An Energetic Cluster-Based Input Organization Protocol in Wireless Sensor Networks

[1]Rubeena Banu, [2]V. Kondal Rao
[1,2]Dept. of CSE, Swarna Bharathi Institute of Science and Technology, Khammam AP, India

Abstract
Recent advancement in wireless communication and microelectronics has enabled the design and development of wireless sensor networks with low cost, low energy consumption and high utilization. Many cluster-based wireless sensor network routing protocols have been proposed. However, most of them take little consideration on communication protection, which is important to ensure the network security. In this paper, a lightweight key management approach is presented. Its analysis shows that this approach is an effective solution to the key management of hierarchical clustered wireless sensor networks.

Keywords
Wireless Sensor Networks (WSN), Input Organization Protocol, Energetic Cluster

I. Introduction

Wireless Sensor Networks (WSN) are wireless networks composed of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. Each sensor node in a sensor network is typically equipped with a radio transceiver or other wireless communication device, a small microcontroller, and an energy source, usually a battery. Although the development of WSNs were originally motivated by military applications such as battlefield surveillance, however, due to the deployment flexibility and maintenance simplicity, wireless sensor networks are now used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control.

As the applications gain more ground, security issues have also become a hot research topic. In [1], a resource oriented security solution (ROSS) was introduced to protect the network connectivity of heterogeneous clustered sensor networks (HCSNs). The security analysis and performance simulation show that ROSS not only achieves the predefined security goals, but also allows a tradeoff between security and performance cost. In [2], three new mechanisms in the framework of random key predistribution were proposed to address the bootstrapping problem, namely the q-composite scheme, the multipath reinforcement scheme, and the random pairwise scheme. Each of these three schemes represents a different tradeoff in the design space of random key protocols. Paper [6] proposed the localized broadcast authentication in large sensor networks to take full advantage of roles of network nodes and local information in a sensor network, and to provide an alternative solutions regarding tradeoff between verification delay and broadcast overhead for satisfying applications with different requirements. A new protocol was introduced in [9] to construct the shared session key in wireless sensor network. This protocol has great scalability in simulation because the time needed to finish key negotiation does not depend on the number the sensor nodes. It can also save power by reducing the number of transmissions. The security attributes of wireless ad hoc networks were discussed in details in.

A WSN has some unique characteristics, such as limited computing power, mobility of nodes, large scale of deployment, unattended operation, limited communication bandwidth, and limited storage resources. Moreover, in some deployment scenarios sensor nodes need to operate under an adversarial condition. Security solutions for such applications depend on the existence of strong and efficient key distribution mechanisms. It is infeasible in uncontrolled environments, to visit a large number of sensor nodes, and change their configuration.Thus, sensor nodes have to adapt to their environments, and establish a secure network by using pre-distributed keys or keying materials, exchanging information with their immediate neighbors. Key distribution is an important issue in WSN design. It is a newly developing field due to the recent improvements in wireless communications. Two major types of cluster-based key management protocols are widely used in sensor networks: One is the low-energy key management protocol, introduced by Jolly et at. The other is the lightweight key management protocol, proposed by Eltoweissy, et al. However, these two protocols suffer the following security issues: cluster heads before the network is deployed. When the network is in operation, key information is exchanged mainly through the communication between cluster heads. The weakness of this approach is that once a cluster head is compromised, the entire cluster will be broken by simple DOS attack. (2) These two protocols all assume fixed cluster heads in a WSN, and require cluster heads to possess big storage capacity and high computing power. They assume that cluster heads can be decided before deployment, and the cluster heads will not change during the operation. This assumption is obviously not consistent with most clustered network routing protocols. It is also against the core design objective of a WSN – being energy efficient, because if the network cannot dynamically select cluster heads to balance energy consumption, the nodes that are far away from the cluster heads will exhaust their energy first and the network will contain "blind spots".

In this paper a dynamic key management protocol is proposed to satisfactorily resolve the above two issues. The protocol assumes that the wireless sensor system has already been equipped with effective security detection mechanisms, which can decide if a sensor node is compromised or has used up its energy. The rest of the paper is organized as follows: Section II, presents the new protocol with its key management mechanism and operation procedures. Section III, analyzes the performance of the new protocol. Section IV, summarizes the paper.

II. A New Key Management Protocol

In this section, we introduce our dynamic key management protocol. We first make some assumptions on the network entities, and then present the protocol and show how it works step by step. The key management protocol presented in this section uses a symmetric key system, and consists of the sub-protocols that define how keys are distributed, added, revoked, and updated during the life time of the sensor Kinit when the sensor nodes are first clustered. The key is erased from memory after the clustering is completed.

Cluster Head: A cluster head is a sensor node with better resources and may be used to collect and merge local traffic and send it to the base station. During the network operation, the cluster head is responsible for the integration of all cluster node data and transmitting the data to the base station. The communication between the cluster head and cluster members uses authentication key Kc for encryption. The first cluster head does not know the share main key Kshare among cluster members.

Clusters: A cluster is composed of a cluster head and node members. Each cluster has a unique cluster ID number and a cluster key Kc. Cluster nodes communicate with their cluster head directly, and there is no data exchange between sensor nodes. Because Exclusion Basis Systems (EBS) provide a framework for scalable and efficient group key management [11], the EBS group key management algorithm is used to ensure that when one or more nodes are captured, the need to send information for updating keys is optimized, and for each node the need for storing the cluster management group key is minimized.

Base Stations: A base station is typically a gateway to another network, a powerful data processing/storage center, or an access point for human interface. Base stations collect sensor readings, perform costly operations on behalf of sensor nodes and manage the network. In some applications, base stations are assumed to be trusted and temper-resistant. Thus, they are used as key distribution centers.

A. EBS Subsets

Group key management protocol in the traditional network has received extensive research. Typical protocols include the logic Level Binary Tree (LKH) protocol, introduced by Wallner et al. This protocol is simple and flexible in management, and each group member node only needs to keep log2(N) (N: number of members nodes in the group) keys to update key group, network. but the group needs to keep 2log2 N −1 keys.

B. Assumptions on Network Entities

Again, we assume that the functionality of intrusion detection is available (e.g., tamper detection, network intrusion detection) to the command node, although we are not aware of any intrusion detection for sensor networks and although we do not specify how one is going to work. Besides, the sensor nodes and the base stations are randomly distributed and are not aware of the topology prior to the deployment.

Sensor Nodes: Each sensor node is assigned a unique ID number by the base station and a main key Kshare before it is deployed. Sensor nodes communicate with the base station using the key encrypted data. Node ID number and key Kshare are saved in the base station.

A base station in the network is allocated with an initial key group has a large number of members, the required storage space is very big. The EBS group key algorithm is introduced to address this issue. With this new group key management protocol, the storage efficiency is nearly doubled compared with LKH.

As defined in , an EBS is a collection of subsets of the set of members. Each subset corresponds to a key and the elements of a subset are the nodes that have that key. An EBS of dimension (N, K, M) represents a situation in a secure group where there are N members numbered 1 through N, and where a key server holds a distinct key for each subset. If subset Ai is in the EBS, then the key Ki is known by each of the members whose number appears in the subset Ai. Furthermore, for each t ε [1, N] there are M elements in the EMS whose union is [1, N] – {t}. This means that the key server can evict any the replacement keys for the K keys they are entitled to know. This is done by multicasting M messages encrypted by the keys corresponding to the M elements whose union is [1, N] – {t}. Each new key is encrypted by its predecessor in order to limit decipherability only to the appropriate members.

To construct subsets of the EBS, a canonical enumeration method [3-4] is used: we consider all possible ways of forming subsets of K objects from a set of K + M objects. For the sequence of bit strings in Canonical(K, M) we form a matrix A, where K and M are understood, and whose C(K + M, K) columns are the successive bit strings of K+M length, each with K ones. "A" is called the canonical matrix for EBS(N, K, M). For example, the canonical matrix A for EBS(8, 3, 2) contains the enumeration of all C(5, 3) ways to form a subset of 3 keys from 5 keys, as shown in Table 1.

Table 1: Enumeration Matrix for EBS(8,3,2)

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
T1	0	0	0	0	1	1	1	1	1	1
T2	0	1	1	1	0	0	0	1	1	1
T3	1	0	1	1	0	1	1	0	0	1
T4	1	1	0	1	1	0	1	0	1	0
T5	1	1	1	0	1	1	0	1	0	0

After matrix A construction is completed, each row of the table corresponds to a subset Ti, where an entry 1 in the row means that the subset contains the corresponding node. Because N = 8, M9 and M10 are useless. Therefore, in Table 1 T1 = [5, 6, 7, 8], T2 = [2, 3, 4, 8], T3 = [1, 3, 4,
6, 7], T4 = [1, 2, 4, 5, 7], and T5 = [1, 2, 3, 5, 6, 8].

It is easy to prove:

[1,8] – [1] = T1 U T2, [1,8] – [2] = T1 U T3, [1,8] – [3] = T1 U T4,

Therefore, when any one node exits from the network, only two subsets of nodes need to send messages to update their keys. Moreover, the protocol requires only five management keys, while LKH needs 15 keys. This helps save storage space and reduces the need to generate key computation.

In this protocol, when the base station constructs EBS(N, K, M) model, the parameters N, K, and M can be raised in value such that more management keys are generated. This way when new nodes are admitted to the clusters, the spare keys can be used directly instead of generating new keys.

C. Key Management Procedure

A key management procedure is an essential constituent of network security. Symmetric key systems require the keys to be kept out of reach of the adversary [8]. That being said, because sensor networks are constrained by limited resources and computation capacity, it is important to balance the security level with these constraints. A good key management protocol must pay attention to the balance.

Below listed are notations used in our key management

protocol:
base station
Cj j-th clusters
Si i-th sensor node
Hj head of the j-th cluster ID(Si) ID of the i-th sensor node
ID(Ci) ID of the j-th cluster
Kshare(Si) the key shared between the i-th sensor node and the base station
Kc(Cj) authentication key of the j-th cluster
Mem(Cj) members of the j-th cluster Mem(Cj)i the i-th node in the j-th cluster Kebs EBS management key set
E(K,M) message M encrypted with key K
|| concatenation operation

1. System Initialization

After nodes are deployed to their physical environment, they first report to the base station their physical locations, and then the network starts to select cluster heads. According to the cluster head selection algorithm, each node decides if it is capable of serving as a cluster head. If so, the node broadcasts a Hello packet. The packet must be authenticated. Therefore for the first cluster formation process we use the initial key Kinit to encrypt the Hello packet.
H j → broadcast : E(K init , Hello)
When a node receives the message, it replies to the cluster that it intends to admit. The reply includes its ID and its response content Ack.
S i → H j : E (K init , ID(S i) || Ack)
Finally, the cluster head assigns IDs to all nodes that intend to join in the cluster and sends the information to the base station. The base station constructs EBS structure for the cluster, assigns the cluster ID, and generates the cluster key and associated management key. According to the EBS structure, the base station uses the shared main key Kshare to encrypt data exchanged with the cluster head and sensor nodes within the cluster. All encrypted data are sent to the cluster head. The cluster head keeps the
H New →
broadcast : E(K c (C1), Hello), E(K c (C 2), Hello),
segment of the data that is intended for it and passes other parts of the data to other nodes. Upon the completion, all
E(K c (C3), Hello) nodes delete key Kinit from their memory. The initial clustering is then finished and the entire logic structure of the WSN is saved in the base station.
H j → B : E (K share (H j), ID(S i) || ID(S m)...)

2. Cluster Head Update

After a period of time of operation, the system (4) After nodes make choices, they report to the base station via their cluster head the cluster they have chosen to join in. When the base station collects all the data, it constructs the EBS for the new cluster, generates the cluster key and management key, and then sends them to the cluster heads, which further pass them to their nodes. Below Cnew represents the new cluster for cluster head Hnew, and Kebs_new represents the new EBS management key set. re-selects the cluster head. Different from the initial cluster head selection, the key Kinit has already been Mem (C j) → H j
→ B : E (K share (Mem(C j)), erased for security reasons. When a potential new cluster head broadcasts a Hello packet, there is no shared key
ID(Mem(C j)) || Ack)
B → H j → Mem(C j) : E (K share (Mem(C j)),
between this node and other nodes. To solve this problem, one of the options is the use of μ TESLA (micro Timed Efficient Streaming Loss tolerant Authentication Protocol)
ID(C
new
) || K c
(C New
) || K
ebs _ new|| ...)
security broadcasting agreement: Before the new cluster head broadcasts a Hello message, it sends a request message to the base station for a key, and then uses this key to encrypt the data for broadcasting. After a while, the base station broadcasts the key to all nodes. Although this approach can guarantee data integrity, a malicious node may broadcast useless information to keep other nodes busy in receiving and saving the data until they run out of energy.
As shown in Figure 1, the network has three clusters with cluster heads H1, H2 and H3. Hnew is to be the new cluster head. According to the physical location of the new cluster and the coverage of the cluster head signal, the base station can calculate out that Hnew broadcast signal could reach the three adjacent clusters. So we can use the cluster keys of these three clusters to encrypt the Hello packet for authentication The concrete steps are as follows:
(1) The new cluster head sends a request to the base station via its current cluster head for the keys of adjacent clusters. (App represents the request.)
H New → H 2 → B : E(K share (H New), App))
(2) When the base station receives the request, it determines that the new cluster head is able to reach the three adjacent clusters based on the location of the new cluster head. Hence it sends the keys of the three clusters to the new cluster head.
B → H 2 → H New : E(Kshare (H new), Kc (C1)
|| Kc (C2) || Kc (C3))
(3) The cluster head encrypts the Hello packet with the received keys and broadcasts the packet to the three clusters.
(5) The base stations sends the IDs of all nodes in the new cluster, the cluster key and management key to the new cluster head Hnew.
B → H 2 → H new : E(K share (H new), ID(Mem(C new))1 || ID(Mem(C new)) 2 ... || K c (C new) || K ebs _ new)

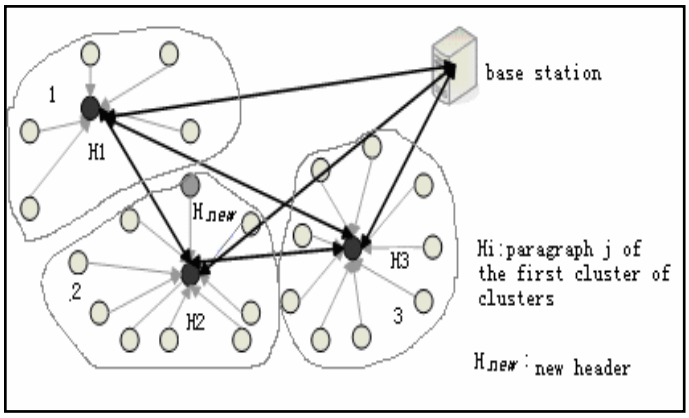

Fig. 1: Clustered WSN structure.

3. Add or Delete Nodes

When a new node is to join in the network, it needs to report to the base station its physical location. Then the base station finds its nearest cluster and sends the cluster key to the new node, and then notifies the cluster head to broadcast the Hello packet to the new node

When a node is failed, it must be excluded from the network. If it is an ordinary node, the only thing to be done is that the base station generates a new cluster key and dispatches it to all nodes using the EBS key distribution method. If the failed node is the cluster head, for example, H2 of cluster 2 in Figure 1, then the base station needs to generate a new key for cluster 2 and remove H2 from the cluster, and then use the new key to encrypt the keys of cluster 1 and cluster 3 and send them to all nodes in cluster 2. After that, the base station requests H1 and H3 to broadcast Hello message encrypted with their cluster keys.

III. Protocol Performance

In the presented protocol, any node in a running state only needs to keep a main key, a cluster key and a management key. The cluster head does not need to keep pair keys with cluster nodes. At the same time, the optimal EBS group key management algorithm eases the management cluster key storage burden and reduces the communication load when updating cluster keys. During the operation of the network, the majority of computation with the EBS algorithm is done by the base station. Therefore, the cluster head needs neither large storage space nor powerful computation capability. Any sensor node can serve as a cluster head.

In the protocol, the issue that broadcasting data is hard to encrypt during the cluster head update is resolved by using the adjacent clusters' keys based on the knowledge of physical locations. This ensures the security of data broadcasting. The attackers cannot use forged data to launch DoS attacks. They have no chance of being selected as the cluster head either to use worms to attack the network.

During the network operation, the actions that the new cluster head requests the keys of adjacent clusters, that nodes accept the invitation from the new cluster head, and that the base station distributes new cluster's management data are all done through the current cluster head.

The protocol also possesses good scalability. For a new node, only a main key is needed. The new node can select the optimal cluster to join in. A default assumption of the protocol is that each sensor node is able to know its physical location. This represents a constraint to the application of the protocol.

IV. Conclusion

As the applications of wireless sensor networks gain more ground, security issues have also become a hot research topic. This paper discussed the clustered WSN key management protocols and proposed a new protocol which is suitable for the key management of dynamic clustered networks, based on their operation mechanisms. The proposed protocol addresses the network security issues with cluster head update. It is distinguished with low power consumption, less computation workload and enhanced security. Besides, the protocol uses a symmetric key system, and consists of the sub-protocols that define how keys are distributed, added, revoked, and updated during the life time of the sensor network.

The protocol assumes that each sensor node is able to get its location information, which is currently a major restriction to its application.

Our next step is to design and develop an experiment software system to quantitatively study the proposed protocol's performance and compare it with that of other existing protocols.

References

[1] X. Cao, G. Chen, "ROSS: Resource Oriented Security Solution for Heterogeneous Clustered Sensor Networks," Int. J. of Intelligent Control and Systems, 12(4): pp. 317-324, 2007.

[2] H. W. Chan, A. Perrig, D. Song, "Random Key Predistribution Schemes for Sensor Networks", in: Proceedings of IEEE Symp. on Security and Privacy, pp. 97-215, Berkeley, CA, 2003.

[3] M.Eltoweissy, M. Heydari, L. Morales, H. Sudborough, "Combinatorial Optimization for Key Management in Secure Multicast Environments", Journal of Network and System Management, 12(1), pp. 33-50, 2004.

[4] M. Eltoweissy, M. Younis, K. Ghumman,"Lightweight Key Management for Wireless Sensor Networks", IEEE International Conference on Performance Computing and Communications, pp. 813-818, 2004.

[5] L. Eschenauer, Virgil.D Gligor, "A Key Management Scheme for Distributed Sensor Networks," in: Proceedings of the 9th ACM Conference on Computer and Communication Security, Washington DC, 2002.

[6] Q. Gu, J. Drissi, Localized Broadcast Authentication in Large Sensor Networks, Int. J. of Intelligent Control and Systems, 12(4): pp. 341- 350, 2007.

[7] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in: Proceedings of the 33rd Annual Hawaii Int'l Conf. on System Sciences. Maui: IEEE Computer Society, 3005−3014, 2000.

[8] G. Jolly, M. Kuscu, P. Kokate, M. Younus, "A Low-Energy Key Management Protocol for Wireless Sensor Networks," in: Proceedings of the 8th IEEE Symposium on Computer and Communications (ISCC), Antalya, pp. 335-340, 2003.

[9] B. Lai, S. Kim, I. Verbauwhede, "Scalable Session Key Construction Protocol for Wireless Sensor Networks," IEEE Workshop on Large Scale Real-time and Embedded Systems (LARTES), Austin, TX, 2002.

[10] A. Manjeshwar , D. Grawal, "TEEN: A protocol for enhanced efficiency in wireless sensor networks," In: Proceedings of the 15th Parallel and Distributed Processing Symp. San Francisco, CA: IEEE Computer Society, 2009−2015, 2001.

Rubeena Banu is a student pursuing M.Tech in Computer Science in Swarna Bharati Institute of Science and Technology, Khammam, AP affiliated to Jawaharlal Nehru Technological University Hyderabad. Her interested areas include Networking Protocols, Cluster Computing.

V. Kondal Rao received his M.Sc degree in Computer Science in the year 2001 from Kakatiya University, Warangal and his M.Tech in Computer Science in the year 2010 from JNT University, Hyderabad. At present working as Assoc. Professor in CSE department at Swarana Bharathi Institute of Science and Technology, Khammam, AP. His research interest includes Network Security and RTOS.