consumption for the whole group of nodes.

A. Performance Analysis for Protocols in Scenario I

Figures 3 and 4 present the amount of energy consumption for the computation and communication cost that each protocol of scenario I imposes, while Figure 5 depicts the total energy cost for each protocol. These results are obtained based on the complexity cost described in Table I, using different values of N. In particular, we considered a group of 125, 216,

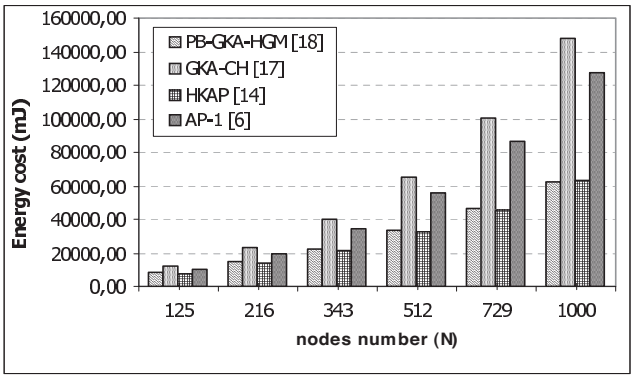

Fig. 5. Total energy cost for protocols of Scenario I

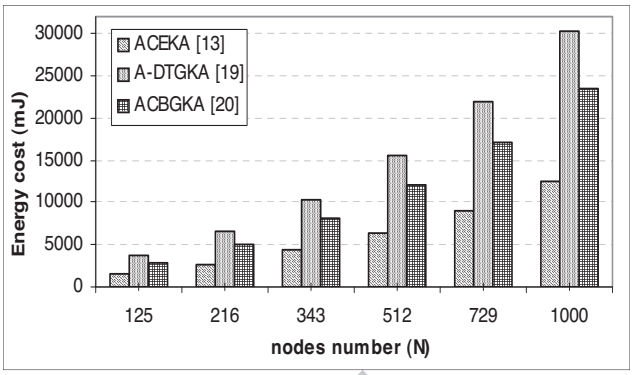

Fig. 7. Communication cost for protocols of Scenario II (with authentication)

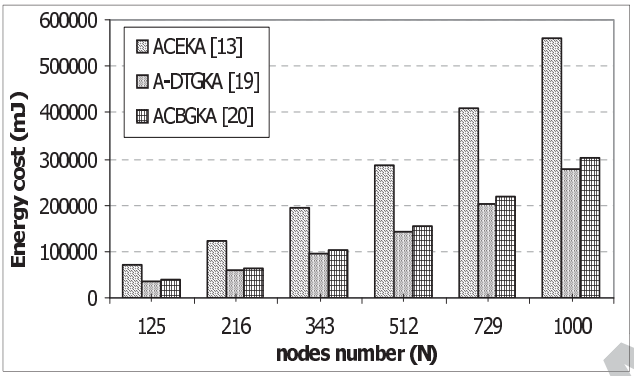

Fig. 6. Computation cost for protocols of Scenario II (with authentication)

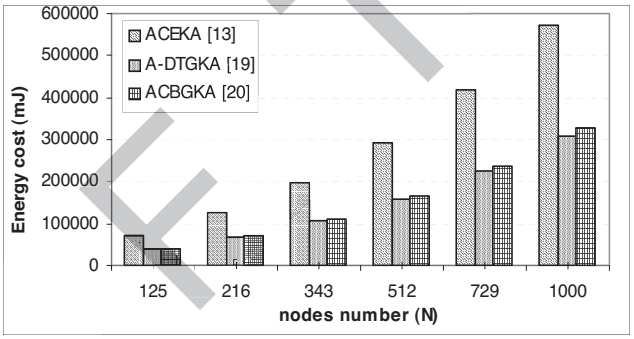

Fig. 8. Total cost for protocols of Scenario II (with authentication)

343, 512, 729 and 1000 nodes and a cluster size of 5, 6, 7, 8, 9 and 10 members accordingly. Given the group size, the cluster size and the height of the network's structure, we can calculate the number of clusters in each protocol.

We notice that protocols GKA-CH and AP-1 do not scale well with the increase of the group size. For a small number of nodes, up to 216, the performance of the four protocols is similar. However, as the number of nodes in the group increases, the lowest energy consumption is achieved by the HKAP protocol followed very closely by the PB-GKA-HGM protocol. Although AP-1 is computationally efficient, its communication complexity causes a high total energy cost. It should also be noted, that despite the similarity of the protocols HKAP and GKA-CH, the application of a different GKA protocol leads us to different results for their performance. GKA-CH applies the BD protocol [10], whereas for protocol HKAP which can be implemented with any GKA protocol, we have assumed the use of GDH.3 protocol [12].

B. Performance Analysis for Protocols in Scenario II

As already stated in section II.B, the protocols in scenario II are grouped into those that provide authentication and those that are unauthenticated. This distinction is also reserved for the performance results which are also obtained based on the complexity cost of each protocol (Tables II and III) for a total population of 125, 216, 343, 512, 729 and 1000 nodes. Wherever there is a need to specify the cluster size (e.g., in protocols [16] and [21]) we consider clusters of 5, 6, 7, 8, 9 and 10 members accordingly. Given the group size and the cluster size we can calculate the height of the tree structure and the number of clusters in each protocol.

Figures 6 and 7 depict the amount of energy required by the computation and communication procedures for each authenticated protocol, while Figure 8 presents the total energy cost for each one of them.

The performance analysis shows that the ACEKA protocol does not scale well with the increase of the group size. This is mainly caused by the increased energy cost of the computation functions. In general, the computation complexity of each protocol is the main factor that affects the total complexity of the protocol. Protocol A-DTGKA has the lowest energy cost, but the cost of ACBGKA is also very close.

Figures 9, 10 illustrate the amount of energy consumption induced by the computation and communication procedures of unauthenticated protocols, while Figure 11 presents the total energy cost for each one of them.

The performance analysis of the unautheticated cluster-based GKA protocols for Scenario II, show that CGDH protocol introduces a high computation cost whereas protocol C-AT-GDH has a high communication cost. Comparing the total complexity of the four protocols, we conclude that the highest energy consumption is caused by protocol CGDH. The most efficient protocols for Scenario II are protocols CEKA and CBGKA.

IV. Conclusion

The main aim of this paper is to examine which of the cluster-based GKA protocols that appear in the literature are

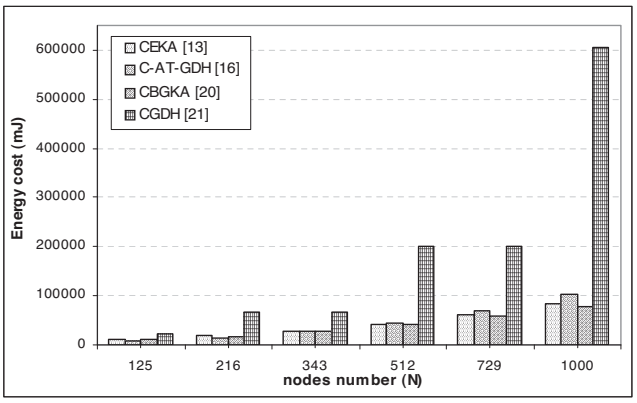

Fig. 9. Computation cost for protocols of Scenario II (no authentication)

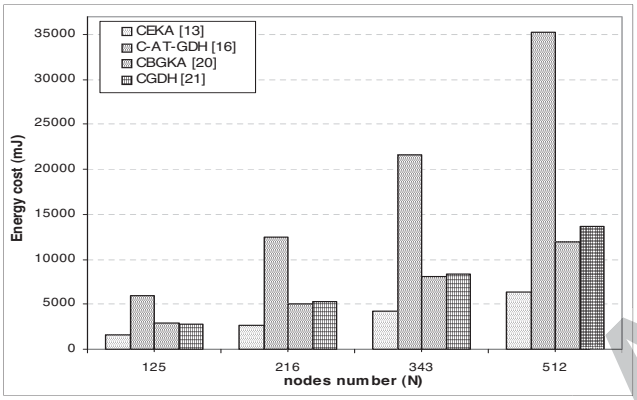

Fig. 10. Communication cost for protocols of Scenario II (no authentication)

applicable to various applications of WSNs. To investigate this issue we have classified WSNs applications into two basic categories or scenarios based on their network architecture. Scenario I deals with the functionality and requirements of applications that need to rely on some type of infrastructure, whereas scenario II deals with infrastructureless environments.

The evaluation of the GKA protocols was driven by two basic questions: a) which cluster-based GKA protocol is suitable for each scenario and b) to which degree these protocols will impact the systems' performance and energy consumption. To answer these questions we had to compute the complexity of each protocol and also the energy cost it will add to the system. This was done by assessing the complexity of each protocol in terms of computation and communication costs. The results reveal which mechanism comprises the best solution for each particular scenario and give an estimation of the scalability of each protocol.

Moreover, since this evaluation is not based on a specific application, the results produced can serve as a reference point for future evaluations and for the design of new, improved cluster-based GKA protocols.

APPENDIX

Tables V and VI summarize the main features of the examined GKA protocols for scenarios I and II.

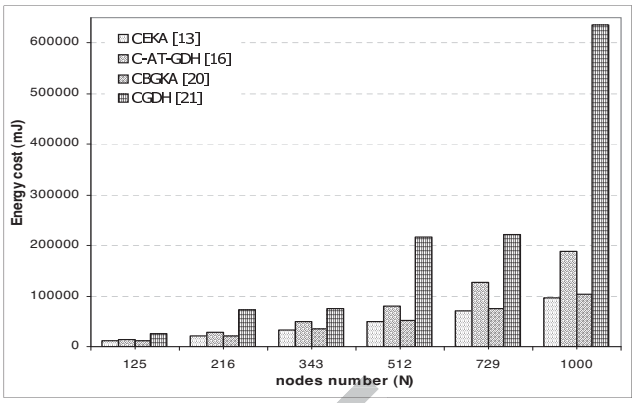

Fig. 11. Total cost for protocols of Scenario II (no authentication)

REFERENCES

[1] M. Dohler, T. Watteyne, F. Valois, J. Lu, "Kumar's, Zipf's and Other Laws: How to Structure a Large-Scale Wireless Network?" Annals of Telecommunications, Vol. 63, No. 5-6, May-June 2008.

[2] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", Ad Hoc Network Journal, special issue on sensor network applications and protocols, Elsevier (2002).

[3] D.R. Raymond and S.F. Midkiff, "Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses", IEEE Pervasive Comput., vol.7 (1), pp.74-81, Jan.-March 2008.

[4] G. Kambourakis, E. Klaoudatou and S. Gritzalis, "Securing Medical Sensor Environments: The Codeblue framework case", in: 2nd International Conference on Availability, Reliability, and Security - 1st International Symposium on Frontiers in Availability, Reliability and Security, pp. 637-643, April 2007, Austria, IEEE CS Press.

[5] J.C. Lee, V.C.M. Leung, K.H. Wong, J. Cao and H.C.B. Chan, "Key Management Issues In Wireless Sensor Networks: Current Proposals And Future Developments", IEEE Wireless Commun., October 2007

[6] R. Dutta and T. Dowling, "Secure and Efficient Group Key Agreements for Cluster Based Network", in Transactions on Computational Science Iv: Special Issue on Security in Computing, Lecture Notes In Computer Science, vol. 5430. Springer-Verlag, Berlin, Heidelberg, 87-116, 2009.

[7] Y. Xiao, V.K. Rayi, B. Sun, X. Du, F. Hu, M. Galloway, "A survey of key management schemes in wireless sensor networks", Computer Communications, Volume 30, Issues 11-12, Special issue on security on wireless ad hoc and sensor networks, 10 September 2007.

[8] N.R. Prasad and M. Alam, "Security Framework for Wireless Sensor Networks", Wireless Personal Communications, vol. 37, no. 3-4, 2006.

[9] E. Klaoudatou, E. Konstantinou, G. Kambourakis, and S. Gritzalis, "Clustering Oriented Architectures in Medical Sensor Environments", In: International Workshop on Security and Privacy in e-Health, pp. 929-934, March 2008, Barcelona, IEEE CS Press.

[10] M. Burmester and Y. Desmedt, "A Secure and Efficient Conference Key Distribution System", in Advances in Cryptology - EUROCRYPT 1994, Lecture Notes in Computer Science Vol. 950 (Springer-Verlag, 1994), pp.275-286.

[11] Y. Kim, A. Perrig, and G. Tsudik, "Tree-based group key agreement", in ACM Trans. Information and Systems Security, vol. 7, no. 1, pp. 60-96, 2004.

[12] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication", in Proc. 3rd ACM Conference on Computer and Communications Security, ACM Press, pp.31-37, 1996.

[13] H. Shi, M. He, and Z. Qin, "Authenticated and Communication Efficient Group Key Agreement for Clustered Ad Hoc Networks", in 5th International Conference on Cryptology and Network Security -CANS 2006, Lecture Notes in Computer Science Vol. 4301, Springer-Verlag, pp. 73-89, 2006.

[14] G. Yao, K. Ren, F. Bao, R.H. Deng, and D. Feng, "Making the Key Agreement Protocol in Mobile Ad Hoc Network More Efficient", in 1st International Conference on Applied Cryptography and Network Security - ACNS 2003, Lecture Notes in Computer Science Vol. 2846, Springer-Verlag, pp. 343-356, 2003.

[15] Y. Chen, M. Zhao, S. Zheng, and Z. Wang, "An Efficient and Secure Group Key Agreement Using in the Group Communication of Mobile

TABLE V
SUMMARY OF THE MAIN FEATURES OF GKA PROTOCOLS FOR SCENARIO I

Protocol	Authentication	Protocol Used	Structure	Limitations and specific characteristics of the Protocol
HKAP [14]	NO	Any GKA protocol	Nodes are grouped into one-hop clusters and form a hierarchical structure. The nodes selected as CHs should be more powerful devices and have the ability to transmit in higher level links	Permits the geographic reorganization of nodes without triggering a key-refresh procedure every time a node moves from one cluster to another
GKA-CH [17]	NO	BD protocol [10] in every layer of a circular-hierarchical group structure	The group is arranged in h hierarchical layers comprised by one or more subgroups organised in a circle	All clusters have the same size
PB-GKA-HGM [18]	YES	A password-based protocol is applied on every cluster	Creates a hierarchical structure based on 3 entities: the main controller C in highest layer, various subgroup controllers S_i and several members (CM) in every subgroup	Every CM should hold a password and a pairwise secret key shared with the subgroup controller. Every subgroup controller should hold a password and a pairwise secret key shared with the Controller. Passwords and secret keys are preloaded into nodes
AP-1 [6]	YES	A variation of the BD protocol, called DB [27] for both cluster and group keys	A sponsor is elected from every cluster to participate in the generation of the final group key	Communication between sponsors should be one hop
AP-2 [6]	YES	The DB protocol for cluster key and DBS [28] for the group key	Sponsors are arranged in a tree structure, which enables efficient handling of dynamic membership changes	The GKA protocol applied in AP-2 is pairing-based. This increases the computation cost and the total complexity of the protocol

TABLE VI
SUMMARY OF THE MAIN FEATURES OF GKA PROTOCOLS FOR SCENARIO II

Protocol	Authentication	Protocol Used	Structure	Limitations and specific characteristics of the Protocol
ACEKA [13]	YES	Diffie-Hellman [23] and Joux's tripartite key agreement protocol [32] for clusters of 2 and 3 members accordingly	• Nodes are grouped into subgroups of 2 or 3 members each • Several cluster key trees. The root node represents the cluster key • A backbone key tree which combines all the cluster key trees. Virtual nodes are used to construct the tree structure	• Nodes in the same cluster are at one-hop distance with the CH • The use of virtual nodes for the construction of the backbone key tree
CDGH [21]	NO	Based on the GDH.2 GKA protocol	Nodes are grouped into clusters of equal size in a hierarchical well balanced structure	The group members do not need to store intermediate keys to generate the session key
A-DTGKA [19]	YES	The protocol is ID-based and uses pairwise keys for entity authentication	Hierarchical tree structure of clusters consisted of neighbor nodes	• Node identities should be publicly available • TA is needed during the setup phase to generate the private keys for every node
ACBGKA [20]	YES	Uses the Joux's [32] tripartite key agreement protocol	Clusters are consisted of either two or three members	A Key Generation Center (KGC) is required for the generation of the public/private keys for each member
C-AT-GDH [16]	NO	The BD protocol is used for the establishment of the cluster key and the AT-GDH [35] protocol for the establishment of the group key	Nodes are grouped into special clusters called cliques	Clusters should have a specific structure (e.g., every CM should communicate in one-hop with all other CMs

Ad-hoc Networks", in International Conference on Computational Intelligence and Security, pp. 1136 – 1142, IEEE Press, 2006.
[16] M. Hietalahti, "A clustering-based group key agreement protocol for ad-hoc networks". Electronic Notes in Theoretical Computer Science, Vol. 192, pp. 43-53, 2008.
[17] J.C.M. Teo and C.H. Tan, "Energy-Efficient and Scalable Group Key Agreement for Large Ad Hoc Networks", in Proc. 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, pp. 114- 121, 2005.
[18] J.C. Teo, and C.H. Tan, "Denial-of-service resilience password-based group key agreement for wireless networks", in Proc. 3rd ACM Workshop on QoS and Security For Wireless and Mobile Networks (Chania, Crete Island, Greece, October 22 - 22, 2007. ACM, New York, NY, 136-143.
[19] A. Abdel-Hafez, A. Miri, and L. Oronzo-Barbosa, "Authenticated Group Key Agreement Protocols for Ad hoc Wireless Networks", International Journal of Network Security, Vol. 4, No. 1, pp. 90-98, 2007.
[20] E. Konstantinou, "Cluster-based Group Key Agreement for Wireless Ad Hoc Networks", in 3rd International Conference on Availability, Reliability and Security - ARES 2008, pp. 550- 557, IEEE Press, 2008.
[21] A. Abdel-Hafez, A. Miri, and L. Oronzo-Barbosa, "Scalable and fault-tolerant key agreement protocol for dynamic groups", International Journal of Network Management, Vol. 16, Issue 3, pp. 185-201, 2006.
[22] A. Balasubramanian, S. Mishra, R. Sridhar, "Analysis of a hybrid key management solution for ad hoc networks", Wireless Communications and Networking Conference, pp. 2082-2087, IEEE Press, 2005.
[23] W. Diffie and M. Hellman, "New directions in cryptography", IEEE Trans. Inf. Theory, 22, pp. 644-654, 1976.
[24] J. Katz and M. Yung, "Scalable protocols for authenticated group key exchange", in Advances in Cryptology - CRYPTO 2003, Lecture Notes in Computer Science Vol. 2729, Springer-Verlag, pp. 110-125, 2003.
[25] I. Blake, G. Seroussi, and N. Smart, "Elliptic curves in cryptography", London Mathematical Society Lecture Note Series 265, Cambridge University Press, 1999.
[26] J. Silverman, "The Arithmetic of Elliptic Curves", Springer Verlag, 1986.
[27] R. Dutta, R. Barua, "Provably Secure Constant Round Contributory Group Key Agreement in Dynamic Setting", IEEE Trans. Inf. Theory 54(5), 2007-2025 (2008).
[28] R. Barua, R. Dutta, and P. Sarkar, "Extending Joux's Protocol to Multi Party Key Agreement", in Progress in Cryptology - INDOCRYPT 2003, Lecture Notes in Computer Science Vol. 2904 (Springer-Verlag, 2003), pp.205-217.
[29] R. Dutta, R. Barua, P. Sarkar, "Pairing-Based Cryptography: A Survey", Cryptology ePrint Archive, Report 2004/064.
[30] R.M. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, F. Vercauteren, "Handbook of Elliptic and Hyperelliptic Curve Cryptography", Chapman & Hall/CRC, 2006.
[31] C.H. Tan and J.C.M. Teo, "Energy-Efficient ID-based Group Key Agreement Protocols for Wireless Networks", in 2nd International Workshop on Security in Systems and Networks - SSN 2006, IEEE Press, 2006.
[32] A. Joux, "A one round protocol for tripartite Diffie-Hellman". In: Algorithmic Number Theory Symposium - ANTS IV, LNCS, Vol. 1838, Springer-Verlag, pp. 385-394, 2000
[33] S.S. Al-Riyami and K.G. Paterson, "Authenticated three party key agreement protocols from pairings", in: 9th IMA International Conference on Cryptography and Coding, LNCS, Vol. 2898, Springer-Verlag, pp. 332-359, 2003.
[34] F. Zhang, S. Liu, and K. Kim: "ID-based one round authenticated tripartite key agreement protocol with pairings", available at: http://eprint.iacr.org, (2002).
[35] M. Hietalahti, "Efficient Key Agreement for Ad Hoc Networks", Master's thesis, Helsinki University of Technology, Department of Computer Science and Engineering, Espoo, Finland (2001).
[36] C. Gunter, "An identity-based key exchange protocol", in: Advances in Cryptology - Proceedings of EUROCRYPT, LNCS 434, 1989, pp. 29-37.
[37] P. Szczechowiak, L. Oliveira, M. Scott, M. Collier, and R. Dahab, "NanoECC: Testing the Limits of Elliptic Curve Cryptography in Sensor Networks", in 5th European Conference on Wireless Sensor Networks - EWSN 2008, Lecture Notes in Computer Science Vol. 4913, Springer-Verlag, pp. 305-320, 2008.
[38] D. Galindo, R. Roman, and J. Lopez, "A Killer Application for Pairings: Authenticated Key Establishment in Underwater Wireless Sensor Networks", in the 7th International Conference on Cryptology and Network Security - CANS 2008, Lecture Notes in Computer Science Vol. 5339 (Springer-Verlag, 2008), pp.120-132.