# SeGCom: The Secure Group Communication Mechanism in Cluster-Tree Wireless Sensor Networks

Olfa Gaddour[1], Anis Koubâa[2,3] , Mohamed Abid[1]

[1]CES Research Unit, National School of Engineers of Sfax-Tunisia

[2]IPP-HURRAY Research Group, CISTER/ISEP, Polytechnic Institute of Porto, Porto, Portugal

[3]Al-Imam Muhammad ibn Saud University, Computer Science Dept., Riyadh, Saudi Arabia

olfa.gaddour@enis.rnu.tn , aska@isep.ipp.pt , mohamed.abid@enis.rnu.tn

## Abstract

*Security is a challenging issue in Wireless Sensor Networks (WSNs) due to the dual impact of their inherent constraints and their operation in open and harsh environments. The problem of securing a WSN becomes even more complex when considering group communications. In this paper, we address this problem and propose a new security mechanism for group communications in cluster-tree WSNs. We define a group as a set of sensor nodes in the cluster-tree network sharing the same sensory information (e.g. temperature, pressure, etc.). Our objective is to limit the access to the group data exclusively to the members that have securely joined the group. The main contributions of the paper are (1) the proposal of an efficient and secure group management mechanism for cluster-tree networks, and (2) a secure key distribution between group members. Finally, our security analysis shows that the proposed scheme is efficient and secure.*

## 1. Introduction

Wireless Sensor Networks (WSNs) are composed of energy-constrained nodes embedding limited transmission, processing and sensing capabilities. Sensor networks have been being deployed for a wide variety of applications, including environment monitoring, health-care monitoring, transportation systems, home automation etc. [1]. As WSNs are basically deployed in hostile environments, security becomes extremely important, since sensor nodes are exposed to different types of malicious attacks. However, due to resource and computing constraints, security in WSNs imposes several challenges that are more complex than in the other traditional networks.

Security in WSNs has attracted several research studies that have addressed various security problems such as authentication [2, 3], key distribution [4, 5], data confidentiality and integrity [6], intrusion detection [7], secure broadcast [8], Cryptography. The security problem in WSNs becomes even more challenging when dealing with the *group* security, as this grouping impose additional overhead in terms of network management. Several works have also addressed the latter problem [9, 10, 11, 12, 13, 14, 15], however, each of them relies on a specific and different grouping concept. In this paper, we focus on securing group communications in *cluster-tree* WSNs, where a *group* is defined as a set of sensor nodes sharing a common private information. This means that sensor nodes in a given group must send and receive messages to/from group members in a way that outsiders are unable to unveil the shared group data, even when they are able to intercept the broadcasted messages [16]. Thus, the main challenges can be summarized as follows: (1) the initiation and distribution of a group key in a secure and efficient, (2) the management of the group in the cluster-tree network. To illustrate the concept, let us assume that we have a WSN, where some sensor nodes collect temperature data, and other sensor nodes collect humidity data. Thus, we may consider that we have two groups in this particular WSN, and the main motivation is to be able to find adequate solutions to restrict the access to the temperature information to the members of the temperature group, and that of humidity to the members of the other group. Members of humidity group, for instance, should not be able to freely access temperature information without a prior authorization. In other words, grouping is based on the type of data of interest, and this group definition represents one of the contributions of this paper as compared to other related works dealing with secure group

communication.

**Related Work : The related work section must be organized and structure in a clear way. You describe several works but there is no relation between them. First, you must define a way to organize related works according to main topics. Then, for each main topic you clearly describe each related work and at the end you mention what is your contribution as compared to the previous works. Look at the way I did in my paper RTSS 2006** Group communication security is a challenging problem that has been addressed in several research works in the context of WSNs [9, 10, 11, 12, 13, 14, 15]. However, these works have different vision to the concept of grouping. In [9, 10], the authors have presented secure group communication by considering the whole network as one group. In [11, 12, 13, 14, 15], the authors define a group as the immediate neighbor nodes around a given sensor node. The authors in (site important) have proposed to form groups of sensor nodes with similar properties; However, this grouping concept is limited to sensor nodes located in a small region. In our paper, we consider a group as a set of nodes in a cluster-tree network sharing a common sensory information (such as temperature, humidity, light, etc.), which make the difference in the proposed security mechanism as compared to previous works.

In the literature, several papers have discussed the secure group communication problem in WSNs. In [17], the authors classified sensor nodes into three types according to their communication capability with the base station. They proposed a scheme using a key tree to manage group members as they join or leave the group. However, the authors did not describe of the group rekeying process. In [18], the authors proposed an energy-efficient level-based hierarchical system for WSNs, which also includes a group key management scheme. The proposed group rekeying scheme requires many exponentially-complex operations, which turns it unpractical for sensor networks. In [19], the authors proposed a centralized group rekeying scheme based on logical key tree hierarchy for WSNs. In all these three previous works [17, 18, 19], the base station is considered as the central controller and the whole WSN is considered as a group.

In [19], the authors proposed a group rekeying scheme for filtering false data in sensor networks. The group is defined as the immediate neighboring nodes around a sensor in the scheme. However, the authors just studied the case when the group members are geographically close to each other and did not consider the case when members are separated by multiple hops.

In [20], the authors proposed to form multicast groups in wireless sensor networks. They consider a conference keying mechanism with symmetric authentication protocols and a key hierarchy on which group key could be distributed. However, they consider an architecture where all the nodes in the network are identical which make their solution not scalable.

The authors in [21] proposed to form groups and based their classification on the common properties of sensor nodes. They propose two centralized group rekeying schemes which includes three steps: the group formation, the group maintenance, and the group dissolution. However, the proposed first scheme is not suitable for large groups and the second scheme requires a big latency when there are many hops between the group members.

In reference [22], the authors address the problem of key establishment in hierarchical sensor networks. They propose a group-based key pre-distribution scheme based on a hierarchical wireless sensor networks using bivariate polynomials and proposed to establish inter group and intra group keys. However, they consider groups with members that are in the same communication range and they omitted the case when a node leaves the group. However, they don't consider the case of multiple hops between group members.

**Contributions of this paper :** In this paper, we propose a group security mechanism for securing communication in a cluster-tree WSNs. The main contributions of this paper are three-folded.

- First, we present the cluster-tree network architecture that we consider in our work (Section 2).
- Second, we present the secure group management mechanisms (Section 3).
- Third, we demonstrate the feasibility and the efficiency of the proposed group security mechanism through a performance analysis (Section 4).

## 2. System Model and Assumptions

### 2.1. Network model

We consider a multi-tiered architecture as shown in Figure 1. Like in any tree network, the cluster-tree topology contains a special node called Base Station (BS), which identifies the entire network. In addition, in a tree network, some special devices may have the ability to allow the association from other nodes. These nodes are called cluster-heads (CH), which defines a

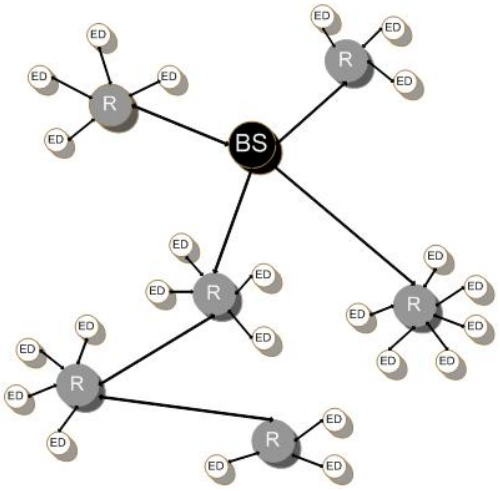


**Figure 1. The Network Model.**

cluster. Other end devices with no ability to associate other devices are called end-devices (ED). We assume that there is a single path between any pair of nodes, based on a certain tree routing protocol. This architecture is similar to the cluster tree topology defined in the ZigBee standard [23]. In what follows, we describe the functionalities of each node type of the aforementioned cluster-tree network model:

- *The base station*: also referred to as *coordinator*, is the root that identifies the whole network. It is responsible for performing critical functions such as assigning device addresses, controlling the network formation and operation, and collecting all the data. Only one BS exists in each network. The base station manages its cluster and all the other clusters in the network.

- *The cluster-head*: also referred to as *router*, has the ability to execute routing algorithms and forward messages to and from the other devices. It is able to establish and maintain multiple connections either as a parent or a child. Each Router receives the information flows coming from its child nodes of its local cluster, or from other cluster-heads, and then forwards the traffic to the base station BS or other cluster-heads.

- *The end-device*: is also called *child node*, and it has limited resources. It is optimized for very low power operation. It does not allow association and does not participate in routing. Each end-device ED is associated to the cluster-tree network through only one cluster-head CH.

### 2.2. Assumptions

We consider the following assumptions in our model:

- The network is static: all cluster-heads and end-devices are assumed to be static. In other words, we do not deal with mobility. This assumption is valid for many existing applications, such as home automation, environment monitoring, where the network infrastructure is fixed.

- The base station BS is assumed to be trusted and safe such that it will not be compromised by an attacker.

- We assume that an adversary can eavesdrop on all the traffic, inject packets and reply old messages previously delivered. We further assume that if an adversary captures a sensor node, all the keying information it holds can also be compromised.

- We also assume that it exists a secure channel between the end-device and the base station **Explain why?**(confidentiality, data authentication, integrity, and freshness).

### 2.3. Pre-deployment settings

Our work relies on the polynomial-based key pre-distribution scheme proposed by Blundo et *al.* in [24]. **you must give a brief description of this Blundo mechanism** Before deploying nodes in the network, we assume that an off-line key set-up server randomly generates a bivariate $t$-degree polynomial $f(x,y)$ over a finite field $Fq$ where: **(Equations must be numbered)**

$$f(x,y) = \sum_{i=0}^{i=\lambda} \sum_{j=0}^{j=\lambda} a_{i,j} x^i y^j \qquad (1)$$

The value of $q$ is a prime number that is large enough to accommodate a cryptographic key. The function $f(x,y)$ has the property $f(x,y)$=$f(y,x)$

For each sensor node $i$, the setup server computes a polynomial share of $f(x,y)$, that is, $f(i,y)$, and loads the single-variate polynomial to this sensor node. **(This sub-section must be extended and explained a little bit more.)**

## 3. SeG-Com: The Secure Group Communication Mechanism

In this section, we present SeG-Com, our proposed secure group communication mechanism for cluster-tree WSNs. The SeG-Com mechanism comprises five

main operations: (1) Group Formation, (2) Group Join, (3) Group Leave, (4) Secure Group Broadcast, and (5) Group Rekeying, which we describe in what follows.

### 3.1. Group Formation

In this section, you must respond to these questions in the order: 1- what is group: set of nodes sharing a common information? 2- How a group is created? step by step 3- How security information of a group are created and sent to GC?

### 3.2. Group Join

Explain step by step how a node can join a group securely

### 3.3. Secure Group Broadcast

Explain how a message is broadcast to the group members securely and step by step.

### 3.4. Group Leave

Explain how a node router or ED leaves the group securely and step by step.

### 3.5. Group Rekeying

Explain how a key is re-initiated step by step.

We define the group controller as a member of the group that has the ability to execute critical functions: key creation, key distribution, messages rekeying, and reports. In our protocol, we will consider that the first node that joins the group is the group controller (GC). The following figure represents the message sequence diagram that describes the communication between the first joining node and the coordinator.

The first node wanting to join a designated group sends a join request to the coordinator. After authenticating the request, the BS sends a confirmation to prove that the node joins successfully the group with a GC-flag to indicate that this node will be the GC. After receiving the flag, the designated GC sends a request to the coordinator to obtain the group identifier $G_{id}$ which will be the multicast address of the formed group. The coordinator then sends the $G_{id}$ to the GC. After receiving the $G_{id}$, the GC generates a random key $K_g$ as the group key.

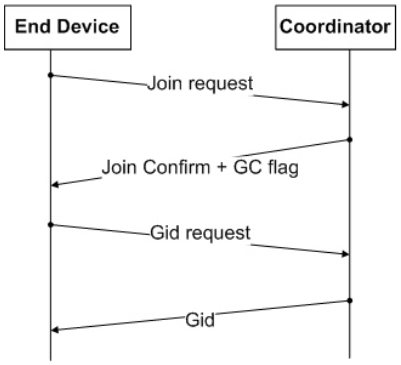


**Figure 2. Message Sequence Diagram between an ED and the BS.**

### 3.6. Broadcasting the Group identifier

The GC broadcasts a message (this message is sent to all the nodes in the network) to make known to all the nodes that it is the GC of this particular group. After receiving this message, all the routers update their routing table to add the location of the GC of this group in the network, and sensor nodes will know the identity of the GC. This will be useful for calculating the pair-wise key between each sensor node and the GC.

### 3.7. Broadcasting authenticated messages

Broadcasting the group identifier must be authenticated so that the adversaries cannot impersonate the group controller and broadcast its identifier to the network. To achieve authentication in the broadcast messages, we use $\mu$TESLA proposed in [25]. The GC and the nodes are loosely time synchronized. The GC computes the Message Authentication Code (MAC) on a packet with a key that is secret at that time and disclosed after a certain pariod of time. When the Receiving node gets this message, it can verify that corresponding MAC key has not been disclosed (MAC key chain $K_i = F(K_{i+1})$, the receiver buffers this packet and authenticates the packet when it later receives the disclosed key. To continuously authenticate broadcast packets, $\mu$TESLA divides the time period for broadcast into multiple intervals, assigning different keys to different time intervals. All packets in a particular time interval are authenticated with the same key assigned to that time interval.

**Particular case :** We consider the case when a GC has been formed in the network but the broadcast mes-

sage does not yet reached all the nodes One particular case may occur when a second node wants to form a group before receiving the broadcast message from the GC. It sends a request to the BS. The BS knows can not assign another GC-flag to another node. So this request is rejected and the sensor node must wait until getting the broadcast message.

### 3.8. The Joining Operation

After the GC designation controlled by the BS, the next group joining operations are controlled by the GC. The following figure represents a message sequence diagram that describes the communication between a second node joining and the GC in order to join the group.

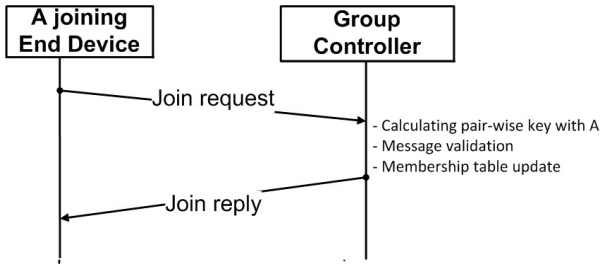


**Figure 3. Message Sequence Diagram between an ED and the GC.**

This joining operation is composed of 2 steps: After receiving the $G_{id}$, the node A wanting to join the group calculates the pair-wise key shared with the GC. This is calculated as follows:

$$f(x,y)_{x=A} = f(A,y) = f_A(y)$$

The node A contains the function. Until receiving the ID of the GC, the node A calculates this function for $y = GC$. The obtained $f(A, GC)$ is the pair-wise key with the GC. The node A then sends a join request to the GC, the join request JREQ has the following structure:

$$JREQ : ID_A|G_{id}|MAC(K_{A,GC}, id_A, G_{id}))$$

The joining node A sends its identity, $G_{id}$ to which it wants to belong and a MAC function as a signature that contains the calcumated pair-wise key, the A and the GC indentifier. The GC, having the function

$$f(x,y)_{x=GC} = f(GC,y) = f_{GC}(y)$$

stored in its memory, calculates this function when y is the A identifier. It obtains

$$f_{GC}(y)_{y=A} = f(GC,A)$$

It then calculates the MAC function received with the pair-wise key. If the two values of the MAC function calculated and received are equals, the message is verified and the GC accepts the request. It sends then a Join Response JREP as follows:

$$JREP : \{K_g, add_{gr}\}_{K_{A,GC}}$$

The GC sends then its reply containing the group key and the group multicast address encrypted by the pair-wise key $K_{A,GC}$. Each GC in the network maintains a membership table that contains all the members' identifiers of the group. After each Join operation, the GC updates its membership table and adds the member node identifier: this table has a very important role in the protection against possible attacks such as replay attack.

This operation is repeated until all the members wanting to join this group join successfully and securely the group.

### 3.9. The Leaving Operation

A leaving operation may be an administrative delete (when the node is for example compromised by an attacker) or may occur when a node request to leave the group. It may occur also when the node become faulty due to battery-energy consumption problem, malfunctioning, etc. When a node wants to leave a group, the same process of the joining operation is repeated as shown in the following figure:

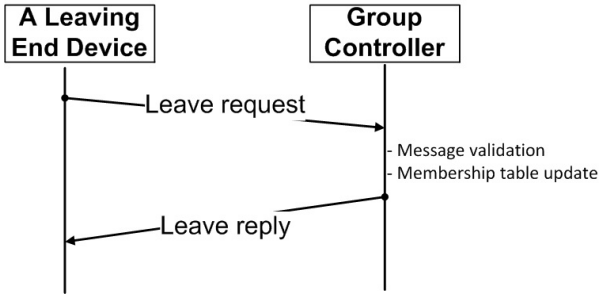


**Figure 4. The Leaving Operation.**

This operation is monitored also by the GC. The leaving node sends a LREP to the GC. The GC checks the validity of the message and sends a leaving reply to inform the member node that it is no longer a member of the group. The GC then updates its membership table.

### 3.10. Group rekeying

After a join operation, the GC sends a rekey message by multicast to all the group members. This mes-

sage contains the new group key encrypted by the old group key. The multicast here is possible because all the nodes can have access to the old group keys. After a leave operation: due to the fact that the leaving node also holds the old group key, we cannot encrypt $K_g$ with the old $K_g$ or any key known to the leaving node, as we did at the key update of Join Events. A generic way to update $K_g$ is to encrypt the updated $K_g$ with the pair-wise key of each member Mi with its group controller and unicast encrypted ($K_g$) KMi,GC to Mi.

# 4. Performance analysis

## 4.1. Storage cost

The pairwise key shared by the group controller with each joining member is built using Blundo's theory [24]. To set up the pair-wise key, the sensor node needs to evaluate the polynomial value at point (*i, GC*). Thus, the additional computation overhead for calculating the pairwise key is almost negligible. To use Blundo's theory, each sensor node $i$ needs to store a t-degree polynomial $f(i, x)$, which occupies *(t + 1)* log q storage space. As a result, if a key is of q-bits, a regular sensor node has to store *(t+2)* keys in its memory before deployment in a target field.

In addition, the GC also needs n storage units for the pair-wise keys shared with the group members and one unit for the group key. A regular sensor node in a deployment group needs to exchange its own id with the GC in that group in order to establish a pairwise key between them. Thus, the communication overhead is mainly due to transmission of the node identifier. We observe that in order to establish a pair-wise key, a regular sensor node needs to evaluate a t-degree polynomial share over a finite field Fq, which requires t modular additions and t modular multiplications over Fq. Liu et al. [26] showed that the evaluation of t-degree polynomial can be done efficiently using optimization technique. Hence, the computational overhead is due to efficient evaluation of a t-degree polynomial over Fq.

## 4.2. Communication cost

Let $|G| = n$ The proposed scheme requires one global broadcast. The GC may receive n join requests and needs to send the group key to $n$ members (the joining and leaving operations requires *O(n)* communication overhead). Thus, to set up the group key among $n$ members, it requires *2n* unicasts and one global broadcast. The pairwise key shared by the group controller with each joining member is built using Blundo's theory. To set up the pairwise key, the sensor node needs to evaluate the polynomial value at point *(i, j)*. Thus, the additional computation overhead for calculating the pairwise key is almost negligible. To use Blundo's theory, each sensor node $i$ needs to store a t-degree polynomial $f(i, x)$, which occupies *(t + 1)* log storage space. In addition, the group controller also needs n storage units for the pairwise keys shared with the group members and one unit for the group key.

## 4.3. Security analysis

- Characteristics of the adapted key pre-distribution scheme: we have adopted the Blundo's scheme in the pre-distribution phase. It was been demonstrated that this scheme is t-collusion resistant: any coalition of at most $t$ compromised nodes knows nothing about the shared keys computed by any pair of non-compromised nodes This scheme also is unconditionally secure: any pair of nodes can establish a shared key without communication overhead (if they know each other's ID).

- Security of the GC: we build a secure channel between the BS and all the sensor nodes. The GC is authenticated by the BS, and only the BS can give the authorization to be a group controller (the first node that requests to join a group is the only group controller). We used $\mu$TESLA to do the authenticated global broadcast between a GC and all the nodes. The scheme guarantees that packets are originated from the GC (authentication) and all the packets are fresh. Multi level $\mu$TESLA is resistant against replay attacks as well as Denial Of service attacks. No sensors can inject any fake messages into the WSN or modify any messages they forward while impersonating a group controller. The adversary cannot replay old rekeying packets because of time stamp information used. An adversary so cannot impersonate a group controller and start a new group.

- Protection against replay attacks: the GC maintains a membership table to save the list of members in the group. An intruder can just hear the join or leave requests and replay it byte-for-byte to the GC (a passive attack). This request is automatically rejected by the GC because the member already exists in the membership table. Thus, replay attacks is impossible in the Join and Leave operations

- Prevention against decoding previous and future messages within a group: the forward secrecy and

the backward secrecy are guaranteed by updating the group key in each join and leave operations. The GC updates the group key and send it to the current members of the group to prevent the insider nodes to get previous or future information of the groups.

- Protection against node capture attacks: the scheme generates a new key once there are modifications on the members of the group. Once the intruder is detected (by the use of some intrusion detection systems), it will automatically leave the group. Thus, the scheme is resilient to node capture attacks.

- Compromise resilience: a fresh key is always generated in each different rekeying session The fresh key is securely diffused among the multicast group members, always encrypted with the group keys Gk that are known only to the member nodes. After a sensor is compromised, it will only be able to decrypt the current multicast data; The security of a multicast message is broken only if at least one of the corresponding recipient sensors is compromised.

- Protection against revocation attack: the key revocation scheme depends on the GC to distribute and update the session key. To start the revocation attack, an adversary must impersonate the GC. The proposed key revocation scheme is immune to revocation attack if the GC is secure (by using the $\mu$TESLA scheme.)

- Network Survivability: the group key is updated efficiently. After the revocation, the adversary cannot lunch further attacks because he isn't still a member of the group. However, detection of compromised nodes is difficult. What an adversary can accomplish after it has compromised a sensor network: Possessing the $K_g$ doesn't enable the attacker to do broadcast messages because $\mu$TESLA is used. Because we deploy a periodic group rekeying scheme, the adversary can decrypt only the messages being encrypted using the current $K_g$.

## 5. Conclusion

In this paper, we have proposed a new concept to group sensor nodes based on the message cotent type. We have proposed a security mechanism, a group management and a group security management. Our scheme guarantees that any node in the network can establish a pairwise key with the Group Controller to join securely the group. Our analysis shows that the proposed scheme is efficient in computation and secure in term of secure communication.

## References


[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," *International Workshop on Wireless Sensor Networks and Applications*, 2002.

[2] Z. Benenson, N. Gedicke, and O. Raivio, "Realizing robust user authentication in sensor networks," *Real-World Wireless Sensor Networks (REALWSN)*, 2005.

[3] H.-R. Tseng, R.-H. Jan, and W. Yang, "An improved dynamic user authentication scheme for wireless sensor networks," *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, pp. 986–990, November 2007.

[4] L. Li, J. Li, L. Tie, and J. Pan, "Ackds: An authenticated combinatorial key distribution scheme for wireless sensor networks," *software Engineering, Artificial intelligence, Networking, and Parallel/distributed Computing, 2007. SNPD*, pp. 262–267, December 2007.

[5] Y. H. Kim, H. Lee, and D. H. Lee, "A key distribution scheme for wireless sensor networks," *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, vol. 17, pp. 572 – 577, March 2008.

[6] R. D. Pietro, P. Michiardi, and R. Molva, "Confidentiality and integrity for data aggregation in wsn using peer monitoring," *Security and Communication Networks*, vol. 2, pp. 181–194, janvier 2009.

[7] H. bin Wang, Z. Yuan, and C. dong Wang, "Intrusion detection for wireless sensor networks based on multi-agent and refined clustering," *International Conference on Communications and Mobile Computing*, vol. 3, pp. 450–454, December 2009.

[8] Z. Du, K. Wang, and L. Zhou, "Efficient broadcast authentication in wireless sensor networks," *IEEE Asia-Pacific Services Computing Conference, 2008*, pp. 187–192, 2008.

[9] Y. Cheng and D. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks," *Ad Hoc Networks (Elsevier)*, vol. 5, no. 1, pp. 35–48, 2007.

[10] D. Huang, M. Mehta, D. Medhi, and L. Harn, "Location-aware key management scheme for wireless sensor networks," *2nd ACM workshop on Security of ad hoc and sensor networks SASN 04*, pp. 29–42, 2004.

[11] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy efficient communication protocol for wireless microsensor networks," *3rd Annual Hawaii International Conference on System Sciences*, January 2000.

[12] S. Zhu, S. Setia, and S. Jajodia, "Leap: efficient security mechanisms for large-scale distributed sensor networks," *10th ACM conference on Computer and communications security. New York, NY, USA: ACM Press*, pp. 62–72, 2003.

[13] L. Zhang, Z. Hu, Y. Li, and X. Tang, "Grouping-based clustering routing protocol in wireless sensor networks," *wireless communications, networking and mobile computing, Wicom*, pp. 2452–2455, 2007.

[14] I. M. Author, "Some related article I wrote," *Some Fine Journal*, 1992.

[15] L. Li, J. Li, L. Tie, and J. Pan, "Ackds: : An authenticated combinatorial key distribution scheme for wireless sensor networks," *the software Engineering, Artificial intelligence, Networking, and Parallel/distributed Computing, 2007. SNPD*, pp. 262–267, 2007.

[16] X. Zou, B. Ramamurthy, and S. S. Magliveras, *Secure Group Communications Over Data Networks*. Springer, 2005.

[17] N. Thepvilojanapong, Y. Tobe, and K. Sezaki, "A proposal of secure group communication for wireless sensor networks," *The 23th Computer Security (CSEC) Group Meeting, IPSJ, Tokyo, Japan*, pp. 47–52, December 2003.

[18] M. Tubaishat, J. Yin, B. Panja, and S. Madria, "A secure hierarchical model for sensor network," *SIGMOD Rec.*, vol. 33, no. 1, pp. 7–13, 2004 1992.

[19] R. D. Pietro, L. V. Mancini, S. E. Y. W. Law, and P. J. M. Havinga, "Lkhw: A directed diffusion-based secure multicast schemefor wireless sensor networks," *ICPPW '03: Proceedings of the 32nd International Conference on Parallel Processing Workshops. IEEE Computer Society Press*, 1992.

[20] W. Zhang and G. Cao, "Group rekeying for filtering false data in sensor networks: a predistribution and local collaboration-based approach," *IEEE INFOCOM, Miami*, pp. 13–17, March 2005.

[21] Y. Wang and B. Ramamurthy, "Group rekeying schemes for secure group communication in wireless sensor networks," *Communications, 2007. ICC apos*, pp. 3419–3424, June 2007.

[22] A. DAS and I. Sengupta, "An effective group-based key establishment scheme for large-scale wireless sensor networks using bivariate polynomials," *Communication Systems Software and Middleware and workshops*, pp. 9–16, 2008.

[23] open-source toolset for the IEEE 802.15.4/ZigBee protocols website, "http://www.open-zb.net."

[24] C. Blundo, A. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," vol. 740, pp. 471–486, 1993.

[25] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The tesla broadcast authentication protocol," 2002.

[26] A.K. and S. I., "An effective group-based key establishment scheme for large-scale wireless sensor networks using bivariate polynomials das," *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE*, vol. 6, pp. 9–16, 2008.

[27] J. L. Hill, *System Architecture for Wireless Sensor Network*. University of California-Berkeley: PhD Dissertation, 2003.

[28] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," *IEEE INFOCOM Hongkong*, 2004.

[29] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-byhop authentication scheme for filtering of injected false data in sensor networks," *IEEE Symposium on Security and Privacy, Oakland, CA, USA*, pp. 259–271, May 2004.