SCIS 2009 The 2009 Symposium on
Cryptography and Information Security
Otsu, Japan, Jan. 20-23, 2009
The Institute of Electronics,
Information and Communication Engineers

A Secure Clustering Scheme over an Energy-aware Routing Protocol for Monitoring Critical Conditions

Myunghan Yoo * Jangseong Kim * Kwangjo Kim *

Abstract— Wireless Sensor Network (WSN) is one of fundamental technologies for building ubiquitous computing environment. As the network consists of many sensor nodes with limited resources (*i.e.*, computation, storage and battery), the network has more security vulnerabilities (*i.e.*, Denial-of-Service attack, Sinkhole/Wormhole/Sybil attack, node compromise, message forgery and traffic analysis). Although routing protocols for monitoring critical conditions should provide resilience against known attacks, most of them do not provide security at all. For example, Boukerche *et al.* [4] proposed HPEQ (Hierarchical Periodic, Event-driven and Query-based) for surveillance of emergency events. By supporting load balancing, it provides longer network lifetime than other routing protocols and meets requirements of monitoring critical condition, simultaneously (*i.e.*, periodic, event-driven and query-based). However, it has several vulnerabilities to be deployed because of no guarantee of security requirements (*i.e.*, confidentiality, integrity of data packets and node authentication). In this paper, we propose a secure clustering scheme over an energy-aware routing protocol. The proposed scheme satisfies security requirement such as confidentiality, integrity of data packets and node authentication. Moreover, our scheme supports misbehavior detection of the selected aggregator without additional cost for selection of observing node.

Keywords: HPEQ, Secure Clustering, Energy-aware routing protocol, Monitoring Critical Conditions

1 Introduction

Wireless Sensor Networks (WSNs) which are one of the most promising technologies for upcoming ubiquitous society are expected to help people not only in the ordinary life, but also in the severe environment, which cannot be visited or needs to be observed for a long time. Therefore, there are a lot of literatures [3, 4, 13, 14, 17], which present various kinds of useful protocols.

HPEQ (Hierarchical Periodic, Event-driven and Query-based) [4] is also useful and efficient protocol to monitor specific wide and dangerous areas. Main objective of HPEQ is to observe critical and physical environments such as a fire on a building, leaking of toxic gases, explosions, and even military battle field. Thus, the reliable event capture and its transmission are important.

However, some critical security vulnerabilities are caused by the naive clustering scheme and the data report. During cluster selection, an external adversary's node can join the process and can be the aggregator responsible for which is aggregating sensed data and reporting events occurring in its cluster to the sink. And on the transmission process of the critical event, an external adversary's node can capture the message and modify the message which presents that the event does not occur.

Thus, this paper proposes a secure clustering scheme that provides authentication of all cluster member nodes as well as aggregator, integrity, confidentiality, and freshness of each message. The proposed scheme requires that each node has only two embedded keys and one *Credential*, but when clustering nodes, the inspector node responsible for which is observing the aggregator's misbehavior should request the sink to authenticate the aggregator and the cluster member nodes. It causes additional communication overhead. However we will show that it is reasonable trade-off between the level of security and the overhead of memory, computation, and communication.

The rest of this paper is organized as follows: In Section 2, we examine related work which can be used for monitoring critical condition and show the reason why we choose HPEQ. In Section 3, we will show an overview of our scheme and what we have done. And next, we describe our scheme in detail with figures and assumptions to operate our scheme properly and securely in Section 4. We analyze the security level of our proposal in Section 5. And then, we evaluate our scheme in additional overheads of memory and communication which occupies the largest part of energy dissipation in Section 6. Finally, we address conclusion and future work in Section 7.

* Information and Communications University, Munjidong,Yuseong-gu, Daejeon, 305-732 Korea, ({bishnu, withkals, kkj}@icu.ac.kr)

2 Related Work

We need to define the requirements of protocols for critical application before examining related work. According to [4], applications monitoring critical condition have to meet the following requirements, simultaneously: periodic, event-driven and query-based. Query-based requirement needs fast path establishment to subscribe the current situation, when event occurs. For example, when a fire breaks out in a building, rescue units need to know where people are, in the urgent situation. Low latency for event delivery and reliability are also important requirements. But meeting these requirements simultaneously is quite difficult, due to the conflicts of requirements. For example, in Directed Diffusion paradigm [8], to mitigate node failures caused by sending packets on one path, transmissions are performed through multi-path which is probabilistically chosen. However, using multi-path may cause more energy dissipation and packet collisions. In the PFR [6] protocol, a source node forwards packets to the sink through nodes, in virtually connected zone which is constructed to propagate the data to the sink. The energy dissipation and costs can be increased by estimating direction of a received packets, since the node have to equip magnetometer module. SW-PFR [15] extended version of PFR uses sleep-awake duration for energy savings. Variable Transmission Range Protocol (VRTP) [1] tries to solve the problems of fault tolerance and energy efficiency by diversifying the data transmission range. Network lifetime, then, is prolonged since the nodes away one hop from the sink can sleep. On the other hand, an additional hardware component is needed.

SPIN (SPMS) [11], focusing on node failures, uses meta-data exchange, before data transmissions. SPMS requests and transmits data through the shortest multi-hop path to reduce energy costs and end-to-end delay. The mechanism for dealing with fault-tolerance, in SPMS, keeps the shortest and the second shortest paths in the routing table. When sensing node failures in the shortest path, a sender will choose the second shortest path. However, in a huge disaster (*i.e.* explosion), a large number of sensor nodes can be destructed including nodes on the second path.

PEQ [3] builds the shortest path for low latency for event delivery by using the hop count metric for routing mechanism which requires a small amount of information. When an event occurs, PEQ utilizes three way: broadcasting a message to intermediate nodes of a source node to find paths, receiving response including intermediates' hop level and identification. As another next hop node, a one hop less node from the sink will be designated. This mechanism is also to avoid loop formation. Simulation results show quite good performance of PEQ in terms of delay and delivery ratio. HPEQ [4] is a hierarchical version of PEQ. HPEQ shows more uniform load balance, lower latency and higher delivery ratio than PEQ, by aggregating data from clusters which consist of a number of nodes. There are several hierarchical protocols: APTEEN [14], PEGASIS [13], and Energy-Aware Routing for Cluster-Based Sensor Networks [17] which are good solutions for energy efficiency and latency, but complicated. The clustering mechanism of HPEQ is inspired from LEACH [9]. In LEACH, we select an aggregator based on probabilistic threshold and normal nodes select their aggregator based on signal strength. However, in HPEQ, a selected node based on probabilistic threshold is just a candidate to be an aggregator, broadcasts a message to request remaining energy to its neighbors. Neighbors, then, reply with their identification and the remaining amount of energy. Finally, a node which has the highest level of energy will be assigned as an aggregator. In LEACH, the communication with the sink is performed by only aggregators with one hop. However, this way is possible, when the scale of a network is only small. On the other hands, HPEQ supports multi-hop communication between an aggregator and a sink.

However, due to the naive aggregator selection, clustering and transmission, HPEQ causes several critical security vulnerabilities. In the communication point of view, there is no guarantee of the confidentiality and the integrity of each message. Thus, anybody can eavesdrop and modify every message. For example, when an event, such as fire and appearance of enemies in the battle field, occurs, an adversary can change the message to report event to the sink to an ordinary report message. And, without any compromised node, an adversary's external node which has abundant computational and communicational resources can join a cluster even as an aggregator by advertising exaggeratedly its remaining amount of energy in the aggregator selection step. Then, the node selected with probability threshold will appoint the adversary's node as the cluster aggregator. The adversary's node, then, can selectively transmit messages by dropping messages.

As above mentioned, HPEQ has the aggregator selection algorithm and the routing protocol based on LEACH. However, even though there are several variants of LEACH [9], such as SecLEACH [16] utilizing the random key predistribution [7] on LEACH and GS-LEACH [2] associating clustering and geographical information, applying security primitives to guarantee the authentication of each node, the freshness of each message, *etc.*, the variants cannot be directly applied to HPEQ [4], since variants of LEACH [9] have assumption that a cluster perform single hop communication between an aggregator and cluster member nodes and selects an aggregator with only the probability threshold like LEACH [9].

3 Design of Architecture

Before discussing our scheme, we will examine an architecture of our scheme to present simply what we have done. The architecture of HEPQ [4] consists of three categories: initial configuration, clustering that contains both the aggregator selection and the cluster configuration, and reporting which includes both data transmission to the aggregator and data transmission to the sink.

On the other hand, the architecture of our scheme is made of four parts: initial configuration, secure clustering, key management, and secure reporting. In initial configuration, firstly, the proposed scheme has a wider range than HPEQ. HPEQ only considers setting up the hop count for each node. However, we have also predeployment as initial configuration. In the predeployment, each node have embedded keys and the unique *Credential* shared with the sink, we will examine in Section 4.

In the secure clustering, our scheme has two characteristics. The one is that a node selected with a probability threshold based on LEACH [9] will be designated as the inspector of behaviors of its cluster. And the other is that all members should prove its validity to the sink.

Key management which determines the security level uses three kinds of keys: a global key shared with all nodes in the network, a unique key for each node used to authenticate the node itself and shared with only the sink, and a cluster key shared with cluster members including the aggregator and the inspector.

In secure reporting, if a cluster is made securely, providing confidentiality and authentication of the sender is naturally possible, due to the cluster key which shared only between the sink and cluster member nodes. Guaranteeing freshness of messages and delivery success is only needed. By the way, the original HPEQ can guarantee enough delivery ratio, even when jamming attack occurs in a way mentioned in [3]. And a nonce and addition operation to it can guarantee freshness of messages.

Therefore, we will mainly focus on secure clustering and key management.

4 Our scheme

4.1 Assumption and Notation

In this section, we will exploit some assumptions we used for the proper operation of our scheme. All nodes initially have the same amount of energy resources. However, the sink has no constraint of energy resources and the computational power and is secure against adversary's impersonation attack and compromising attack. Each node has two embedded keys: a global key and unique key. The global key is shared nodes deployed in the field and the sink and is used to prohibit external adversary's node from joining the network. The unique key is used to authenticate the own node and to guarantee confidentiality of the encrypted message with the unique key. There is another intrinsic assumption that cryptographic primitives such as the hash function, the encryption algorithms, *etc.* are cryptographically strong. And the last assumption is that, in the aggregator selection, probabilistically chosen inspector node has lower probability of compromising itself than nodes which are candidates of the aggregator. Finally, Table 1 presents notations which we use in this paper.

Table 1: Notations

REQ_EN	Request of the remaining amount of energy to received node
REP_EN	Reply message to the sender node
SET_AGR	Message designating a node as the aggregator
AGR_NTF	Message encouraging nodes to join the aggregator
ID_X	ID of node X
CN, A, I, P, N, C, S	Candidate of aggregator, aggregator, inspector, parent node, normal node, all cluster member nodes, the sink
$Nonce$	Randomly generated bits
CK, K_S	Cluster key and the sink's unique key
$E_{K_G}(M)$	Encrypted message M with the global key
$E_{K_X}(M)$	Encrypting message M with the node X's unique key
$Credential_X$	Pseudonym of node X, $E_{K_S}(ID_X \Vert Nonce)$
TR	The number of Transmission and Receiving
$AUTH\_REQ_X$	Authentication token of node X, $Credential_X \Vert$ $E_{K_X}(ID_X \Vert TR \Vert Nonce)$
$MAC_{K_X}(M)$	M's Message Authentication Code keyed with global or node X's key
$\Rightarrow, \rightarrow$	Broadcast and unicast transmission

4.2 Aggregator Selection

Like aggregation selection scheme of HPEQ, a node chosen with the probability threshold, which is called as the inspector node, broadcasts a message to its neighbors called as candidates in this step. The encrypted message with global key enables only valid nodes to decrypt the message.

(S1) $[I \Rightarrow CN]$ $REQ_EN \Vert E_{K_G}(ID_I \Vert Nonce \Vert$Amount of Energy$)$

Unlike HPEQ that lets all candidates reply, candidates only which have more or almost same amount of energy answer with a following message:

(S2) $[CN \rightarrow I]$ $REP\_EN \Vert E_{K_G}(ID_{CN} \Vert Nonce + 1)$

Nonce is added by 1 from the original nonce for freshness of sent message.

Then, even if the inspector has the more amount of energy than all candidates, the inspector node sends

SET_AGR and a encrypted message with the global key including ID of the inspector node and $Nonce$ adding 2 from the original one for freshness of this message to a candidate which replies the largest amount of energy among candidates.

(S3) $[I \rightarrow A]$ $SET\_AGR||E_{K_G}(ID_I||Nonce + 2)$

However, although the inspector node selects a candidate as the aggregator, we assume that the inspector node does not believe the selected aggregator yet, since an adversary can compromise a normal node and exaggeratedly inform its remaining amount of energy resources. Thus, inspector will ask the sink to authenticate the selected aggregator, in the Cluster Configuration.

4.3 Cluster Configuration

After assigned as the aggregator, the aggregator floods a notification message and encrypted message including new $Nonce$ to guarantee freshness. Neighbors also flood the received message, recursively, until hop count becomes 0, according to HPEQ.

(S4) $[A \Rightarrow N]$ $AGR\_NTF||E_{K_G}(ID_A||newNonce)$

Children node, receiving 0 of hop count, answer with its authentication token. The authentication token ($AUTH\_REQ$) includes two essential factors: $Credential$ and TR. $Credential$ is to prevent exposure of the cluster topology from eavesdropper by encrypting node's ID and a $Nonce$ with the sink's unique key. TR is a remaining energy metric for observation by the sink. This metric consists of just 16 bit. Half bits are for transmission and the other is for receiving. If the number of communication is over 8 bit, it will be set into 0, but the sink can calculate properly.

(S5) $[N \rightarrow P]$ $AUTH\_REQ_N$

Parents, receiving reply from their children, attach their authentication token, transmitting a message recursively to higher parents which sent notification message to them before.

(S6) $[P \rightarrow A]$ $AUTH\_REQ_P||AUTH\_REQ_{N_1}||\ldots$

Finally, the aggregator gathers authentication tokens. Above procedures is presented on Figure 1. And the aggregator report the aggregated message, adding its $Credential$ and MAC of the message, to the inspector node.

(S7) $[A \rightarrow I]$ $AUTH\_REQ_1||\ldots||AUTH\_REQ_n||$
$Credential_A||MAC_{K_A}(M)$

The inspector node add REP_EN sent by the aggregator and MAC keyed with its unique key to the aggregated message, sending it to sink through multihop set in initial configuration.

(S8) $[I \rightarrow S]$ $AUTH\_REQ_C||REP\_EN||$
$Credential_I||MAC_{K_I}(M)$

And then, the sink authenticates the message. If the received message is valid, the sink generates the cluster key (CK), and new $Credential$s, which contain each node's ID and new $Nonce$ and are encrypted with the key of the sink, encrypts them with nodes' unique keys, and transmits them.

(S9) $[S \Rightarrow C]$ $Credential_X||E_{K_X}(CK||\text{new}Credential_X)$

The generated cluster key will be used for aggregation of sensed data from legal cluster members.

5 Security Analysis

The proposed scheme provides confidentiality, freshness, and integrity during clustering. The proposed scheme achieves the security level that can defend, against exaggeratedly advertised amount of energy resources by adversary's external nodes, by utilizing the global key distributed to all nodes in the network. However, using only the global key, it cannot mitigate effects of compromised nodes which attempt to be the aggregator. Thus, we apply statistical calculation from the sink with reports of the number of transmissions and receiving. From statistical calculation of remaining amount of energy of nodes, the sink will not let the aggregator to be by sending the cluster key only valid members of the cluster, designating the inspector as the aggregator by sending additionally routing information based on the reported members of nodes.

The proposed scheme can also mitigate jamming attacks, when reporting and aggregating events, due to the path repair mechanism of PEQ [3]. The communication in HPEQ [4] is $hop-by-hop$ communication. The sender on the path to the aggregator or the sink sends the message. However, the destination node will not notify the sender with its ACK because of the jamming attack. The sender, then, floods $SEARCH$ message to find another path, but if there is no node answering $SEARCH$ message, the sender will spend more energy on transmitting widely as described in [3].

However, if jamming attacks and other attacks (*e.g.* sybil attacks, sinkhole attack, and selective forwarding attacks which are performed from the inside of the network with compromised nodes), the proposed scheme cannot mitigate. Firstly, an adversary perform jamming attacks. The sender, then, will broadcasts $SEARCH$ message as following the path repair mechanism. At that time, a compromised neighbor node by an adversary will lure the sender into setting destination node. And then an adversary will drop or selectively send the reporting messages.

6 Overhead Evaluation

We address additional computation and communication overhead from original HPEQ caused by applying cryptographic primitives and additional messages for secure communicate. We assume to use 128-bit AES

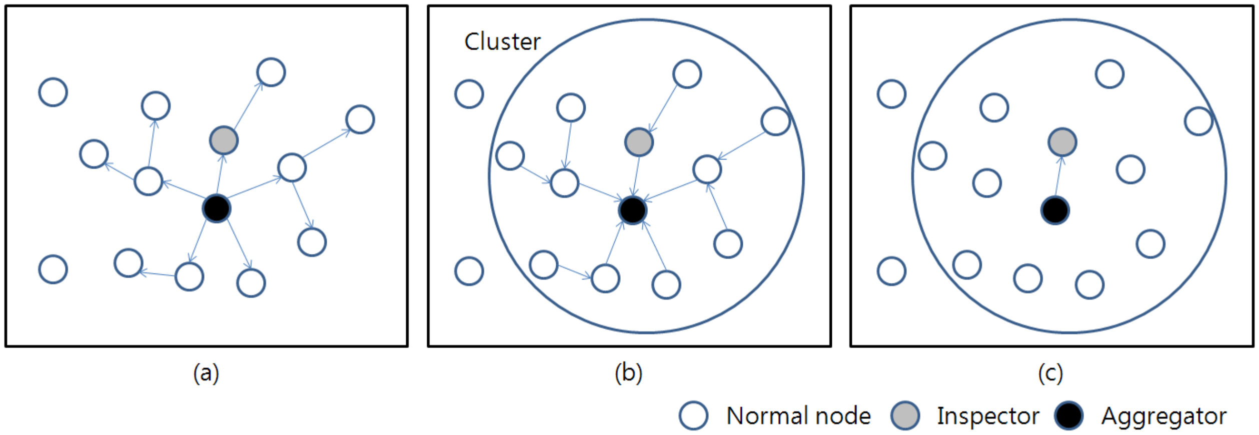

Figure 1: Cluster Configuration

cryptographic algorithm for encrypting/decrypting messages, SHA-1 as a message digest algorithm. Thus, each node has to store the required amounts of keys and *Credential* as shown on Table 2. Total 68 byte is very small, even if a sensor node has the extremely limited memory capacity.

Table 2: Size of Embedded Message Elements

Data	Size (byte)
Global key	16
Unique key	16
Credential	20
Cluster key	16
Total	68

We also consider of communication overhead which occupies the largest part of energy dissipation. According to the radio model in [9], the amount of energy consumption on communication is affected by a transmission range and the length of sent message. A transmission range depends on how sensors are deployed. Thus, this factor is out of scope in this paper. We try to clear size of each element in messages on each step in our scheme as summarized in Table 3.

Table 3: Size of Message Elements

Message Element	Size (byte)
Key	16
Hashed message	20
Nonce	6
ID	3
Amount of Energy	2
The number of Transmission and Receipt	2

Table 4 presents length of messages used for each step in our scheme. Notations, w, x, y, and z denote *REQ_EN*, *REP_EN*, *SET_AGR*, and *AGR_NTF*, respectively. According to [5] which measures computational and communication energy costs of cryptographic primitives, such as hash function: SHA-1 and symmetric key algorithms: RC5, DES, and 128-bit AES on MICA2 from CrossBow and on Ember EM2420 from Ember, from 8 byte to 24 byte length of messages, sending messages has very small gap of energy dissipation. Thus, we can ensure that our scheme shows almost the same amount of energy consumption of communication, until Step 4. At step 5, we have a fewer communication overhead, since [5] presents that 32 byte length of messages increases overheads by approximately 33.55% from 24 byte length. However, this overhead is reasonable to authenticate all member nodes of cluster. Steps 6 and 7 have to more overheads. Therefore, we need to divide message length into 24 byte, because, as above mentioned, 24 byte of message length consume almost same as 8 byte for transmission. Without sending all authentication tokens to the inspector node, our proposed scheme requires the same number of transmissions as original HPEQ to minimize additional transmission overheads.

Table 4: Size of Messages on each step

Step	Message Size (byte)	
	Each element	Total
S1	w + 3 + 6 + 4 + 3 (*padding*)	w + 16
S2	x + 3 + 6 + 7 (*padding*)	x + 16
S3	y + 3 + 6 + 7 (*padding*)	y + 16
S4	z + 3 + 6 + 7 (*padding*)	z + 16
S5	16 + 3 + 2 + 6 + 5 (*padding*)	32
S6	(n + 1) * (32)	(n+1) * 32
S7	C * 32 + 16 + 20	C * 32 + 36
S8	C * 32 + 36 + y + 16 + 20	C * 32 + y + 72
S9	16 + 16 + 16	48

7 Conclusion and Future Work

We examine requirements of monitoring critical conditions and apply security primitives to clustering scheme used for HPEQ. Our proposed scheme provide authentication of an aggregator and all cluster members. This way can be auxiliary of detecting adversary's compromised node, since the sink can continuously be reported from the inspector node and observe condition of whole of the network. Our proposed scheme achieves quite high energy efficiency minimizing additional transmissions from original HPEQ. However, we do not perform simulation or implementation on real sensor nodes.

References

[1] A. Boukerche, I. Chatzigiannakis, and S. Nikoletseas, "A New Energy Efficient and Fault-tolerant Protocol for Data Propagation in Smart Dust Networks using Varying Transmission Range", *In 37th ACM/IEEE Annual Simulation Symposium - ANSS*, 2004.

[2] P. Banerjee, D. Jacobson, and S. N. Lahiri, "Security and Performance Analysis of a Secure Clustering Protocol for Sensor Networks", *Proceedings of 6th IEEE International Symposium on Network Computing and Applications (NCA 2007)*, 2007.

[3] A. Boukerche, R. W. N. Pazzi, and R. B. Araujo, "A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications", *International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWIM'04)*, pp. 157-164, October 04-06, 2004, Venice, Italy.

[4] A. Boukerche, R. W. N. Pazzi, and R. B. Araujo, "HPEQ - A Hierarchical Periodic, Event-driven and Query-based Wireless Sensor Network Protocol", *Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)*, pp. 560-567, 2005.

[5] C. Chang, D. J. Nagel, and S. Muftic, "Measurement of Energy Costs of Security in Wireless Sensor Nodes", *Proceedings of ICCCN 2007*, Honolulu, Hawaii, USA, August 13 - 16, 2007.

[6] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis, "A Comparative Study of Protocols for Efficient Data Propagation in Smart Dust Networks", *In Proc. 2nd ACM - POMC2002*, 2002.

[7] L. Eschenauer and V. D. Gligor, "A key management scheme for distributed sensor networks", *In 9th ACM conference on Computer and communications security*, pp. 41-47, 2002.

[8] D. Estrin, R. Govindan, and J. Heidemann, "Embedding the Internet", *Communication ACM 43*, 2000

[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks", *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS'00)*, Hawaii,January 2000.

[10] J. Ibriq and I. Mahgoub, "A Hierarchical Key Establishment Scheme forWireless Sensor Networks", *Advanced Information Networking and Applications, (AINA '07). 21st International Conference on*, pp. 210-219, 2007.

[11] G. Khanna, S. Bagchi, and Y. Wu, "Fault Tolerant Energy Aware Data Dissemination Protocol in Sensor Networks" *IEEE DSN*, Florence, Italy, 2004.

[12] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures", *Elsevier's Ad-Hoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, 2003.

[13] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power Efficient GAthering in Sensor Information Systems", *in the Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, 2002.

[14] A. Manjeshwar and D. P. Agrawal, "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks", *in the Proceedings of the 2nd IPDPS02*, Ft. Lauderdale, FL, 2002.

[15] S. Nikoletseas, I.Chatzigiannakis, A. Antoniou, H. Euthimiou, A. Kinalis, and G. Mylonas, "Energy Effcient Protocols for Sensing Multiple Events in Smart Dust Networks.", *Proc. 37th Annual ACM/IEEE ANSS'04, IEEE Computer Society Press*, pp. 15-24, 2004.

[16] L. Oliveria, H. Wong, M.Bern, R. Dahab, and A.A.F. Lourerio, "SecLEACH: A Random Key Distribution Solution for Securing Clustered Sensor Networks", *In the Fifth IEEE International Symposium on Network Computing and Applications (NCA'06)*, 2006.

[17] M. Younis, M. Youssef, and K. Arisha, "Energy-Aware Routing in Cluster-Based Sensor Networks", *in the Proceedings of the 10th IEEE/ACM MASCOTS' 02*, Fort Worth, TX, 2002.