

A Secure permutation routing protocol in multi-hop wireless sensor networks

Hicham Lakhlef, Jean Frédéric Myoupo

Université de Picardie-Jules Verne, UFR Sciences, 33 rue Saint Leu, 80039 Amiens France

{ lakhlef.hicham@yahoo.fr, jean-frederic.myoupo@u-picardie.fr }

Abstract: A growing number of researches is done on the permutation routing problem in wireless networks, however, none of these studies do address the problems of security in the permutation routing in wireless sensor networks. The permutation routing problem in a military application is the fact that each soldier has items (information), that not concerned by him, and perhaps data which concerned, that is, in such applications and for confidential reasons, during the deployment, a soldier may hold items which are not necessary its own. The soldier to accomplish his task must receive its items from other soldiers in the network where it belongs. The necessity and the importance of secure permutation routing appear well when the permutation is a military application. The aspects of security that we deal with in this paper are not merely the authenticity, confidentiality, integrity, and non-repudiation, but we also show how we secure the partitioning into clusters and cliques in order to get consistency clusters and cliques.

Key Works: Wireless Sensor-Actuator Networks,
Permutation Routing, Security

1. INTRODUCTION

A sensor network is composed of a large number of sensor nodes which are densely deployed in a area. WSN can be deployed to provide continuous surveillance over an area of interest referred to as a sensor field [7, 21]. Wireless sensor nodes perform collaborative work [9] via wireless communication channels to retrieve information about targets that appear in the sensor field or to exchange some information. Higher-level decision making can then be carried out based on the information received from the sensor nodes. These networks can be deployed in inhospitable terrain or in hostile environments to provide continuous monitoring and information [9], or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants [1, 9, [21]note that in sensor network energy are limited and we must to minimize as possible the number of the broadcast in order to increase the lifetime of the system .

There are two types of wireless networks: Single hop wireless networks in which each station can transmit or communicate directly with any other station. All the stations use the same channel to communicate, and the message broadcast by one of the stations on the common channel is simultaneously heard by all other stations. In the multi-hop wireless networks intermediate nodes are used to route message from the source to the destination.

Permutation Problem: Consider a MANET (n, p) of p stations with n items saved on it. Each item has a unique destination which is one of the p stations. Each item has a unique destination which is one of the p stations. Each station has a local memory of size p/n in which n/p items are stored. It is important to note that in general, some of the n/p items stored in the station, say i , have not i as destination station. And even, it can happen that none of these n/p items belongs to it. In the other hand, the situation in which initially all items in i belong to i can also occur. The permutation routing problem is to route the items in such a way that for all $i, 1 \leq i \leq p$, station i contains all its own items.

A large variety of permutation routing protocols in a single-hop Network are known to day. These permutation routing protocols assume that the network are a single Hop Ad-Hoc Network, hence there is always a path connected by wireless links between a source and the destination. However, these varieties of methods are not adapted in the case of multi-hop Ad Hoc Networks. One way to solve this problem is to partition nodes into clusters where principal node in each cluster, called clusterhead, is responsible for routing items.

In reality, the change of information in wireless sensor networks is not secure, and the malicious node can do all tricks to prevent a normal run of permutation routing protocol. It can change (active attack) or intercept (passive attack) the information, and if a malicious node intercepts all information of a node say j , it can know the behavior of j , and thus the consequences in a military application for example will be very serious. Other behavior of the malicious node with the same consequences can occur with an active attack, when the malicious node modifies one or more information destined to node j . It can also make an attack to break the normal run of a clustering algorithm carried out by the sensors.

1.1. State of the art:

The first algorithm that treats the permutation routing in multi-hop wireless networks [3], needs

$$(k+1)n + O(|HUB_{\max}|) + k^2 + k$$

Broadcast rounds in the worse case. Where n is the number of the data items stored in the network, p is the number of sensors, $|HUB_{\max}|$ is the number of sensors in the clique of maximum size and k is the number of cliques after the first

clustering. The number of broadcast rounds was improved to $3n + 6 \log_2 k$ in protocol in [14].

The number of studies specifically targeted to permutation routing in single hop wireless networks has grown significantly. It is shown in [17] that the permutation routing of n items saved on wireless sensor network of p stations and k channels with $k < p$, can be carried out efficiently if $k < (p)^{1/2}$. Datta in [4] derived a fault tolerant permutation routing protocol of n items saved on mobile Ad-hoc network of p stations and k channels MANET(n, p, k) for short. He also assumed that in the presence of faulty stations some data items are lost. We came out with our work in [12] presenting a fault tolerant protocol which avoids the loss of items. The first energy-efficient permutation routing appeared in [18]. A more efficient energy-efficient permutation routing protocol was presented in [5]. In [23] Walls et al. propose an optimal permutation routing on mesh networks. Another approach as an application of an initialization algorithm appeared in [11]. All these approaches assume that the WSN is a single hop networks and none of these protocol is secure as in [13, 16].

1.2. Our contribution:

We consider a WSN (n, p) with n items, p stations. We first propose to partition the network into single-hop clusters also named *cliques* with a secure algorithm. Secondly, we run a secure local permutation routing to broadcast items to their local destinations in each clique. Next we partition the cluster-heads of cliques with the hierarchical clustering technique but this hierarchical algorithm is not secure, we define the possible attacks on this clustering protocol. Next we propose the solutions to overcome these attacks. We show how the outgoing items can be routed securely to their final destination cliques.

The rest of this paper is organized as follows: section 2 we define the preliminaries and some definitions, section 3 we present an protocol of permutation routing in multi hops wireless sensors network with an single channel, after an protocol without collision and conflict in the channels, using the maximal capacity of the network this is obtained by defining an optimal colouring algorithm, in section 4 we present the experimental results, and in the section 5 we conclude and we define the future works.

2. PELIMINARIES

We assume that each station has a local clock which keeps synchronous time by interfacing with a Global Positioning System (GPS). Time is divided into slots and all packet transmissions take place at slot boundaries in WSN (n, p).

As in [3, 4, 5, 13, 16, 17, 18], we suppose that the n items denoted a_1, a_2, \dots, a_n are saved on a WSN(n, p) such that for every $i, 1 \leq i \leq p$, station i stores the items. Each item has a unique destination station. It is important to note that hereafter a sensor knows the destination of items it holds. In fact the data item it holds is a couple $S(s, d)$, where s is the

real data item belonging to sensor source and d is the sensor destination. For every sensor $h, 1 \leq h \leq p$, let h_d be the set of items whose destination is sensor h .

The permutation routing problem is to route the items in such a way that for all $h, 1 \leq h \leq p$, sensor h contains all the items in h_d . Consequently, each h_d must contain exactly n/p items.

1.1. A clustering scheme in cliques

Our approach uses the secure clustering protocol from [22] to partition network into clusters (cliques). The figures 1. a and 1.b blow illustrate a network in which each clique is a single hop sub network. Each clique is a single hop network.

Figure 1. a: network with 11 sensors

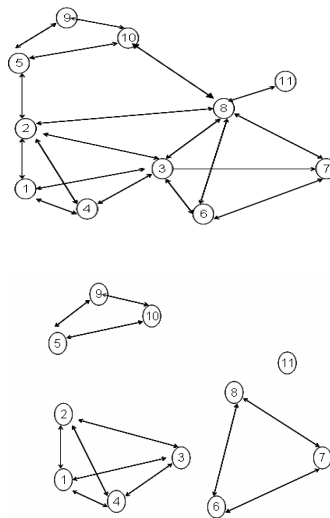


Figure 1. b: Resulting cluster formation in cliques

2.2 Hierarchical Control Clustering

Banerjee and Khuller [2] proposed a clustering algorithm for multi-hop sensor networks.

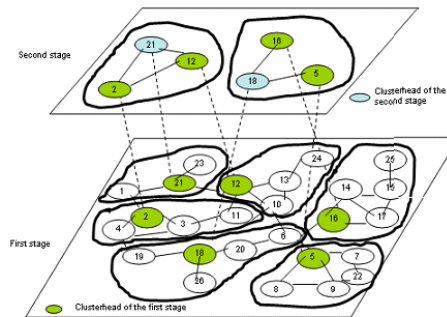


Figure 2: Hierarchical control clustering with $k=3$

Their clustering scheme is motivated by the need to generate an applicable hierarchy for multi-hop wireless environment.

Their method yields a multi-stage clustering. To reach their goal they construct a breath first search tree such that each level is composed of clusterheads of the immediate low level. These Clusters are by definition disjointed and the number of the nodes in a cluster remains between k and $2k$ for some integer k . Figure 2 shows a hierarchical clustering of a network of 25 sensors with $k=3$.

1.3. Problem Statement

Objective: The objective of our paper is to secure the information sent and to secure the clustering algorithms partitioning a sensor network into mutually disjoint cliques or clusters, so that all nodes in the same cliques or the same clusters can communicate with each other. We denote the view of cluster for node i as C_i . For brevity, we call C_i as the *cluster* of node i . We call a node a *normal node* if it follows our protocol. Otherwise, it is a *malicious node*. We would like to guarantee that all normal nodes have consistent clusters, as reflected by the following cluster agreement property. *Cluster agreement* for a normal node i is defined as:

Definition 1: (Cluster Agreement). For each sensor $j \in C_i$, $C_j = C_i$.

Definition 1 implies that for each normal node $j \notin C_i$, $i \notin C_j$ must hold. That is, each normal node belongs to only one cluster. Cluster agreement is broken if *cluster Inconsistency* is detected. For node i , cluster inconsistency is defined as:

Definition 2: (Cluster inconsistency). There exists a node $j \in C_i$ such that $C_i \neq C_j$.

It is desirable that each node find a cluster as large as possible. We do not consider trivial solutions with which each node forms a cluster that only includes itself.

Assumptions:

We assume that each node in $WSN(n,p)$ share unique pairwise private keys used for digital signatures with other nodes, and unique private key shared by all nodes in $WSN(n,p)$ this keys is used in phase 2 and phase 4, and All unicast messages exchanged between nodes are authenticated with the key shared between the two nodes. We use the low-end sensor nodes (e.g., MICA2 motes with 8-bit processors) defined in the recent investigations [8, 15] where it is shown that sensor can use public keys to perform cryptographic operations. Moreover, recent development of sensor platforms such as Intel motes uses more advanced hardware, and can perform public key cryptographic operations efficiently.

We use a combination of μ TESLA [20] and digital signature to authenticate broadcast messages. We use digital signatures when non-repudiation is necessary and μ TESLA for efficient broadcast authentication in other cases. We assume the clocks of the normal nodes are loosely synchronized, as required by

μ TESLA. We also assume the public keys used by the sensor nodes are properly authenticated. One approach to ensure this is to issue to each node a certificate for its public key so that other nodes can validate the node's public key by verifying this certificate. Moreover, the malicious nodes may launch Sybil attacks [6] or Wormhole attacks [10]. However, we assume these two kinds of attacks can be detected by using the techniques proposed in [19] and [10], respectively.

Now we propose to secure our algorithm phase by phase.

3. SECURE PERMUTATION ROUTING PROTOCOL

3.1 Phase 1:

We partition the sensors into cliques according to the secure protocol secure in [22], with this protocol we can't find Inconsistency Cliques. This algorithm gives k cliques, and the k clusterheads $CH_{clique-i}$ (i.e. for each clique a clusterhead), the purpose of this phase is to obtain sub-WSNs single hop sub-WSNs(n_i, p_i), $0 < i \leq k$, that perform data exchange between them and permutation routing as in phase2.

After, we consider the clusterhead graph named G_{KH} , a link between sensor A and sensor B (cluster-heads A and B) is possible if there is at least one direct link between sensor in clique of A and other in clique of B . The role of clusterhead $CH_{clique-i}$ is to collect all data items whose destination sensors are not in clique i . We Note $HUB(i)$ the clique i , and HUB_{max} the clique that contains the maximum number of sensors.

3.2 Phase2:

In this phase each item a_1, a_2, \dots, a_n in $WSN(n, p)$ is a secure hash value signed with the with key shared in $WSN(n, p)$. We run in each clique the permutation routing protocol as *the single-channel-routing* in [17], i.e., for the wireless network single hop.

Once more, the idea of this phase is similar to the protocol single-channel-routing [17], where each sensor broadcasts its items one by one and every time unit. Clearly if in a slot t_0 a sensor i broadcasts an item, the sensor, say j , whose identity matches the destination of the item being broadcasted copies it in its local memory. At t_0+1 j broadcasts an acknowledgment. If no sensor of the clique is the destination of the broadcasted data item then no action is taken and each sensor of the clique knows that the item is an outgoing item. Therefore each sensor counts the number of outgoing items it holds. Note that the clusterhead has the IDs of all the residents of its cluster. The broadcasts are carried out on cliques. So the clique with the great number of sensors should help to estimate the total broadcast rounds of this phase. At the end of this phase all data items that do not belong to the sensors of a clique are saved on the sensors of the clique. The goal now is to route these outgoing items to their final destinations.

This phase is carried in $(n/p)|HUB_{max}|$, because the phase processed in parallel and the clique that has the great number of sensor estimate the maximum number of

broadcast . Since $HUB_{max} \leq p$, $(n/p) \leq n$. Therefore the number of broadcast rounds of this phase cannot exceed n .

3.3 Phase 3:

In this phase we use the clustering algorithm in [2], which is not a secure algorithm. We recall the principle of this algorithm and we define the possible attacks and we propose solutions to counter them.. It takes as parameters a graph and an integer C , and it generates clusters with size greater than C and lower than $2C$. It generates a single cluster if the size of the graph in terms of number of nodes is $<2C$.

The distributed version of this algorithm is decomposed into two steps: *Tree-discovery* and *Cluster-Formation*:

Step 1: Tree-discovery needs five information on each node $\{src-id, parent-Id, root-Id, root-seq-no, root-distance\}$. Each node sends a tree discovery beacon which indicates its shortest hop-distance to the root. On receiving this beacon, the node discovers a shorter path to the root. it updates its hop-distance to the root according to the information in the beacon , and updates its parent as in figure 3: node E is originally at distance 3 from the root A, E receives a beacon from node D, at distance 1 from the root and consequently E chooses D to be its new parent . This decreases the distance of E from the root to 2

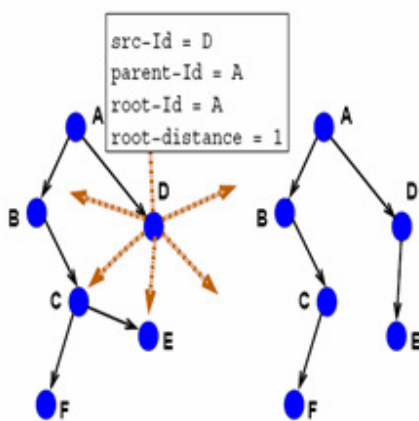


Figure 3. Tree-discovery of [2]

Step 2: In cluster formation starting with the leaves each node sends the number (counting himself) of sensor nodes that exists in the sub-tree rooted at him. It also sends this corresponding sub tree in the same message $\{ subtree-size, node-adjacency\}$. If this number is greater than C it initializes a cluster. If the size of the tree is $<2C$ the algorithm gives a single cluster.

Example: Set $C < 3$. In figure 4 $\langle 8 \rangle$ and $\langle 9 \rangle$ are sub-trees rooted à 8 and 9 respectively. $\langle 8 \rangle$ and $\langle 9 \rangle$ contains $0.5C$ nodes each. This figure shows an example of formation of clusters: Node 3 sends to its parent (Node 1) the message $\{subtree-size, node-adjacency\}$ (i.e., $\{C+1, node-adjacency\}$). Node 1 notes that the number of nodes in its

subtree-size equals $C+1$. So it creates the cluster $C_1 = \{\langle 8 \rangle, \langle 9 \rangle, 3, 1\}$. The number of the remaining nodes (3 nodes) is less than C . Hence the cluster $C_2 = \{2, 4, 7\}$ is created.

3.4 Attacks and solution:

a. Attacks: a malicious node can launch an attack with a lack of information sent to its neighbor in the step tree-discovery. The attack is in the construction of the tree. The malicious node which has one parent sends another beacon to other neighbor to take him as another parent. So the malicious node has two parents and it creates a cycle in the tree. The impact of this cycle is in the step *cluster-formation*: the malicious node sends to its both parents the message $\{ subtree-size, node-adjacency\}$, which initiate a cluster-formation at the same time, and the malicious node participates in the two initiations to provide inconsistency clusters. For example in figure 4 if we consider the cycle, created with the malicious node of identity 3, it sends the information $(C+1=0,5C+0,5C+1)$ to 1 and 4 that initiate *Cluster-Formation*. The resulting clusters are $C_1 = \{1, 3, \langle 8 \rangle, \langle 9 \rangle\}$ and $C_2 = \{4, 3, \langle 8 \rangle, \langle 9 \rangle\}$ that are inconsistency clusters and the cluster agreement is broken. The malicious node may also modify this information $(C+1)$ with an active attack in Cluster-Formation to eliminate the involvement of other nodes in Cluster-Formation.

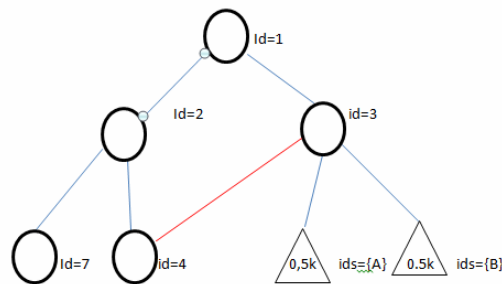


Figure 4. Tree-discovery and cluster formation

b. Solutions:

i. Secure Tree-discovery: Our solution for the secure tree discovery is *controller based*. The root controls this step. Each node N that wants to take neighbor N_v as a parent calculates a secure hash value over the message $M = \{src-id, N_v-id, parent-Id, root-Id, root-distance\}$, and signs this hash value with the key shared with the root (controller). Next it sends it to the root N_r . The root checks if there is no cycle in the tree. To reach this goal it calculates two secures hash values over $M_r = \{M, accord\}$ one with the key shared with N_v and the second key shared with the node N , and signs this hash value. It sends M_r as a response to N and N_v . However if there is a cycle or other problem the message of the root will be $M_r = \{M, not agree\}$. N_v verifies the information using the key shared with the root and takes it as a parent if there is an accord and if the root-distance is minimum.

ii. Secure Cluster-Formation: After the application of algorithm [22] in phase I each node knows the neighbors of its neighbors.

Starting from the leaves in the tree created in Secure Tree-discovery, the node N sends a secure hash value of the messages $M=\{m=\{ subtree-size, node-adjacency\}, m_p\}$ to its parent p . M is signed and hashed with the key shared with p , and m_p is signed and hashed with the key shared with the parent of p . And N requests a secure feedback from the parent of p because p may do an active attack on the information. p checks if the size of the sub-tree is $>C$ in order to initiate a cluster formation, but it cannot attack the information sent by its children, because its parent can detect this attack.

3.5 Phase 4:

Remark: Before we give the details of this phase, note that a node of figure 5 below is a clusterhead of a level in the hierarchical control clustering. Thus a node of figure 5 is in the cluster of at least C sensors and at most $2C+1$ sensors.

In this phase each item a_1, a_2, \dots, a_n in $WSN(n, p)$ is a secure hash value signed with the key shared in $WSN(n, p)$. We use the tree created and used in phase 3. In the tree a node broadcasts only to its parent or its children. The goal of this phase is to route the outgoing items saved to their final destination. To route these elements without collision and without conflict we use the optimal coloring algorithm defined in [14]. Figure 5 show an example of this coloring procedure.

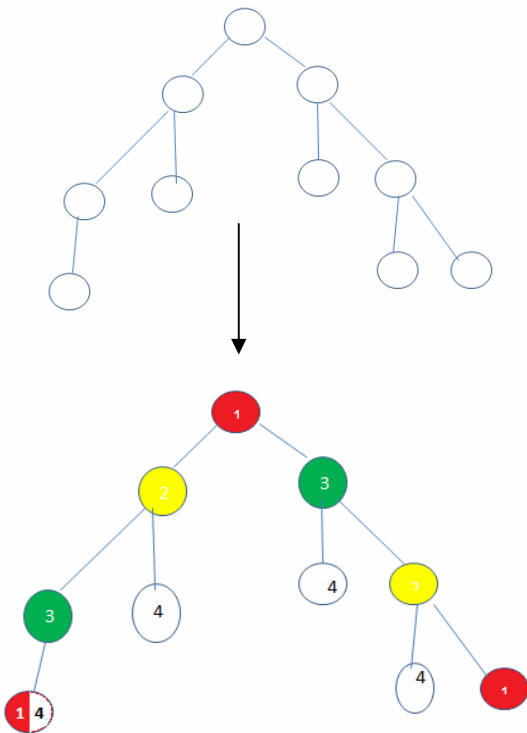


Figure 5. Optimal tree coloring

The goal of this sub-phase is to route the element to the clusterhead of its sensor destination. We use the following coloring algorithm.

Algorithm routing_of_outgoing_items

COL is the number of colors, in our example

Begin

If $((\exists color_j \text{ in } set_color_i) \text{ and } (color_j \bmod COL = 0))$
then

Broadcast the item;

for each color of j $color_j = color_j + 1$;

else

for each color of j $color_j = color_j + 1$;

Other nodes r copies the item in their own local memory if the destination match with one of the identity in $HUB(r)$;

end

a. External broadcasts

The algorithm works as follows: in each time slot the node (here clusterhead) checks if this slot is the appropriate time to broadcast. The appropriate time to broadcast is the time when the broadcast is taken without collision and without conflict, with instruction $color_i \bmod C = 0$. The first group that has the color C broadcast their outgoing items in the first slot to their neighbors. But in the second slot, the round is for the group that has the color $C-1$, and so on. Clearly in a slot a clusterhead (node of figure 5) invites its resident sensors to broadcast one by one their outgoing data items to him. In the sub-slot that follows it broadcasts the received item to its neighbors. However this broadcasting process carried out by a clusterhead takes one slot time.

b. Internal broadcasts

On receiving an item whose destination matches with the ID of a sensor in $clique-i$ the clusterhead of $clique-i$ forwards it to its destination sensor in $clique_i$. It selects a sensor that has to save this item otherwise (this is possible in virtue of the precedent remark)

4. CONCLUSION

One of the most challenges on permutation routing in wireless network multi-hop or single hop is the security. In this paper we proposed a secure permutation routing in multi-hop wireless sensors network which the objective is to secure the information exchanged by the sensors.

However some open problems remain. The derivation of a fault tolerant algorithm from the multi hop protocol of this paper which guarantees the delivery of data items to non faulty nodes is to be investigated. Also, the construction of an energy-efficient permutation routing protocol for multi-hop ad hoc network is a challenge.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks Elsevier Journal*, Vol. 38, No. 4, pp. 393-422, March 2002.
- [2] S. Banerjee, S. Khuller, *et al.*, "A Clustering Scheme for Hierarchical Control in Multi-Hop Wireless Networks," *Proceedings of the 20th IEEE International Conference on Computer Communications*, Vol. 3, 2001, pp. 1028-1037.
- [3] A. Bomgni, J. F. Myoupo, "A Deterministic Protocol for Permutation Routing in Dense Multi-Hop Sensor Networks". *Wireless Sensor Network* vol.2 pp. 293-299, 2010.
- [4] A. Datta, "Fault-Tolerant and Energy-efficient Permutation Routing Protocol for Wireless Networks," *Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, Nice, 2003, pp. 22-26.
- [5] A. Datta and A. Y. Zomaya, "An Energy-Efficient Permutation Routing Protocol for Single-Hop Radio Networks". *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 4, 2004, pp. 331-338.
- [6] J. R. Douceur. The sybil attack. In First International Workshop on Peer-to-Peer Systems (IPTPS'02), Mar 2002.
- [7] D. Estrin, R. Govindan, J. Heidemann and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, Seattle, 1999, pp. 263-270.
- [8] N. Gura, A. Patel, and A. Wander. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In *Proceedings of the 2004 Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2004.
- [9] T. Haenselmann (2006-04-05), *Sensor networks*, GFDL Wireless Sensor Network textbook, retrieved 2006-08-29.
- [10] Y. Hu, A. Perrig, and D. Johnson. Packet leases: A defense against wormhole attacks in wireless ad hoc networks. In *INFOCOM*, April 2003.
- [11] D. Karimou and J. F. Myoupo, "An Application of an Initialization Protocol to Permutation Routing in a Single-hop Mobile Ad-Hoc Networks," *Journal of Super-computing*, Vol. 31, No. 3, 2005, pp. 215-226.
- [12] D. Karimou and J. F. Myoupo, "A Fault Tolerant Permutation Routing Algorithm in Mobile Ad Hoc Networks," *International Conference on Networks-Part II*, 2005, pp. 107-115.
- [13] D. Karimou and J. F. Myoupo, "Randomized Permutation Routing in Multi-hop Ad Hoc Networks with Unknown destinations," *IFIP International Federation of Information Processing*, Vol. 212, 2006, pp. 47-59.
- [14] H. Lakhlef, J.F.Myoupo "An efficient permutation routing protocol in multi-hop wireless sensor network", manuscript, 2011.
- [15] D. J. Malan, M. Welsh, and M. D. Smith. A public-key infrastructure for key distribution in tinysec based on elliptic curve cryptography. In *SECON*, October 2004.
- [16] F. Myoupo, "Concurrent Broadcasts-Based Permutation Routing Algorithms in Radio Networks," *IEEE Symposium on Computers and Communications*, 2003, pp. 1272-1278.
- [17] K. Nakano, S. Olariu and J. L. Schwing, "Broadcast-Efficient Protocols for Mobile Radio Networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 10, No. 12, 1999, pp. 1276-1289.
- [18] K. Nakano, S. Olariu and A. Y. Zomaya, "Energy-Efficient Permutation Routing in Radio Networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 12, No. 6, 2001, pp. 544-557.
- [19] B. Parno, A. Perrig, and V. Gligor. Distributed detection of node replication attacks in sensor networks. In *IEEE Symposium on Security and Privacy*, May 2005.
- [20] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar. SPINS: Security protocols for sensor networks. In *Proceedings of Seventh Annual International Conference on Mobile Computing and Networks*, July 2001.
- [21] K. Römer; Friedemann Mattern (December 2004), "The Design Space of Wireless Sensor Networks", *IEEE Wireless Communications* **11** (6): 54–61.
- [22] K. Sun, P. Peng and P. Ning, "Secure Distributed Cluster Formation in Wireless Sensor Networks," *22nd Annual Computer Security Applications Conference*, Las Vegas, 2006, pp. 131-140.
- [23] I. S. Walls and J. Žerovnik, "Optimal Permutation Routing on Mesh Networks," *International Network Optimization Conference*, Belgium, 22-25 April 2008.