Cluster Based Mobile Key Management Scheme to Improve Scalability and Mobility in Wireless Sensor Networks

M.Shainika¹, Mrs.C.Hema²
¹Post graduate student, ²Assistant professor
B.S.AbdurRahman University

Abstract- The demands of Wireless Sensor Networks (WSN) increase the challenges in terms of scalability and mobility. The scalability is important to improve the energy efficiency and network lifetime, while mobility helps to improve the reachability of network. In this paper a new Cluster-based Mobile Key Management Scheme (CMKMS) is proposed. The CMKMS Scheme algorithm is used for the management and maintenance of keys under cluster-based mobile WSN network. In this scheme a cluster is formed and Cluster Head (CH) is selected and it is acting as a key manager. The work makes an assumption that sensor nodes and CH can move from one position to another. CH manages and maintains the private keys of sensor nodes. This algorithm also shows less computational overheads and energy consumption. This paper accomplishes the scalability of WSN using mobility-supported key management algorithms.

Key words: Wireless sensor network (WSN), Cluster-based Mobile Key Management scheme (CMKMS), Cluster Head (CH), scalability, mobility.

I. INTRODUCTION

The wireless sensor networks (WSNs) are distributed autonomous sensors to identify physical conditions like temperature, motion etc and to cooperatively pass their data through the network to a main location. The Wireless Sensor Networks (WSNs) are used in different real-time and mission-critical applications. The more modern networks are bidirectional, and also enabling control of sensor nodes activities. The development of wireless sensor networks was motivated by different military applications such as battlefield surveillance; today such networks are used in various industrial and consumer applications, such as process of industries, monitoring and control, machine health monitoring etc.

The WSN is built of nodes which from a few to several hundreds or thousands, where each node is connected to one or different sensors. That sensor network node has several parts: a radio transceiver with an internal antenna or connection to an external antennas, a microcontroller, an electronic circuits for interfacing with the sensors and an energy source, especially a battery or a form of energy harvesting. A sensor node might vary in their size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created.

The cost of sensor nodes is ranging from a few to hundreds of dollars, depending on the complexity of the sensor nodes. Size and cost constraints on wireless sensor nodes result in corresponding constraints on multiple resources such as memory, energy, computational speed and communications bandwidth. The topology of the WSNs can differ from a simple star networks to an advanced multi-hop wireless mesh networks. The propagation technique between the hops of the network can be routing or flooding. Wireless Sensor Networks are an active research area in computer science and telecommunication with numerous workshops and conferences arranged each year.

II. RELATED WORK

The key management and maintenance techniques [1] are defined using cluster based mobile key management scheme with static Cluster Head (CH). The available key management algorithm is efficient in terms of security, but it is not scalable enough to the changing conditions of network and do not work efficiently under node mobility. This method shows less computational overheads, energy consumption and delay as compared with state-of-art
solution. The mobility management is difficult when using static cluster head.

The method [2] a new energy efficient hierarchical clustering algorithm shows the performance of clustering algorithm based on soft threshold and member bounds for Wireless sensor networks using different clustering protocols. This algorithm has better performance in CH election and energy efficient. The hierarchical clustering algorithm also improves the lifetime and the energy consumption significantly in the wireless sensor networks. But this scheme is not scalable enough when the changing condition of network.

The energy efficiency [3] is essential requirement for wireless sensor networks. The energy efficient deterministic key management (EDDK) scheme is focuses on establishment and maintenance of pairwise keys and local cluster keys. The EDDK scheme is using the elliptic curve digital signature algorithm for both new and mobile sensor nodes can join or rejoin a sensor network securely. In this scheme is the energy consumption and overhead is very low. But this scheme is not scalable for large networks.

The existing key management algorithms such as network wide key, full pair-wise, probabilistic, matrix-based and polynomial-based schemes are efficient only in terms of security and they are not scalable enough when the changing conditions of network and not efficient under node mobility.

III.METHODOLOGY USED

A.Cluster-based Mobile Key Management Scheme

The Cluster-based Mobile Key Management Scheme (CMKMS) is used to improve the salability and mobility and efficiency of sensor nodes (SN). The network is partitioned into clusters. The node with maximum trust ability and efficiency is selected as Cluster Head (CH). The CH is acting as a key manager it aggregates information from all other nodes in the cluster. The work makes an assumption that sensor nodes and CH can move from one position to another, but CH and BS are fixed at one position.

![Cluster Formation](image)

**Fig: cluster formation**

This CMKMS algorithm considers the following two scenarios:

(i) **Scenario 1**, Here, the Cluster Head is static and aggregates information from all other sensor nodes in the cluster. The other sensor nodes are mobile nodes and they may move from one place to another i.e. may move from one cluster to other cluster. Here CH is acting as a Key Manager (KM) that manages the keys of all nodes within the cluster. Whenever a node is changing position and moving from one CH to other, i.e. from its home CH (HCH) to foreign CH (FCH), the key management scenario is considered.

(ii) **Scenario 2**, Here, CH and nodes both are mobile, i.e. Key Manager and nodes both are mobile. The CH should transfer the key management responsibility to other node in cluster, i.e. to make new KM or CH in network. Whenever the CH or KM is coming near the boundary of cluster, it transfers the key management responsibilities to other CH by running CH selection algorithm. This CMKMS approach establishes two private keys, home key and foreign key. The home key is used for the communication within own cluster and the foreign key is used when node moves from one cluster to another.

The main notations used to explain the algorithm are follows:

- N: node ID;
- C: cluster ID;
- K_h: home key;
- K_f: foreign key;
- N_c: average number of nodes per cluster;
- n_c: number of cluster in the network;
- l: average number of cluster neighbour;
- MAC: hash function.
IV. PROPOSED WORK

In the existing algorithm when a cluster head moves from one cluster to another cluster, the CH send the key management responsibilities to another node within the own cluster. The new CH is selected randomly using CH selection algorithm i.e. the new CH is selected based on its efficiency and trust ability. The proposed work here is the moving CH selects the new CH for the cluster. By this method the mobility management is improved. It is also increases the network lifetime and efficiency of the wireless sensor network.

A. Working Mechanism

The working mechanism is divided into two parts, which are given as follows:

(i) Set-up phase: In this phase a cluster is established in the network and cluster keys are set up in the network.

The set-up phase is divided into two parts, organizing network into clusters and setting up a cluster keys for each cluster in the network. It is responsible for establishing a secure link between clusters to make the whole network connect securely. Here, a unique ID is assigned to each sensor nodes that identifies them distinctly in a network. The algorithm considers that each node maintains the two key, key Kh, i.e. home key, and Kf, i.e. foreign key. Kh is used to communicate within the own cluster and CH, and Kf is used for the communication with foreign CH or nodes during node mobility. These keys will be used for secure information exchange in between the nodes.

The set-up phase is defined as the following algorithm steps:

Step 1: Start the program
Step 2: The sensor nodes waits for random amount of time.
Step 3: The nodes broadcast Hello message
Step 4: nodes receive Hello message
Step 5: If node decide the role, it rejects all the messages to become CH and member at same time and sends the ACK message.
Step 6: join the cluster of the node that send the message.
Step 7: Set Kh and Kf as cluster key and construct polynomial h(x).

(ii) Key maintenance: The key maintenance phase is responsible for maintaining and managing the home keys and foreign keys during node mobility.

This phase maintains the keys in following different situations:

Case 1: When new node joins the cluster
Step 1: The new node sends a beacon message to Cluster.
Step 2: The cluster nodes picking the beacon message and forward it to the Cluster Head.
Step 3: The cluster head merges the new node with the Cluster.
Step 4: New node runs the Cluster setup phase and gets the (Home key and Foreign key).

Case 2: When any node moves from one cluster to the other cluster
Step 1: The Leaving node sends a beacon message to the Cluster head of Home cluster.
Step 2: The CH informs this to its members and Neighbor CH.
Step 3: The Neighbor CH understands this and communicates with the incoming node with foreign key.
Step 4: Finally, new node runs the Cluster setup phase and gets the (Home key and foreign key).

V. SIMULATION RESULTS

The implementation is performed by using discrete event simulator Network Simulator-2 (NS-2). The NS-2 is an event driven packet level network simulator Version 2 included a scripting language called Object oriented Tcl (OTcl).

A. Results without WSN mobility

In case of key management algorithms, large numbers of overheads are occurring during the network initialization phase. It includes the encryption and authentication of the local broadcast messages, the verification and decryption of the received message from neighbor cluster and calculation of functions. The overheads are majorly measured in terms of computation overheads, which show number of packets transmitted for initialization, average energy consumption and delay incurred for it.

The CMKMS algorithm is compared with Energy Efficient Hierarchical clustering (EEHC) algorithm. By compared with EECH the CMKMS algorithm is less computational overhead, average energy consumption and delay.
b. Results with random WSN node mobility

The simulation considers that the nodes in network are mobile which moves from one position to another expect the CH. The results of computational overheads, average energy consumption and average delay show that performance overheads in case of mobility are more than the results without mobility. The main reason of increasing performance overhead is when node goes mobile, it changes its neighborhood. The change in neighborhood directly effects on calculation of pair-wise and individual keys.

C. Results with mobile CH

The performance overhead is increases when the node mobility. The main reasons are:

- It changes its neighborhood and the change in neighborhood effect on calculation of keys, pairwise- and individual keys; and
- It also reflects in re-election of CHs.

VI. CONCLUSION

The different key management algorithms are proposed to define a new key management approach to satisfy the increased scalability and mobility requirements of WSN applications. The proposed CMKMS algorithm satisfies the scalability and mobility requirement by reducing the computational complexity of the algorithm. The CMKMS shows less computational overheads, average energy consumption and improves network lifetime and efficiency. The CMKMS algorithm shows 20-23 percent improvements over state-of-art algorithm.

REFERENCES


