Binomial Tree based Key Establishment Schemes for Heterogeneous Wireless Sensor Networks

AnnapurnaH.S[#1], SiddappaM*[2]
[#]Research Scholar, Sri Siddhartha Academy of Higher Education,
Tumakuru, Karnataka, India
[1]hsassit@gmail.com
[*]Professor & Head, Dept. of Computer Science & Engg.,
Sri Siddhartha Institute of Technology, Tumakuru, Karnataka, India
[2]siddappa.p@gmail.com

***Abstract*-Clustering is one of the techniques used in Wireless Sensor Networks (WSNs) to enhance the network lifetime. Network lifetime can further be improved by reducing the number of computations performed by cluster members and cluster head. This paper presents two low cost key establishment schemes for clustered sensor networks which use binomial trees for managing the clusters: BLKH (Binomial tree based Logical Key Hierarchy) scheme and BWLKH (Binomial tree based Wireless Logical Key Hierarchy) scheme. The proposed schemes avoid rebalancing of key trees during join/leave operations by replacing logical key tree by binomial key tree. In both schemes, the number of new keys generated and the number of encryptions performed during node addition/eviction are less compared to LKH++ and WLKH schemes and are quite efficient. The proposed schemes increase the performance of a sensor node by minimizing the computations at cluster members. Thus, the schemes are scalable and suitable for large scale sensor networks.**

Keywords: Binomial tree, Cluster member, Cluster head, Key tree.

I. INTRODUCTION

Wireless Sensor Network is the primary area in research and is developing a promotable solution to different applications such as healthcare, military, automotive etc., [5]. It consists of small, inexpensive, battery-powered sensing devices which have low computation, communication ability, storage capacity, and bandwidth. Sensor nodes have the ability to transmit data through the channels and are limited in terms of energy and power[6]. The data present in the network can be easily eavesdropped, inject unwanted information or alter transmitted information. Sensor networks have the capability of collecting the data, process it and send the processed data to the central node for processing it further. Group communication is significant issue in WSN [11]. As the sensor network operates, the encryption of each message is required from primary node and communications which are interchanged between sensor nodes and central node need to be protected. So, there is need for providing security to the group communications in the sensor network. Usually, the security of the WSN is investigated widely in recent years. Most of the solutions are obtained to protect WSNs from known attacks or defensive strategies. Secure group communication has become a vital internet design issue because several applications such as distributed interactive discussion, real-time information services are all based on a group communication model. Secure group communication has one main point which is key management. It helps in protecting information in WSNs. Nodes in WSN are repeatedly exiting and getting added to the group. The key used for communication within the group is considered as the essential safety mechanism which provides secure group communication [7]. The newly added node should not be able to obtain the previous communications though it can get the future communications of the group [8].

The group key management scheme is one of the fundamentals in securing group communication [17]. So, in WSN there are many key management schemes which are primarily based on the symmetric and asymmetric encryption algorithms [15]. The simple method to transmit the group key is to preload it to all sensor nodes at the time of deployment. The easiest method for key establishment is to use a network-wide key. Unfortunately, even if a single node in the network is compromised, it would reveal the secret key and result in disclosure of all network traffic. In symmetric cryptographic algorithms, we have pool-based and probabilistic key pre-distribution schemes. The basic idea behind this is that a set of keys are picked from a key pool by the nodes before the deployment process so that any two sensor nodes have a probability of sharing one common key. Public-key cryptography (such as Diffie-Hellman key establishment) is another type of management scheme which gives more solutions with excellent security protection compared with symmetric key cryptographic algorithms. A trusted-server scheme is another type of key management technique that has been used in general networking environment which depends on a trusted-server for key distribution between nodes [9]. While

establishing a sensor network, the first step followed is the establishment of the cryptographic keys. The research has been conducted for a variety of protocols over many years. Key management techniques need to scale to large networks with thousands of nodes. The patterns of the communication networks vary between the existing techniques. The general wireless network consists of base station, cluster head and resource constrained nodes. The applications require the mobility of network nodes for the support. The network topology works more dynamically when the nodes move from one cluster to the next cluster. In this paper, we consider two schemes namely BLKH and BWLKH which are based on LKH++ and WLKH respectively. The proposed schemes make use of binomial trees instead of logical key trees which reduces the computational overhead at sensor nodes whenever there is a change in the membership. The remainder of the paper is organized in the following manner. The work carried out so far is explained in Section II. System model and assumptions made are discussed in Section III. In Section IV, we explain the LKH++ and WLKH Schemes. The proposed schemes are detailed in Section V and VI. We discuss the implementation results in Section VII and summarize the work in Section VIII.

II. RELATED WORK

The recent works carried out by various researchers towards the security of the group communication in WSNs are discussed in this section. [10] presents a new key management framework which is based on the combinatorial formulation of the group multicast key management problem which helps in managing the general challenge of operating keys for any trusted group communication. The Exclusion Basis System existence is explained and thereby the framework separates key management from encrypted message transmission resulting in the more efficient implementation of key management. The advantages of the Exclusion Basis System is considerable over current systems which use binary tree logical data structure to store keys. [11] investigates the replacement of the public key cryptography operations with the symmetric key services which are more efficient. Public key authentication is used to verify the authenticity of another party's public key to make sure that the person owns the public key it is claimed to belong, an efficient alternativethat uses the one-way hash function only. The scheme uses all sensors' public keys to construct a forest of Merkle trees of different heights. By randomly selecting the height of each tree, the computation and communication costs are minimized. The results in this framework show that the public key cryptography in sensor networks is limited and optimized. The significant savings of the power consumption is evaluated. The future work focuses on the variety of security protocols based on the public key cryptography.

[12] presents two optimizations for logical key tree organizations that utilize information about the characteristics of the group members to reduce the group rekeying further. The temporal patterns of group members have the partitioned key tree organizations which join and depart to decrease the overhead of rekeying. The results of this show that optimization can achieve up to 31.4% reduction in key server bandwidth overhead over the unoptimized scheme. The second approach is based on the loss probabilities of group members. The results of the latter approach show that the optimization reduces the rekeying overhead by 12.1%. [16] proposed a scheme based on the location-based virtual network infrastructure and is built upon a combinatorial formulation of the group key management problems. The efficient and secure key initialization is achieved in the proposed scheme by nodes without any communications. The system enables dynamic setup and management of arbitrary safe group structures with dynamic group membership. [14] discusses the strengths and weaknesses of LEAP. The vulnerability of the protocol to various attack models is analysed. Its effectiveness in defending against many sophisticated attacks such as HELLO Flood attack, Sybil attack, and Wormhole attack is shown. [13] describes LEAP+ (Localised Encryption and Authentication Protocol), a key management protocol for sensor networks which is designed to support in-network processing while at the same time restricting the security impact of a node compromise to the next system of the compromised node. The performance analysis shows that the LEAP+ is very efficient regarding computational, communication and storage costs. A low cost authentication scheme is proposed in [18] which offers high level of security. The scheme proposed provides confidentiality as well as authenticity in WSN using symmetric encryption and HMAC. The work which is based on LKH schemes is explained by Dimitris et al [19]. It reduces the rekeying cost by dividing the network into clusters and by localizing rekeying operations. A key generation scheme proposed in [20] uses a set of equations to set up secret keys which can be used for secure communication among nodes. The method uses a set of linear equations of two variables over polynomial equations for key generation which enhances network security.

III. SYSTEM MODEL AND ASSUMPTIONS

Our cluster-based WSN consists of three types of nodes, namely Base Station (BS), Cluster Head (CH) and Cluster Member (CM). The BS is assumed to be rich in resources, trust worthy and cannot be compromised. The CHs are powerful in terms of computation, communication and storage and are small in number. The CMs are resource limited sensing nodes and are large in number. All the nodes can be compromised by an adversary except BS. The cluster and CMs are managed by CH and all CHs are under the control of Base Station. Every

CM is assumed to be capable of reaching its CH. The CM is responsible for sensing the data in the deployment area and sending the sensed data to CH. The CH senses the data, performs aggregation on the data sensed and received from CMs and further sends aggregated data to the BS. The BSis connected to the outside world and communicates the data received from CHs. In this way, data flows from sensing nodes to BS in an hierarchical manner through CHs.

IV. LKH++ AND WLKH SCHEMES

In this section, we explain LKH++ and WLKH group key management schemes proposed in [1],[2].

A. LKH++ Scheme

A secure key management scheme based on Logical Key Hierarchy [3] has been proposed in [1]. This scheme uses a tree of keys to manage secure groups. The leaf nodes in the tree correspond to CMs. The intermediate nodes correspond to secondary keys which are known only to a subset of CMs and these keys are used to encrypt new cluster key. The root of the tree corresponds to a cluster key which is known to all the members in the cluster (both CM and CH). The key tree is maintained by a CH. Initially the CH is loaded with a unique identification number *i* and a key pair (pl_i, pr_i) consisting a public key and a private key. CM is loaded with unique identification number *i* and a private key pk_i which is used for confidential communication with CH. BS stores the IDs and the private keys of every sensing node.

1) *Cluster Formation:* After deployment, every CH broadcasts a hello message containing its ID and its public key pl_i. In response to this hello message, each CM selects a CH whose hello message has best signal noise ratio as its CH and replies with a message containing its ID, its private key and ID of the selected CH encrypted with the public key of CH as shown below:

$$CH \rightarrow * : (CH_i, pl_i)$$

$$CM \rightarrow CH : E_{pl_i}(CM_i, CH_i, pk_i)$$

2) *Key Tree Construction:* After receiving the replies from the CMs, CH constructs a key tree containing three types of keys.

The root key *RK* which is the cluster key used for confidential communication within a cluster.

1. Node keys NK_1, NK_2,…….. which are used to distribute *RK* to each CM.
2. Leaf keythat represents the private key of each CM.

Fig. 1 below shows a key tree for a cluster with 8 members.

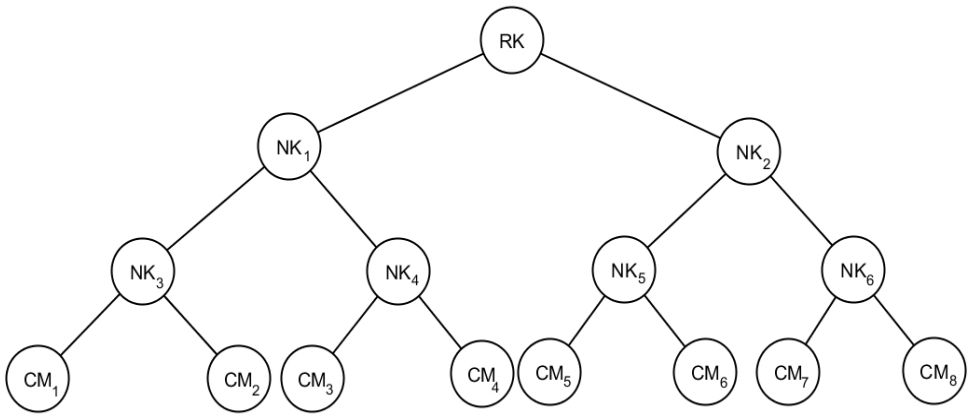

Fig. 1. Initial Key Tree

In this scheme, the keys along the path from the parent node to root node are always generated by the left most child in the tree using its private key and one-way hash function. Other child nodes receive the keys from its CH. For example, in Fig. 1 NK_3, NK_1, *RK* are generated by CM_1, NK_4 by CM_3, NK_5, $NK_2$by CM_5 and NK_6 by CM_7 as shown below.

$$NK_3 = H(PK_1), NK_1 = H(NK_3), RK = H(NK_1), NK_4 = H(PK_3), NK_5 = H(PK_5), NK_2 = H(NK_5), NK_6 = H(PK_7)$$

CH calculates all the node keys from NK_1 to NK_6 and the root key *RK* without communicating with any member and sends them to cluster members as given below.

$$CH \rightarrow CM_2: E_{PK_2}(NK_3),\ CH \rightarrow CM_3: E_{PK_3}(NK_3),\ CH \rightarrow CM_4: E_{PK_4}(NK_4, NK_1),\ CH \rightarrow CM_5: E_{PK_5}(RK),$$

$CH \rightarrow CM_6: E_{PK_6}(NK_5, RK),\ CH \rightarrow CM_7: E_{PK_7}(NK_2, RK), CH \rightarrow CM_8: E_{PK_8}(NK_6, NK_2, RK)$

3) Member Join: When a new member joins the cluster all the keys that will be revealed to new members are to be updated to ensure backward access control. Therefore CH sends a temporary key K_{tmp} encrypted with root key to all CMs. The CMscompute new keys by XORing K_{tmp} with every key they hold. CH sends the new keys to the joining member. For example, consider a key tree of Fig. 2 below.

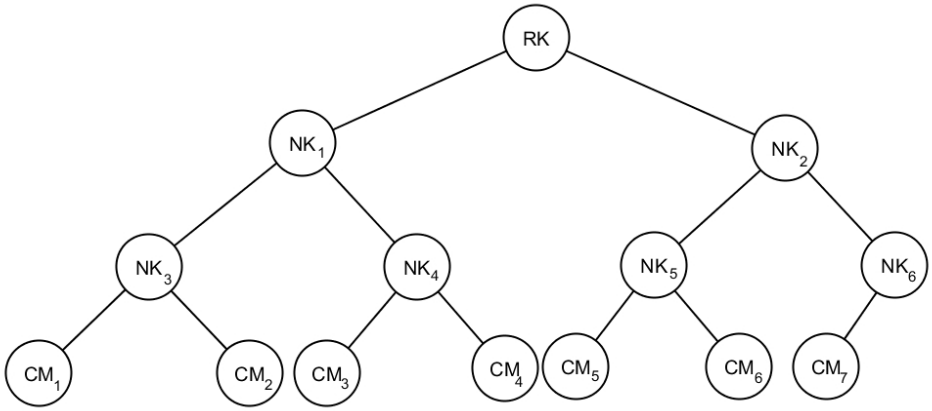

Fig. 2. Key tree for join operation

Now, CM_8 wants to join the cluster. The computation of new keys by CMsand CH and their distribution to new CMis as shown below.

$CH \rightarrow CM_1 to\ CM_7: E_{RK}(K_{tmp})$

CH computes new *RK*, NK_2 and NK_6 and sends the new keys to new member. CM_1 to CM_4 compute new *RK*, CM_5 and CM_6 compute new NK_2 and *RK*. CM_7 computes new *RK*, NK_2 and NK_6 as follows:

$RK' = RK \oplus K_{tmp}, NK_2' = NK_2 \oplus K_{tmp}, NK_6' = NK_6 \oplus K_{tmp}, CH \rightarrow CM_8: E_{pk_8}(NK_6', NK_2', RK')$

Key tree after CM_8 joins the cluster looks as shown in Fig. 3 below.

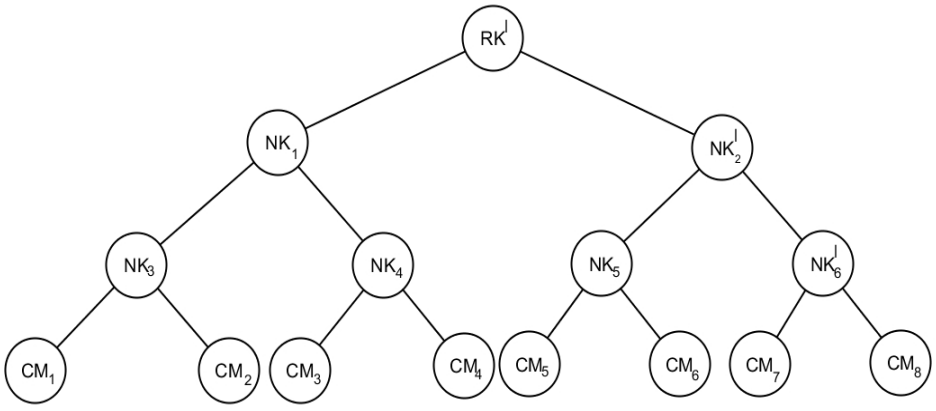

Fig.3. Key tree after CM_8 joins

4) Member Eviction: When an existing member is compromised it is expelled from the cluster and the key tree is reconstructed by CH. All the keys known to the evicted member must be changed. Referring to a key tree with 8 members (Fig.1), if CM_8 is evicted from the cluster, all the keys from its parent to root are to be changed. CH sends a temporary key K_{tmp} to remaining 7 members. The resulting tree is shown in Fig. 4 below.

$CH \rightarrow CM_1\ to\ CM_4 :\ E_{NK_1}(K_{tmp}), CH \rightarrow CM_5\ to\ CM_6 :\ E_{NK_5}(K_{tmp}), CH \rightarrow CM_7\ :\ E_{PK_7}(K_{tmp})$

The existing members compute new node keys and root key by performing XOR of $K_{tmp}$and old keys. CH computes new *RK*, NK_2 and NK_6. CM_1 to CM_4 compute new *RK*. CM_5 and CM_6 compute new *RK*, NK_2. CM_7 computes new *RK*, NK_2 and NK_6 as follows:

$RK' = RK \oplus K_{tmp}, NK'_2 = NK_2 \oplus K_{tmp}, NK'_6 = NK_6 \oplus K_{tmp}$

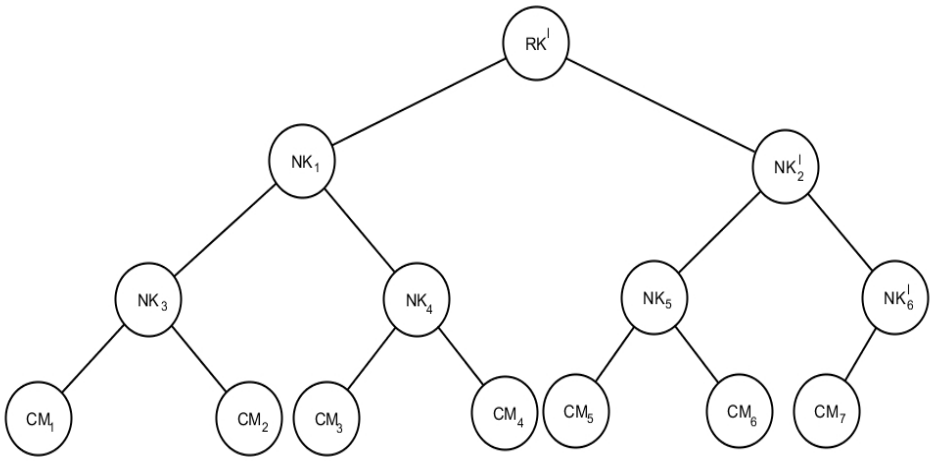

Fig. 4. Key tree after CM_8 leaves

B. WLKH Scheme

A group key management scheme based on LKH++ [1] has been discussed in [2]. LKH++ scheme is vulnerable to CH compromise attack since the private keys of CMs are known to CH. WLKH scheme [2] overcomes this problem by storing the hash value of CMs'private keys themselves. The node keys and root key are calculated using the hash value of private keys. The scheme also minimizes the computation cost at CM by having CH calculate the node keys and root key, thus making it more appropriate for WSNs.

1) Cluster Formation: CH broadcasts a message containing its ID, public key as part of cluster formation process. The CH joining the cluster acknowledges with a message containing its ID, hash value of its private key and ID of the selected CH, encrypted with the public key of CH as shown below.

$CH \rightarrow * : (CH_i, pl_i)$

$CM \rightarrow CH : E_{Pl_i}(CM_i, CH_i, H(pk_i))$

2) Key Tree Construction: After collecting the replies from CM, CH constructs the key tree as explained in LKH++ scheme and calculates the root key and node keys using the hash value of private keys of CMs. The keys computation and distribution by CH are shown below for a cluster with 8 members (See Fig. 1).

$NK_3 = H(H(PK_1) \oplus H(PK_2))$, $NK_1 = H(NK_3)$, $RK = H(NK_1)$, $NK_4 = H(H(PK_3) \oplus H(PK_4))$,

$NK_5 = H(H(PK_5) \oplus H(PK_6))$, $NK_2 = H(NK_5)$, $NK_6 = H(H(PK_7) \oplus H(PK_8))$, $CH \rightarrow CM_1 : E_{H(pk_1)}(NK_3)$

$CH \rightarrow CM_2 : E_{H(pk_2)}(NK_3)$, $CH \rightarrow CM_3 : E_{H(pk_3)}(NK_4, NK_1)$, $CH \rightarrow CM_4 : E_{H(pk_4)}(NK_4, NK_1)$,

$CH \rightarrow CM_5 : E_{H(pk_5)}(NK_5, RK)$, $CH \rightarrow CM_6 : E_{H(pk_6)}(NK_5, RK)$, $CH \rightarrow CM_7 : E_{H(pk_7)}(NK_6, NK_2, RK)$,

$CH \rightarrow CM_8 : E_{H(pk_8)}(NK_6, NK_2, RK)$

3) Member Join: Whenever a new member joins the cluster, the CH updates all the keys that will be disclosed to new member by performing XOR of old keys and K_{tmp} and sends these keys to the joining member by encrypting them with hash value of private key of joining member. Similarly the existing members also update these keys by XORing them with K_{tmp}.

4) Member Eviction: When a node is exiting the cluster, all the keys held by existing member must be updated to preserve forward secrecy. The CH sends K_{tmp} to existing members, encrypted using appropriate keys. The CMs then update the root key and node keys by XORing them with K_{tmp}. For example, in a cluster with 8 members as in Fig. 1, if CM_8 leaves the cluster, then the keys *RK*, NK_2 and NK_6 are to be updated. After receiving K_{tmp} from CH, CM_1 to CM_4 compute new *RK*, CM_5 and CM_6 compute new *RK* and $NK_2$and CM_7 computes new *RK*, NK_2 and NK_6 with XOR operation.

V. BLKH SCHEME

In this section, we explain our proposed scheme referred to as BLKH which is based on LKH++ scheme. Schemes based on LKH use hierarchical key tree to manage secure groups. BLKH scheme manages secure groups by maintaining Binomial Key Trees [BKTs] to store node information and key information [4]. Based on the number of members in the cluster, the BKT consists of multiple binomial subtrees that are rooted at different nodes. LKH++ scheme requires rebalancing of key tree if member join/eviction operations results in tree

imbalance. BLKH scheme does not require tree rebalancing as the tree always remains balanced in case of join/eviction operations. These operations may require reconstruction of key tree to restore the properties of binomial trees. The scheme results in reduced computation and communication cost, making it more suitable for WSN.

A. Key Tree Construction

The CH constructs a BKT which consists of multiple binomial subtrees (BSTs) rooted at different nodes. Fig.5 below shows initial key tree with 8 members together with the keys stored at each member.With 8 members, we have one binomial tree S_3 of height 3 and total number of BSTs is $2^{(3+1)}$-1=2^4-1=15. Out of 15, we have 8 BSTs of height 0, 4 BSTs of height 1, 2 BSTs of height 2 and one BST of height 3.

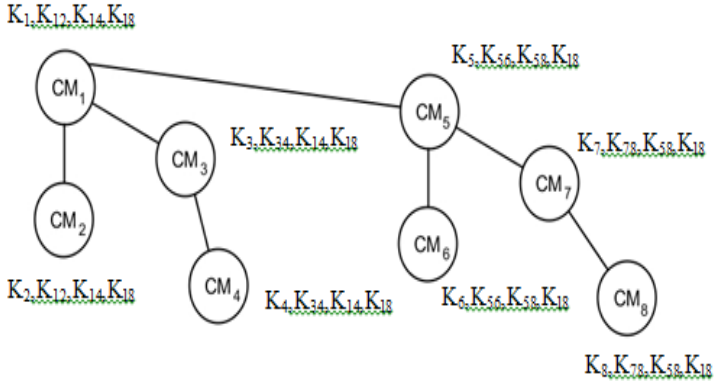

Fig. 5. Initial Binomial Key Tree

Referring to Fig.5, CM_1 computes K_{12}, K_{14}, K_{18}, CM_3 computes K_{34}, CM_5computesK_{56}, K_{58}, CM_7 computes K_{78}using their private keys and one-way hash function as shown below.

$K_{12} = H(K_1)$, $K_{14} = H(K_{12})$, $K_{18} = H(K_{14})$, $K_{34} = H(K_3)$, $K_{56} = H(K_5)$, $K_{58} = H(K_{56})$, $K_{78} = H(K_7)$

As CH stores the private keys of all the members in the cluster, it computes the sub cluster keysK_{12}, K_{34}, K_{56}, K_{78}, K_{14}, K_{58} and cluster key K_{18}without any communication and sends them to CMs by encrypting them with private keys of CMs.

$CH \rightarrow CM_2: E_{K_2}(K_{12})$, $CH \rightarrow CM_3: E_{K_3}(K_{14})$,$CH \rightarrow CM_4: E_{K_4}(K_{34}, K_{14})$, $CH \rightarrow CM_5: E_{K_5}(K_{18})$,

$CH \rightarrow CM_6: E_{K_6}(K_{56}, K_{18})$, $CH \rightarrow CM_7: E_{K_7}(K_{58}, K_{18})$, $CH \rightarrow CM_8: E_{K_8}(K_{78}, K_{58}, K_{18})$

B. Member Join

When a new member joins the cluster, all the keys from the joining point till the root are to be updated and Binomial Tree (BT) has to be reconstructed. Join operation causes less overhead than eviction since old cluster key can be used to communicate new keys to current CMs. CH uses the private key of the new member to communicate the new keys to it. Consider a BKT with 7 members as shown in Fig. 6 for a join operation. Suppose CM_8 wants to join the cluster.

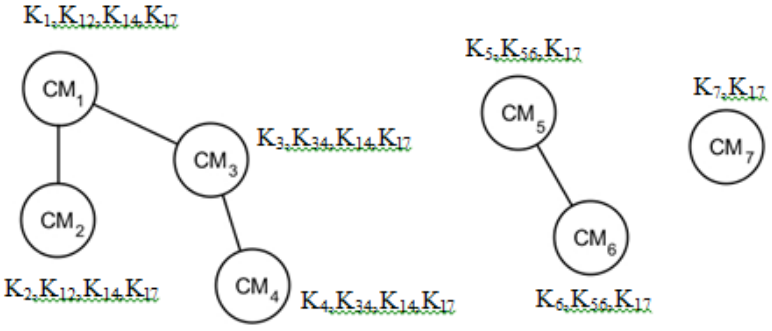

Fig. 6. BKT for join operation

The BKT after CM_8 joins the cluster looks as shown in Fig. 5. CM_1 to CM_4 compute new cluster key K_{18}, CM_5 and CM_6 compute K_{18}, K_{58}, CM_7 computes K_{18}, K_{78}. CH computes K_{58}, K_{78}, K_{18} and sends them to appropriate members as shown below:

$CH \rightarrow CM_1\ to\ CM_7: E_{K_{17}}(K_{tmp})$, $K_{18} = K_{17} \oplus K_{tmp}$, $K_{58} = H(K_{56})$, $K_{78} = H(K_7)$,

$CH \rightarrow CM_7: E_{K_7}(K_{58})$, $CH \rightarrow CM_8: E_{K_8}(K_{78}, K_{58}, K_{18})$

In this way all the keys that will be sent to CM_8 after it joins the cluster will be changed before it joins.

C. Member Eviction

Consider a BKT with 8 members as in Fig. 5. Suppose CM_8 leaves the cluster. The remaining 7 members from CM_1 to CM_7 form a secure group and require a new cluster key. BKT after CM_8 leaves the cluster is shown in Fig. 6. The new BKT has 3 binomial subtrees: S_2 with CM_1 to CM_4, S_1 with CM_5, CM_6 and S_o with CM_7. The computation of new key is shown below. CH and CM_1 to CM_7 compute the new cluster key K_{17}.

$$CH \rightarrow CM_1 \; to \; CM_4 : E_{K_{14}}(K_{tmp}), CH \rightarrow CM_5, CM_6 : E_{K_{56}}(K_{tmp}), CH \rightarrow CM_7 : E_{K_7}(K_{tmp}),$$

$$K_{17} = K_{18} \oplus K_{tmp}$$

VI. BWLKH SCHEME

Binomial Tree based WLKH scheme (BWLKH) improves WLKH scheme proposed in [2] by avoiding rebalancing of key tree during join / eviction operations. WLKH scheme requires tree rebalancing since join / eviction result in tree imbalance. BWLKH uses binomial trees to manage secure groups [4]. The binomial tree consists of several subtrees based on the number of members in the cluster.

A. Key Tree Construction

As a response to the hello message sent by CH, each sensor node sends a message containing its ID, hash value of its private key and ID of the selected CH. After collecting the responses from the CMs, CH constructs BKT in which nodes represent the CMs. CH then computes the sub cluster keys and cluster key and communicates them to all the members in the cluster. As an example consider a BKT representing a cluster of 8 members shown in Fig. 5. The keys held by CM are also shown at the nodes of the BKT, CH computes the cluster and sub cluster keys using the hash values of private keys of CMs as shown below:

$$K_{12} = H\big(H(K_1) \oplus H(K_2)\big), K_{34} = H\big(H(K_3) \oplus H(K_4)\big), K_{56} = H\big(H(K_5) \oplus H(K_6)\big),$$

$$K_{78} = H\big(H(K_7) \oplus H(K_8)\big), K_{14} = H(K_{12}), K_{58} = H(K_{56}), K_{18} = H(K_{14})$$

The keys computed by CH are sent to CMs by encrypting them with appropriate keys as follows:

$$CH \rightarrow CM_1: E_{H(k_1)}(K_{12}), CH \rightarrow CM_2: E_{H(k_2)}(K_{12}), CH \rightarrow CM_3: E_{H(k_3)}(K_{34}, K_{14}), CH \rightarrow CM_4: E_{H(k_4)}(K_{34}, K_{14})$$

$$CH \rightarrow CM_5: E_{H(k_5)}(K_{56}, K_{18}), CH \rightarrow CM_6: E_{H(k_6)}(K_{56}, K_{18}), CH \rightarrow CM_7: E_{H(k_7)}(K_{78}, K_{58}, K_{18}),$$

$$CH \rightarrow CM_8: E_{H(k_8)}(K_{78}, K_{58}, K_{18})$$

B. Member Join

When a new member joins the cluster, CH sends $K_{tmp}$to the current members using which the CMs update their keys by XORing them with K_{tmp}. CH also updates the sub cluster keys and cluster key and sends them to joining CM. Consider BKT with 7 members as in Fig.6.

Suppose CM_8 joins the cluster, the BKT changes as shown in Fig. 5. After the construction of BKT, CH sends $K_{tmp}$to existing CMs encrypting it with old cluster key. Each CM updates the keys it holds by XORing them with K_{tmp}. The computation of new keys by CM and CH and their distribution in given below. CM_1 to CM_4 compute new cluster key K_{18}, CM_5 and CM_6 compute K_{18}, K_{58}, CM_7 computes K_{18}. CH computes K_{58}, K_{78}, K_{18} and sends them to appropriate members as shown below.

$$CH \rightarrow CM_1 \; to \; CM_7: E_{(K_{17})}(K_{tmp}), K_{18} = K_{17} \oplus K_{tmp}, \; K_{58} = H(K_{56}), K_{78} = H\big(H(K_7) \oplus H(K_8)\big),$$

$$CH \rightarrow CM_7: E_{H(K_7)}(K_{78}, K_{58}), \; CH \rightarrow CM_8: E_{H(K_8)}(K_{78}, K_{58}, K_{18})$$

C. Member Eviction

Referring to Fig. 5, suppose CM_8 exits from the cluster of 8 members, CH and CM_1 to CM_7 compute the new cluster key K_{17}. Thus, the keys known to the exiting members are changed as shown below.

$$CH \rightarrow CM_1 \; to \; CM_4: E_{k_{14}}(K_{tmp}), \; CH \rightarrow CM_5, CM_6: E_{k_{56}}(K_{tmp}), CH \rightarrow CM_7: E_{H(k_7)}(K_{tmp}), K_{17} = K_{18} \oplus K_{tmp}$$

VII. RESULTS AND DISCUSSION

A. Performance Analysis

Here, we analyse the performance of our proposed schemes in terms of storage, computation and communication overhead and compare with WLKH scheme. Table below depicts the storage, computation and communication cost of the proposed schemes and existing scheme. In WLKH scheme, each CM stores its private key, sub-cluster keys and the cluster key for confidential communication and the storage cost is $\log_2 n$. In the proposed schemes, hierarchical key tree is replaced by binomial key tree and the number of keys stored at each CM varies from 2 to $\log_2 n$. Thus the storage cost in our proposed schemes is less compared to WLKH scheme.

In BLKH scheme, during key initialization, the leftmost member calculates all the keys on its own. Other CMs calculate some keys on their won and receive some keys from the CH. This requires n-1 encryptions and n-1 hash at CH and 1 decryption and 0 to $\log_2 n$ hash at CM. In BWLKH scheme all the keys are calculated by CH using hash value of the private keys of CMs and hence takes n encryptions and n-1 hash at CH and 1 decryption at CM which reduces the computational overhead at CM. For WLKH, ittakes $2n\log_2 n$ operations. The computation cost for all the three schemes is evaluated separately for initialization, join and eviction operations which is O(n) for WLKH and $O(\log_2 n)$ for BLKH and BWLKH schemes.

TABLE: STORAGE, COMPUTATION AND COMMUNICATION COST OF WLKH, BLKH AND BWLKH SCHEMES.

	WLKH	**BLKH**		**BWLKH**	
Storage cost	$\log_2 n$ keys	2 to $\log_2 n$ keys		2 to $\log_2 n$ keys	
Computation cost		CH	CM	CH	CM
1. Initialization	$2n\log_2 n$	n-1 encryptions n-1 hash	1 decryption 0 to $\log_2 n$ hash	n encryptions n-1 hash	1 decryption
2. Join	n	$\log_2 n$ encryptions 0 to $\log_2 n$-1 hash	1 decryption 1 hash	$\log_2 n$ encryptions 0 to $\log_2 n$-1 hash	1 decryption 1 hash
3. Eviction	$n+\log_2 n$	$\log_2 n$ encryptions	1 decryption	$\log_2 n$ encryptions	1 decryption
Communication Cost		CH	CM	CH	CM
1. Initialization	$n\log_2 n$	n-1 messages	----	n messages	----
2. Join	1	$\log_2 n$ messages	----	$\log_2 n$ messages	----
3. Eviction	$\log_2 n$	$\log_2 n$ messages	----	$\log_2 n$ messages	----

The communication overhead is high during key initialization for WLKH which is $O(n\log_2 n)$ and it is O(n) for BLKH and BWLKH schemes. In WLKH it is O(1) and $O(\log_2 n)$ for join and eviction respectively. It is $O(\log_2 n)$ for both join and eviction in our schemes. Graphs in Fig. 7 below show the comparison between the existing schemes and the proposed schemes in terms of number of new keys generated and the number of encryptions performed. Fig. 7a and 7b show the number of computations in LKH++ and BLKH schemes during join and eviction respectively. WLKH and BWLKH schemes are compared in Fig. 7c and 7d. It is evident from the graphs that BLKH and BWLKH schemes are efficient compared to LKH++ and WLKH schemes.

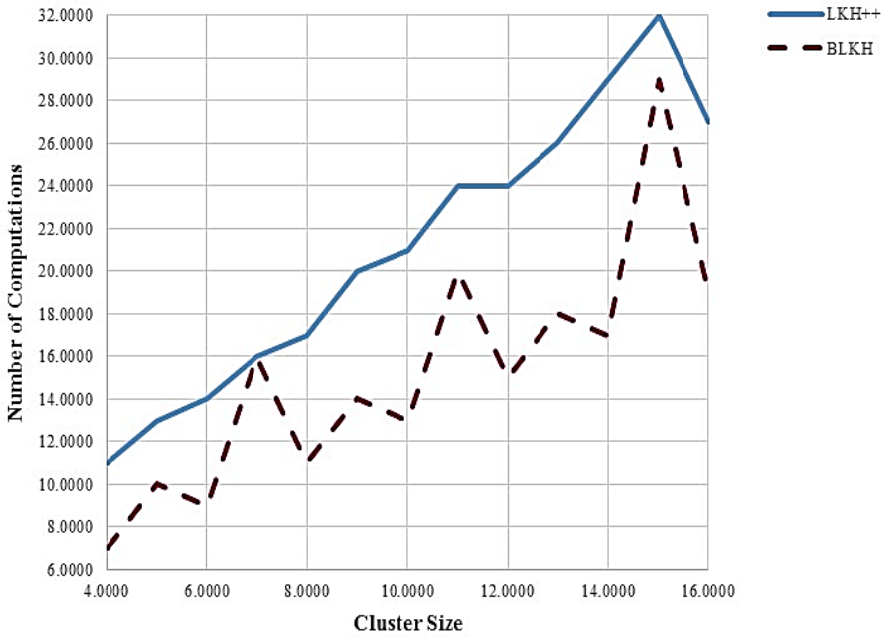

Fig. 7a. Computation Cost for Join

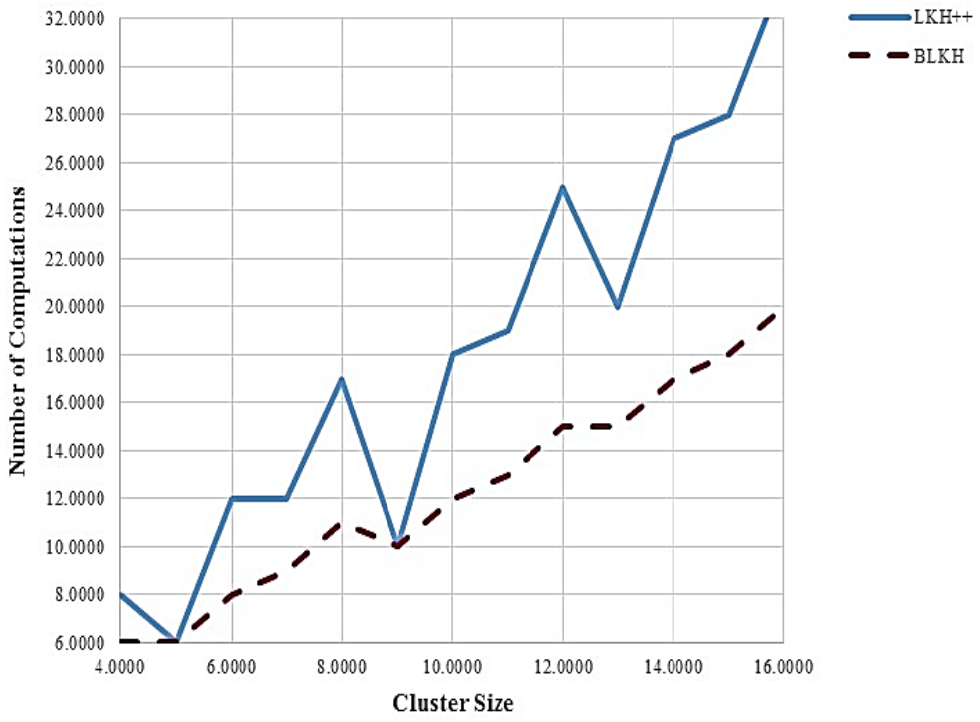

Fig. 7b. Computation Cost for Eviction

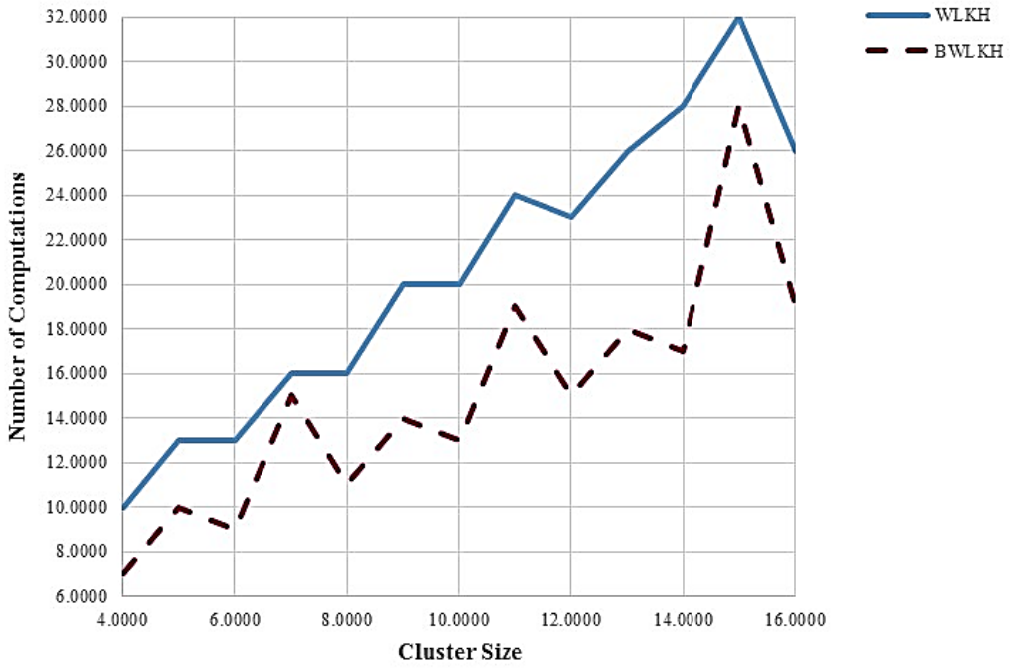

Fig. 7c. Computation Cost for Join

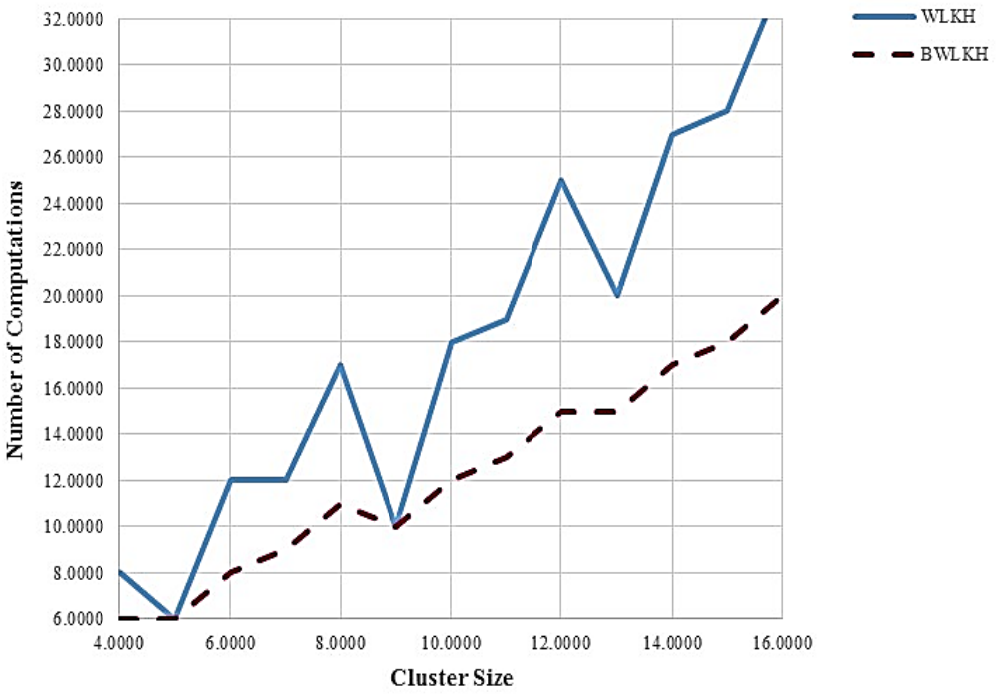

Fig. 7d. Computation Cost for Eviction

B. Simulation Results

We have simulated the cluster based SGC model using NS-2 to study the energy consumption of the proposed schemes and existing schemes. Fig. 8a and Fig. 8b below show the energy consumption for join and leave events respectively. The energy consumption of BLKH is almost same as that of BWLKH for both join and leave. WLKH scheme consumes more energy for both join and leave events compared to BLKH and BWLKH schemes as the number of new keys generated is more at both CM and CH whenever tree imbalance occurs during node addition/eviction. For instance, when the cluster size is 16 and a new member joins the cluster, 24 keys are generated in WLKH and 17 keys in BLKH and BWLKH. For the same cluster size, it is 30 keys in WLKH and 16 keys in BLKH and BWLKH for eviction. The tree will never get imbalanced in the proposed schemes since binomial key tree is used to store the keys and hence it is not necessary to rebalance the tree. It may require reconstructing the tree after join/leave event to restore binomial tree property.

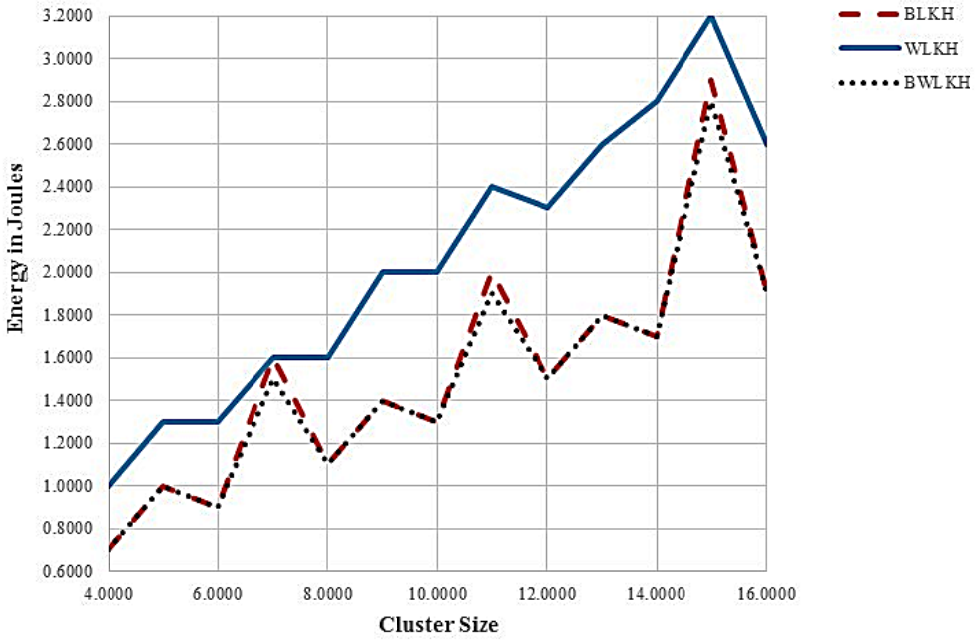

Fig. 8a Energy Consumption for Join

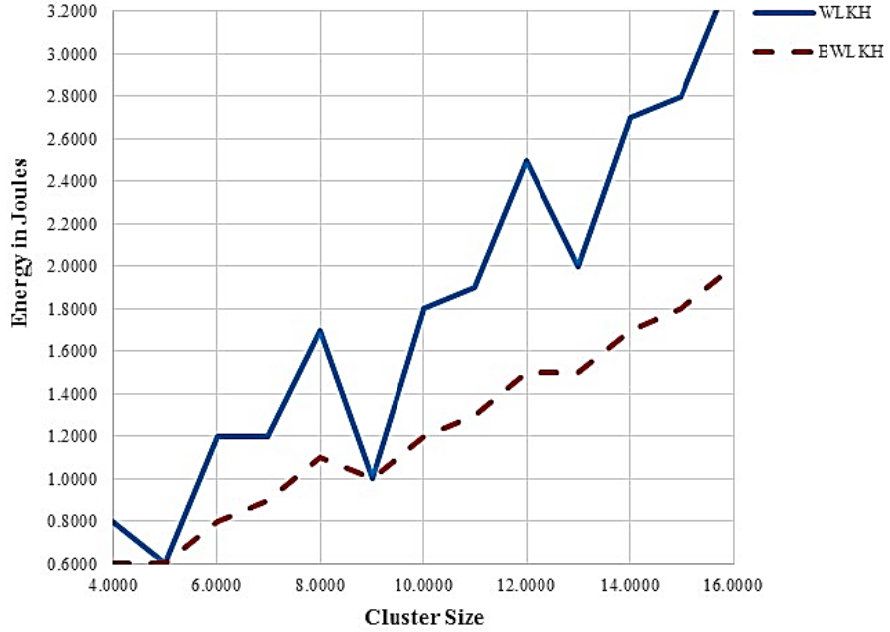

Fig. 8b Energy Consumption for Eviction

VIII. CONCLUSION

The two key management techniques proposed in this paper, BLKH and BWLKH are improvement over the schemes proposed by Roberto et al. and Wenbin et al. The proposed schemes improve the performance of the sensor nodes and reduce the storage at each node by replacing the logical key tree of [1] and [2] by binomial key tree. The binomial approach eliminates the need of rebalancing the tree during node join and eviction and thus reduces the computations at sensor nodes. The experimental results show that the performance of the proposed schemes is better than LKH++ and WLKH schemes in terms of storage and computations.

REFERENCES

[1] Di Pietro R, Mancini L. V. and Jajodia S, "Efficient and secure keys management for wireless mobile communications", Proceedings of the second ACM international workshop on principles of mobile computing,pp. 66–73, 2002.
[2] Wenbin Yao, Si Han, Xiaoyong Li, "LKH++ Based Group Key Management Scheme for Wireless Sensor Network", Wireless Personal Communications, 83, no. 4, 3057-3073, 2015.
[3] K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs", IEEE/ACM Transaction on Networking, 8(1), 2000.
[4] R. Aparna, B.B. Amberker,Divya Pola, Pranjal Bathia, "Secure Group Communication using Binomial Trees", Third IEEE InternationalSymposium on Advanced Networks and Tele commun. Systems(IEEE ANTS 2009), pp.142-144, 2009.
[5] Diop, Abdoulaye, Yue Qi, and Qin Wang. "Efficient group key management using symmetric key and threshold cryptography for cluster based wireless sensor networks." International Journal of Computer Network and InformationSecurity, 6, no. 8, 2014.
[6] Poornima, A. S., and B. B. Amberker. "A secure group key management scheme for sensor networks." In Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference , pp. 744-748. IEEE, 2008.
[7] Simplício Jr, M. A., Barreto, P. S., Margi, C. B., & Carvalho, T. C. "A survey on key management mechanisms for distributed wireless sensor networks",Computer networks, 54(15), pp. 2591-2612, 2010.
[8] Kausar, Firdous, Sajid Hussain, Jong Hyuk Park, and Ashraf Masood. "Secure group communication with self-healing and rekeying in wireless sensor networks." International Conference on Mobile Ad-Hoc and Sensor Networks,Springer, Berlin, Heidelberg,pp. 737-748, 2007.
[9] Perrig, A., Stankovic, J., & Wagner, D. Security in wireless sensor networks. Communications of the ACM, 47(6), 53-57, 2004.
[10] Eltoweissy, Mohamed, M. Hossain Heydari, Linda Morales, and I. Hal Sudborough. "Combinatorial optimization of group key management." Journal of Network and Systems Management 12, no. 1, pp. 33-50, 2004.
[11] Cho, J. H., Chen, R., & Wang, D. C, "Performance optimization of region-based group key management in mobile ad hoc networks". Performance Evaluation, 65(5), pp. 319-344, 2008.
[12] Zhu, Sencun, Sanjeev Setia, and Sushil Jajodia. "Performance optimizations for group key management schemes." In Distributed Computing Systems, Proceedings of 23rd International Conference,pp. 163-171. IEEE, 2003.
[13] Zhu, Sencun, Sanjeev Setia, and Sushil Jajodia. "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks." ACM Transactions on Sensor Networks (TOSN) 2, no. 4, pp. 500-528, 2006.
[14] Arundhati Nelli , Sushant Mangasuli, Manasa N. "Localized Encryption and Authentication Protocol for Secure Key Management in Wireless Sensor Networks", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, Special Issue 6, July 2017.
[15] Mansour, I., Chalhoub, G., & Lafourcade, P. "Key management in wireless sensor networks". Journal of sensor and actuator networks, 4(3), pp. 251-273, 2015.
[16] Eltoweissy, Mohamed, Ashraf Wadaa, Stephan Olariu, and Larry Wilson. "Group key management scheme for large-scale sensor networks." Ad Hoc Networks 3, no 5, pp. 668-688, 2005.
[17] Gu, H., & Potkonjak, M, Efficient and Secure Group Key Management in IoT using Multistage Interconnected PUF. In Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design (p. 8), July,2018.
[18] Danyang Qin, Shuang Jia, Songxiang Yang, ErfuWang, Qun Ding, "A Lightweight Authentication and Key Management Scheme for Wireless Sensor Networks", Hindawi Publishing Corporation, Journal of Sensors, Volume 2016, Article ID 1547963, 2016.
[19] Dimitris Tsitsipis, Anthony Tzes, Stavros Koubias, "CHAT: Clustered hierarchical key management for wireless sensor networks using network topology", International Journal of Distributed Sensor Networks,Vol. 13(11) , 2017.
[20] Zhan, F., Yao, N., Gao, Z., & Tan, G, "A novel key generation method for wireless sensor networks based on system of equations", Journal of Network and Computer Applications, 82, pp. 114-127, 2017.

AUTHOR PROFILE

H S Annapurna is currently working as Associate Professor in the department of Computer Science & Engg., Sri Siddhartha Institute of Technology, Tumakuru, Karnataka, India. She has obtained her Bachelor of Engineering from University of Mysore, Mysore, Karnataka, India. She has received Masters degree in Software Systems from BITS, Pilan, Rajastan, India. She is currently pursuing Doctoral degree in the area of Wireless Sensor Networks from Sri Siddhartha Academy of Higher Education, Tumakuru, Karnataka, India. She has published around 15 papers in international conferences and journals.

M Siddappa received B.E and M.Tech degree in Computer Science & Engineering from University of Mysore, Karnataka, India in 1989 and 1993 respectively. He has completed doctoral degree from Dr.MGR Educational Research Institute Chennai under supervision of Dr.A.S.Manjunatha, CEO, Manvish e-Tech Pvt. Ltd., Bangalore in 2010. He has worked as project associate in IISc, Bangalore under Dr. M.P Srinivasan and Dr. V.Rajaraman from 1993 – 1995. He has teaching experience of 26 years and research experience of 10 years . He has published 65 Technical Papers in National, International Conferences and Journals. He has citation index of 113 till 2015 and h-index of 4 and i10-index of 3 to his credit. He is a member of IEEE and Life member of ISTE. He is working in the field of data structure and algorithms, Artificial Intelligence, Image Processing and Computer Networking. He has worked as Assistant Professor in the Department of Computer Science &Engineering from 1996 to 2003 at Sri Siddhartha Institute of Technology, Tumakuru. Presently, he is working as Professor and Head, Department of Computer Science & Engineering at Siddhartha Institute of Technology, Tumakuru. He has visited Louisiana university Baton rouge and California university. He has received “Best Engineering Teacher Award” from ISTE in the year 2011.