Received May 14, 2019, accepted June 3, 2019, date of publication June 10, 2019, date of current version June 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2921999

Logical Tree Based Secure Rekeying Management for Smart Devices Groups in IoT Enabled WSN

MUHAMMAD ARIF MUGHAL[1,2,3], PENG SHI[4], ATA ULLAH[5], (Member, IEEE), KHALID MAHMOOD[5], MUHAMMAD ABID[6], AND XIONG LUO[1,2,3], (Senior Member, IEEE)

[1]School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 10083, China
[2]Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China
[3]Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China
[4]National Center for Materials Service Safety, University of Science and Technology Beijing, Beijing 10083, China
[5]Department of Computer Science, National University of Modern Languages, Islamabad 44000, Pakistan
[6]Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences, Islamabad 44000, Pakistan

Corresponding author: Peng Shi (shipengustb@sina.com)

This work was supported by the National Key R&D Program of China under Grant 2017YFB0803302.

ABSTRACT With the rapid growth in a huge number of devices connecting online, Internet of Things (IoT) is rapidly growing and getting interested of researchers. IoT enabled wireless sensor network (WSN) plays a significant role to collect sensing data and transmit to central repositories. Moreover, multicasting ensures efficient group communication for disseminating the same query or command to all smart devices to perform mobile service computing. It is applicable in the smart home, healthcare, smart cities, and smart industries for monitoring and control. To secure such sensitive information exchange, we have considered a secure group communication scenario where logical trees are maintained for each group. The main problem is unnecessary rekeying when a smart device frequently joining or leaving the network. It causes computation, communication, and energy overheads. To overcome the excessive rekeying problem, we have presented a logical tree-based secure mobility management scheme (LT-SMM) using mobile service computing in IoT. It includes the group deployment phase where smart devices securely setup a group by registering with group heads for future secure information exchange. We have presented group deployment, mobile node joining and mobile node migration protocols. Moreover, we have used chaotic map based one-way hash functions to ensure message integrity. To validate our work, extensive simulations are performed using NS 2.35. TCL code is used to configure smart devices, deploy logical tree, messaging. C language is used for algorithm implementation and messaging backend coding. The results verify the supremacy of our scheme as compared to existing tree based schemes in terms of computation, communication, and energy consumption.

INDEX TERMS Internet of Things (IoT), multicasting, rekeying, wireless sensor networks (WSN).

I. INTRODUCTION

Internet of Things (IoT) comprises of a large number of sensing devices that can communicate with each other and across the network to exchange data for storage and future analysis. IoT is applicable in a wide variety of scenarios in our daily life for smart sensing in healthcare, smart homes, smart cities, agriculture, smart industries and internet of vehicles [1]–[4]. Wireless Sensor Networks (WSNs) are considered to be future key contributor to IoT for smart device level sensing, computation, caching and communication with limited resources [5]. In this respect, heterogeneous IoT can be the future direction for mobile service computing to resolve the bottlenecks due to huge number of smart sensing devices working individually or in a group [6], [7]. In IoT, smart devices can be exposed to large number of attacks due to limited power, storage, bandwidth, mobility and sensing bottleneck. Secure key management is mandatory and quite challenging to guard against these security attacks and exchange information by ensuring confidentiality, integration, authentication and non-repudiation [8]. Machine learning has been widely adopted in different areas and showed good performance [9], [10]. It is also an available solution for IoT data analysis. In the recent years, chaos theory based security is getting growing interest. Chaotic map based one-way hash functions are quite reliable for ensuring

The associate editor coordinating the review of this manuscript and approving it for publication was Wenbing Zhao.

message integrity [11]. Due to mobility, node joining or leaving the group results in re-establishment of keys. It ensures that leaving or joining nodes are unable to get communication beyond their access. Moreover, it is quite essential to keep an updated record of member devices to multicast the message towards member nodes.

Multicasting in group based deployment is considered to be efficient for message dissemination among group members through a group controller instead of individually device-to-device level messaging. It is quite beneficial in IoT enabled smart application scenarios with mobile service computing where group controllers are connected with multiple smart devices in a group. It can deliver same query or command to all sensors in case of smart home, smart cities, and smart industries for innovative applications. Literature includes schemes for secure group communication but efficient multicasting solution is still awaited in smart devices groups [12], [13].

The main problem is that in logical tree based approaches for multicasting, a single mobile node leaving the group can result in re-keying of all the nearby mobile nodes in the subtree. Rekeying is mandatory to ensure forward and backward secrecy where newly joining mobile node should not be able to access previous communication of nearby nodes in the cluster. Rekeying may occur in short range mobility when node is moved from one parent to the other within small range. It becomes worse when mobile nodes are frequently moving around in a small region. Rekeying overheads hinder the scalability of group and requires additional computational cost. However, mobile service computing ensures the accessibility of services anywhere and anytime that should be achieved by resolving the above mentioned problem.

This paper presents a Logical Tree based Secure Mobility Management (LT-SMM) Scheme to provide rekeying efficiently when new node joins or existing node leaves the group. Our main contributions are as follows;

1) Three protocols are proposed as follows; i) Group deployment protocol for group formation; ii) Node joining protocol that manages new node addition in the tree for a particular group; iii) Node migration protocol that handles short-term or long-term migration of node from one group to other neighboring or distant group.

2) In such environments mobility is found frequent as nodes are mobile and is able to leave or join any group at any time. The main focus of the work is to eliminate unnecessary rekeying by maintaining the demanded security strengths in logical tree oriented schemes to effectively avail benefits of mobile service computing.

3) Moreover, we have ensured the message integrity by using chaotic map based one-way hash functions. We have simulated our work using NS-2.35. Tool Command Language (TCL) is used for deployment, mobility patterns, node configuration and message initiation for sensor and H-sensor. Moreover, C language is used for hashing, encryption, decryption and logical tree management.

4) Simulation results prove the dominance of LT-SMM over counterparts in terms of communication, energy consumption, resilience and rekeying cost.

The rest of the paper is organized as follows: Section 2 explores system model and literature review is presented in section 3. The proposed LT-SMM is presented in section 4. Results and analysis along with simulation environment are presented in section 5. Section 6 concludes our work and discusses about future work.

II. SYSTEM MODEL FOR MULTICASTING MESSAGES

In this section we present a system model for multicasting the cloud or FoG server initiated messages, queries or commands to multiple groups. We have discussed about two types of groups including group of sensor nodes and group of patients. The former, comprises of low power smart sensing devices or nodes and cluster head (CH) with more computation power, memory and battery life. The latter, includes the set of medical sensors for each patient where cell phone act as group controller (GC). In this case, the mobile device contained by doctor or hospital is considered as multicast controller (MC) that avails mobile service computing. In IoT enabled sensor network, cloud or FoG servers can multicast the queries and commands towards GCs. In the similar vein, CH act as GCs and sink server act as MC for group of sensor nodes as illustrated in Figure 1. We have assumed that the sink server has capability to handle request and response from these both categories of groups and even can support more type of groups as well in future IoT scenario. Sink can communicate with FoG server for quick communication to improve delays due to waiting time for cloud response. The FoG server can communicate with cloud when mandatory to exchange information.

In the group, it is assumed that the CH has ability to directly communicate with its group members by adopting routing algorithm. Due to mobility, topological tree structure also changes, resulting in modifying routing tables as well. In addition, the pair-wise keys and group keys are not pre-distributed as nodes and CHs are mobile and each group is randomly deployed. The groups have flexibility in their nature that a group member may leave or join any group. All communications take place with the involvement of CHs either it is intra-cluster or inter-cluster scenario. Similarly, in group of patients, the medical sensors can also be disconnected from GCs and multiple patients can also leave and join the MC for receiving intended multicast messages.

In our proposed model, logical tree is maintained at sink, MC and CHs. It updates the tree of mobile nodes after node joining and leaving operations. Moreover, key update procedure is also performed for rekeying. CHs transmit messages to all mobile nodes in the group but response is generated by the nodes that match the query requirements initiated by the requester. The proposed IoT enabled group-based WSN involves two data communication layers including Node-to-CH layer and CH-to-Server layer for mobile services

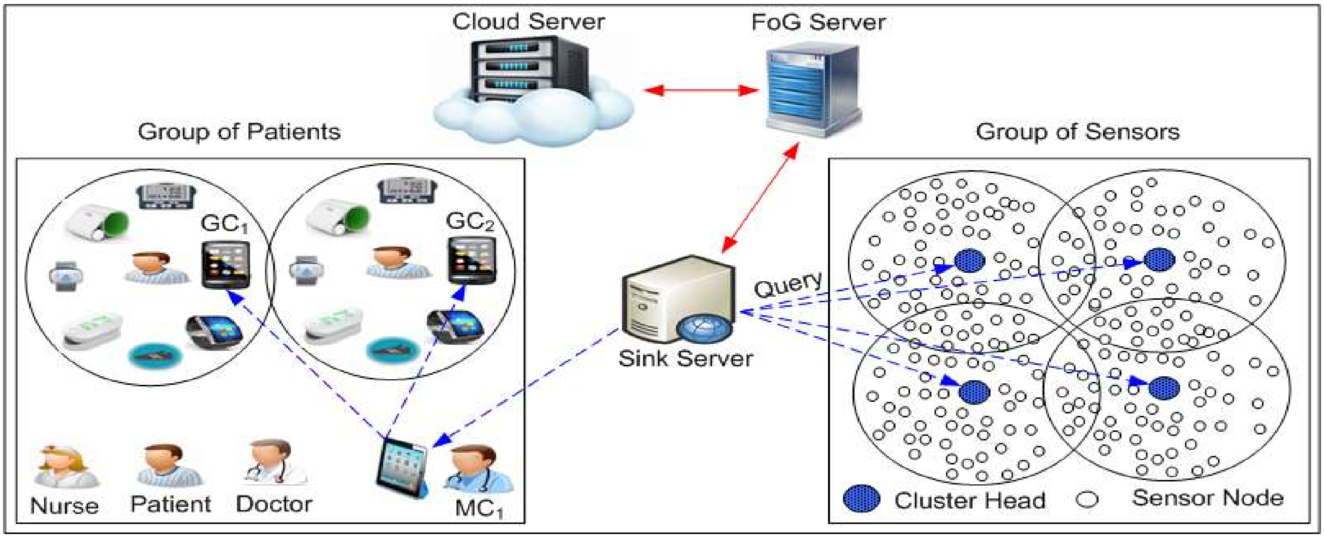


FIGURE 1. LT-SMM System Model for Multicasting messages for group of sensors via Cluster Heads (CHs) and for group of patients via group controllers (GCs) and multicast controller (MC).

computing to ensure wide connectivity via FoG server and cloud service providers.

III. RELATED WORK

We have explored the existing schemes that focus on logical tree based key establishment scenarios. The main concern is to maintain a logical tree for neighboring mobile devices to highlight the key establishment hierarchy. It also illustrates the degree of a node to show its set of linkages. Our main focus is to explore the presented schemes and identifying the changes that occur during key management when a node leaves the group or the new node joins the group. Topological Key Hierarchy (TKH) scheme maintains a topology based key-tree to manage rekeying but communication and computations costs increases with group size [14].

A highly scalable multi-group keying for IoT is proposed to ensure the backward and forward secrecy. It also ensures the co-existence of variety of related services in the same region along with independent security parameters [15]. Kung and Hsiao have presented a lightweight scheme for group key management in dynamic IoT scenarios. It utilizes two-tiers where one device can be the member of multiple groups and hence establishes multiple keys. It prevents from taking unnecessary information from other devices in case of a malicious user. The scheme also ensures forward and backward secrecy along with guarding against collusion attacks [16], [17]. Zhou and Yi found that there are two fundamental boundaries on this scheme: i) μTESLA calls for synchronization of nodes which may be very difficult to gain in WSNs, ii) the scheme does not provide an explanation for how the authentication technique is performed and it gives a conversation overhead [18].

The self-organized Group Key Management (GKM) protocol includes the logical key encryption key (KEK) tree. Every participant must be able to compute all the tree secret keys from the leaf node where it is placed to the root of tree. Consequently, each member needs to know both its key and the blinded keys of each sibling node in its path to the root. A weight is maintained at each node that includes depth of the tree, weight of tree and node value. This weight is unique value to identify the affiliation precedence of the member node with head. Different rules of weight challenge may additionally have an effect on the network [19].

A pair-wise key is needed for hop-to-hop communication for which an extra agreement procedure required. The pair-wise key procedure executed each time whenever any node changes its tree location. Cheikhrouhou *et al.* has presented the ring based secure group communication (RiSeG) schemes [20], [21]. It involves resource constrained controller. The concept of the scheme is to divide the group controller undertaking among group participants by means of constructing logical ring structure. This logical ring allows delivering the rekeying messages as follows. Firstly, the group controller sends the rekeying message to the subsequent hop within the ring after which the message is forwarded from a node to any other till the message re-turns lower back to the group controller. Secondly, the logical ring is replaced in case of joining/leaving. The scheme reduces the conversation, computation and garage price on the organization controller, however, it introduces a huge latency that cannot scale with a huge organization.

N. Ferrari *et al.* have presented the lightweight scheme for IoT devices to setup group key. It involves Elliptic Curve Cryptography along with one-way accumulators to prove the betterment in terms of reducing energy consumption, computational and communication costs [22]. In [23], the energy-level of nodes is considered to distribute the key generation burden among nodes accordingly. There are two roles in key management, first is Group Leader (GL) and second is Group Member (GM) that is simple node. It is observed that in

EADGH, CH is the last node in the group it means that it is at highest index. CH receives energy level of each GM periodically and decides the index for each GM. This phase is called numbering strategy. The numbering strategy mainly depends upon the size of the cluster.

M. Garcia *et al.* has presented a cooperative group-based scheme to resolve the limitation in [23] that each node requires $i + 1$ multiplication operations. It keeps the CH and member nodes busy for sharing messages about energy factors [24]. In [25], a tree-code modeling based scheme is presented where the addressing mechanism for non-ID physical objects is also considered in IoT scenario. In [26], a logical neighbor tree (LNT) based secure group communication scheme where sensor nodes belonging to the same group can communicate securely. It includes secure group creation, joining and leaving processes. It also presents a method to update the key after each membership change for the sake of guaranteeing forward and backward secrecy properties.

In [27], an innovative batch-based group oriented key management scheme is presented. It is useful in IoT scenario with a Key Distribution Center (KDC) that plays a major role for authenticating devices. It is a group key is distributed and data is encrypted with secret group key. Therefore, it requires suitable group key distribution mechanism and an efficient method to distribute a new key upon every group member change. Trusted KDC is responsible for maintaining secure association with all users and generates new group key every time the group membership changes and distributes new group key to all group members. Dini and Savino [28] have presented a secure and scalable rekeying technique for WSSN.

Abdmeziem and Charoy presented a Decentralized Batch-based Group Key (DBGK) scheme that involves several sub-groups which are managed by area key management server. Moreover, the whole group is managed by the general keying server. In this scheme, group key is composed of long-term and short term keys. Similarly, security credentials are shared with member nodes as per availability of resources including storage, computation and residual energy [29]. But scheme is improved to reduce the overhead of communicating with two servers. It uses Distributed Batch-based Group Key (DsBGK) scheme that utilizes polynomial based computations to setup the key among collaborative groups in IoT environment. A number of dynamic members that can leave or join the group are quite challenging to protect backward and forward secrecy especially for large groups. It also considers the heterogeneity of the devices with multiple capabilities under IoT enabled sensing networks [30].

Existing schemes keep track of leaving and newly joining nodes and generate the new keys appropriately to ensure reliable multicasting in groups. In [1], [10], [12], and [30], each group is divided into smaller sub-groups or logical sub-trees as depicted in Figure 2. In these schemes, three major issues are noticed as follows;

1) Every time neighbor detection procedure is performed whenever a node joins or leaves.

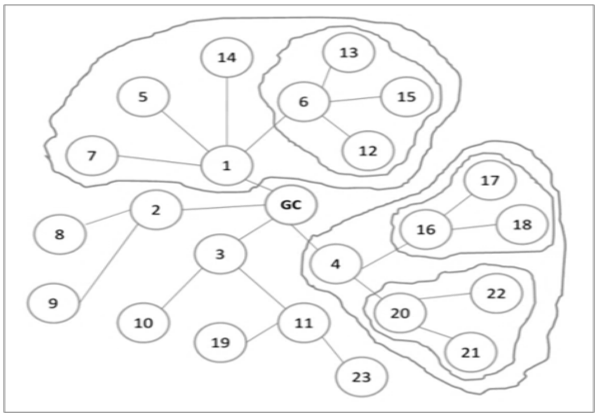


FIGURE 2. Un-necessary Rekeying in sub-trees due to minor mobility.

2) Parent-child selection procedure is executed on change.

3) Rekeying functions are computed and keys are updated in the whole group on each joining/leaving operation. Rekeying is also performed whenever node moves from one sub-group to another sub-group within the main-group.

In existing schemes, multi-hop communication for rekeying causes communication overhead at GC and intermediate senor nodes. It results in a number of issues enumerated as follows;

1) Nodes communicates with sink before joining GC [30].
2) More complex tree structure is adopted to setup the groups in hierarchical architecture [19], [20], [30].
3) Inter-cluster communication is not supported [30].
4) In [1], [19], [30], LNT maintained on GC causes computation and memory overheads.

IV. LOGICAL TREE BASED SECURE MOBILITY MANAGEMENT SCHEME (LT-SMM)

In this section, we present a secure mobility management scheme for handling the multicast in smart devices groups. In proposed LT-SMM, following two major points are considered: i) Rekeying – is performed frequently in each of the scheme, which obviously effects computation, memory and communication overhead. ii) Mobility Management – for mobile nodes, provide security to manage joining and leaving operations. In our case, CHs maintain logical tree for a group of mobile nodes to manage node leaving and joining operations. CHs also share these updates with sink. The proposed scheme thoroughly explores nodes deployment scenario along with node joining and node migration protocols. The scheme has positive impact on secrecy, communication, computation and energy measures. During rekeying, server maintains the record of active nodes and their links to better manage the multicast scenario. It also ensures the forward secrecy by confirming that a group participant should not be able to read any future conversation from its existing key once it leaves the network. Moreover, backward secrecy is ensured by confirming that if a new member joins the network, it remains not able to discover the preceding

TABLE 1. List of notations for LT-SMM.

Notation	Description
BS	Base Station
CHj	Group Head
Ni	Mobile Device
ts_{Ni}	Time Stamp at Ni
Δt	Time difference
ID_j	Identity of CHj
E_K	Encryption Key
$\otimes$	XOR
K_{CHj-Ni}	Key between CHj and Ni
$\|\|$	Concatenation
n_j	Nonce Value at CHj
H(M$_1$)	Hash of Message M1

conversation from its current assigned key. A list of notations for LT-SMM is provided in Table 1.

A. GROUP DEPLOYMENT

During the deployment, smart sensing devices are deployed as members in multiple groups where a powerful device is also deployed as group head CH_j. It is assumed that a symmetric key is loaded offline in each of the mobile device N_i and CH_j. The CH_j is responsible to initiate all the member smart devices to join the group. For this purpose, CH broadcasts an encrypted joining message along with some security parameters by encrypting using pre-established keys. Group deployment protocol is illustrated in Protocol-I and stepwise description is explored as follows; Cluster head CH_j initiates message $(ID_j||n_j||ts_j||JOIN_{REQ})$ encrypted using the pairwise symmetric key K_{CHj-Ni} between CH_j and N_i. In the message, represents identity of CH_j, n_j *is* a random nonce value, ts_j is time stamp and $JOIN_{REQ}$ is request from CH_j to join the group in a secure manner. Moreover, a chaotic map based one-way hash $H(M_1)$ is also included to guard against bit alteration of message modification attacks. In the similar vein, timestamp is included to prevent from replay attacks. For replying to CH_j, each member smart device N_i first decrypts the message and then transmits the message $(ID_{Ni}||n_j||ts_{Ni})$ encrypted with its pre-established symmetric key K_{Ni-CHj}. Reply message consists of node IDi concatenated with nonce and a new time stamp $ts_{Ni}{}'$. All the nodes relating the group reply in same regards on receiving group join message broadcasted by CH_j.

On receiving reply message from node N_i, CH_j calculates the time stamp after decryption. If the difference between time stamps $ts_{Ni}^{'} - ts_{Ni}$ is less than threshold Δt then CH_j computes step 4. Otherwise, joining is failed and the message is discarded. In step 4,CH_j compares with nonce, if the nonce is equal to the received nonce then CH_j performs the following steps otherwise, the message is discarded. CH_j sets the status of the node N_i as member node. After listing the node in members list, CH_j sends joining success message encrypted with symmetric key to N_i. After successful join procedure, N_i transmits data securely in the network to node N_i.

Protocol 1 Group Deployment Protocol

```
$CH_j \rightarrow * : E_{K_{CHj-Ni}}\{M_1 = (ID_j||n_j||ts_j||JOIN_{REQ}), H(M_1)\}$
Each Ni verify freshness and Hash Value for Message Integrity
$N_i \rightarrow CH_j : E_{K_{Ni-CHj}}\{M_2 = (ID_{Ni}||n_j||ts_{Ni}), H(M_2)\}$
$CH_j$ :
  If ($ts_{Ni}^{'} - ts_{Ni} < \Delta t$) then
    If ($n_j$ equals $n_j$ _received) then
      Add to node_list and set status as member
      $CH_j \rightarrow N_i : E_{K_{CHj-Ni}}\{M_3 = (ID_j||SUCCESS), H(M_3)\}$
    Else
      Join Failed due to Nonce Mismatch and Message Discarded
    End if
  Else
    Join Failed and Message Discarded due to Freshness Expiry
  End if
```

B. NEW NODE JOINING PROTOCOL

Each new node is loaded with chaotic map based hashed *PIN* code P_{Code} that is used to compute the key between CH and the node N for the secure communication. In the proposed scenario, let node Ni want to join a group with CH_j as cluster head. A detailed step-by-step description of node joining protocol is explored in Protocol-II. Initially, BS loads a hashed PIN code into a new node to compute key between CH and new Node i. The node that wants to join the group initiates the joining procedure. In this regard, N_i sends joining request to CH_j concatenate with itsidentity ID_{Ni}, time stamp ts_{Ni} and hashed PIN code P_{Code} pre-loaded by BS. Upon receiving joining request from Ni, CH_j checks it's ID_{Ni} in its rejected nodes list maintained by itself. If node id ID_{Ni} is not found in the list then CH_j requests PIN code form BS for this node. The PIN code request message $PCode_{REQ}$ is encrypted with symmetric key between BS and CH_j. The message includes IDj,ID_{Ni}, time stamp and PIN code request. If node id ID_{Ni} is found in the rejected list, CH_j discards the message consequently.

The encrypted message consisting ID_{Ni} and relevant P_{Code} is sent back to CH_j from BS as a reply of PIN code request. In this step, CH_j computes chaotic map based hash of the PIN code received and verifies that if the computed hash P'_{Code} is equal to the hash P_{Code} provided by the node Ni then it performs the following two operations a and b. In other case it discards the message simply. The cluster head CH_j generates a nonce n_{CHj} and obtains the key K_{CHj-Ni} by taking XOR of chaotic map based hash values of node id ID_{Ni}, P_{Code} and concatenated value of $n_{CHj}||ts_{Ni}$. The cluster head CH_j also sends key generation request $KeyGen_{REQ}$ to node Ni. Key generation request message has ID_j and nonce n_{CHj}.

After the successful generation of key, N_i replies to CH_j by sending a reply message encrypted with the newly generated key. This message contains ID_{Ni}, nonce, and key generation reply $KeyGen_{REP}$. It is mandatory to enlist the node N_i in member list, acknowledge it with successful join and register it with BS for future conversation. In this step, CH_j first verifies that the nonce is same as sent by CH_j. After that, following three tasks are performed. In cases when nonce is not verified, the message is discarded and the node is enlisted in the black list. CH_j adds the node Ni into member node list. After adding the node in member list the CH_j sends join success message encrypted with symmetric key to node N_i. After the successful join procedure the N_i can transmit data securely in the network through CH_j. In this same step, CH_j also sends an encrypted message of key generation request to BS for this node.

Protocol 2 New Node Addition Protocol

```
    $N_i \rightarrow CH_j : E_{K_M}\{M_3 = (ID_{Ni}||JOIN_{REQ}||ts_{Ni}||P_{Code}), H(M_3)\}$
  *CHj:*
  ***If*** *$ID_{Ni}$ NOT in rejected_node_list* ***then***
    $CH_j \rightarrow BS : E_{K_{CHj-BS}}\{M_4 = (ID_{CHj}||ID_{Ni}||ts_{CHj}||PCode_{REQ}), H(M_4)\}$
  ***Else*** *Discard the Request, considered as malicious node*
  ***End If***
  *BS: Verify freshness and Hash Value for Message Integrity else discard message.*
  $BS \rightarrow CH_j : E_{K_{BS-CHj}}\{M_5 = (ID_{BS}||K_M||ts_{BS}||PCode'), H(M_5)\}$
  *CHj:*
  *Verify message freshness and Integrity else discard message*
  ***If*** *$P_{Code}$ equals $P'_{Code}$)* ***then***
        $K_{CHj-Ni} = h(ID_{Ni}) \oplus h(PCode) \oplus h(n_{CHj}||ts_{Ni})$
        $CH_j \rightarrow N_i : E_{K_M}\{M_6 = (ID_{CHj}||KeyGen_{REQ}||n_{CHj}), H(M_6)\}$
  ***Else***
        *Discard Message due to Pin-Code Mismatch*
  ***End If***
  $N_i \rightarrow CH_j : E_{K_{Ni-CHj}} M_7 = (ID_{Ni}||n'_{CHj}||ts_{Ni}||KeyGen_{REP}), H(M_7)\}$
  ***If*** *$n_{CHj}$ equals $n'_{CHj}$* ***then***
        *Add to node_list*
        $CH_j \rightarrow N_i : E_{K_{CHj-Ni}}$ *("JOIN_SUCCESS")*
        $CH_j \rightarrow$ *BS:* $E_{K_{CHj-BS}}(ID_{CHj}$ || *nonce* || *KEY_GEN_REQ)*
  ***Else***
  *Discard the Message and add to rejected_node_list*
```

C. NODE MIGRATION PROTOCOL

Mobility management is the major direction of this research work. As described earlier, in the deployment scenario that the sensor nodes are mobile in nature and can move from one place to the other. Due to the mobility nature of sensor nodes, the group joining and leaving operations occurs frequently. For the secure migration of sensor nodes from one group to other group a novel node migration protocol-III is proposed. This protocol is carefully designed to provide forward secrecy and backward secrecy both in an efficient manner. It eliminates the requirement of rekeying on each group change. The previously computed key for each node securely used to authenticate the node before joining new group, and the previous group is also informed to update its member node list. It is assumed that the node presence or absence in the group checked periodically by the CH_j by sending beacon signals. If the node is no more available in the group then, CH_j can remove that node from the list. Moreover, list is updated when list update message is received from BS due to mobility of N_i from one group to other. In node migration protocol it is assumed that the node N_i moves outside from the range of CH_j and receives strong signals from another neighboring cluster head CH_n. Therefore, the node N_i now wants to join new group controlled by CH_n. The protocol is depicted briefly in Figure 3, and detailed step-by-step description is provided in protocol-III.

Protocol 3 Node Migration Protocol

```
 N$_i$ → CH$_n$: (ID$_i$|| ID$_j$|| JOIN_REQUEST || Ts)
 If (ID$_i$ not in rejected_node_list)
 CH$_n$ → BS: EK$_{CB}$ (ID$_n$|| ID$_i$|| ID$_j$|| KEY_REQUEST || Ts)
 BS → CH$_n$: EK$_{BC}$ (ID$_i$|| PRE_KEY)
 CH$_n$ → N$_i$: K$_{CNi}$ (ID$_j$|| nonce || SEND_MSG_REQ)
 N$_i$ → CH$_n$: K$_{NiC}$ (ID$_j$|| nonce || Ts)
 If (Ts' – Ts) < Δ t
 If(nonce == nonce_received)
       Add Node in node_list of CH$_n$
       CH$_n$ → N$_i$: EK$_{CNi}$("JOIN_SUCESS_MSG")
       CH$_n$ → BS: EK$_{CB}$ (ID$_n$|| ID$_i$|| TREE_UPDATE_REQ)
    Else
       Join Failed and Message Discarded
    BS → CH$_j$: EK$_{BC}$ (ID$_i$|| TREE_UPDATE _REQ)
    CH$_j$ → BS: EK$_{CB}$ (ID$_i$|| UPDATE_SUCCESS)
 Else
    Discard Message from Rejected Node
```

Steps (1) – (4): In case of node migration from one group to other group, the group joining initiative is taken by the sensor node N_i when it receives strengthen signals from cluster head CH_n supposed. The same request is sent by the node N_i to new cluster head CH_n as in node join protocol but with ID$_j$ (id of the previous cluster head) this time. In step 2, upon receiving the join request from node N_i, the new cluster head CH_n examines its rejected node list first. If ID$_i$ is found in the list, message is discarded. However, if ID$_i$ is not found in rejected node list then CH_n requests for key of node N_i. Next, CH_n sends encrypted request message concatenating ID$_i$, ID$_j$, ID$_n$ and time stamp. After that, BS sends an encrypted message

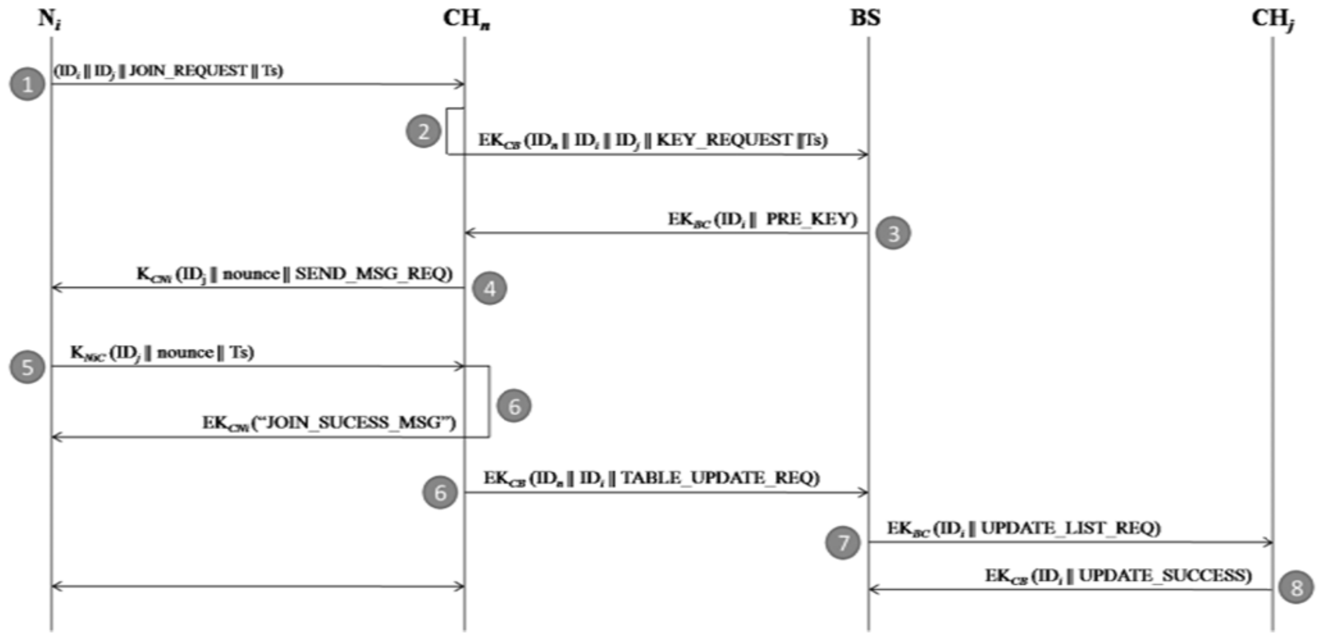

N$_i$
CH$_n$
BS
CH$_j$
1
(ID$_i$ ‖ ID$_j$ ‖ JOIN_REQUEST ‖ Ts)
2
EK$_{CB}$ (ID$_n$ ‖ ID$_i$ ‖ ID$_j$ ‖ KEY_REQUEST ‖Ts)
3
EK$_{BC}$ (ID$_i$ ‖ PRE_KEY)
4
K$_{CNi}$ (ID$_j$ ‖ nounce ‖ SEND_MSG_REQ)
5
K$_{NiC}$ (ID$_j$ ‖ nounce ‖ Ts)
6
EK$_{CNi}$ (“JOIN_SUCESS_MSG”)
6
EK$_{CB}$ (ID$_n$ ‖ ID$_i$ ‖ TABLE_UPDATE_REQ)
7
EK$_{BC}$ (ID$_i$ ‖ UPDATE_LIST_REQ)
8
EK$_{CB}$ (ID$_i$ ‖ UPDATE_SUCCESS)

FIGURE 3. Node migration protocol.

to CH_n containing ID$_i$ and its pre-established key. In step 4, cluster head CH_n sends a message to node N_i encrypted with the key received from BS. The message contains ID$_j$, nonce and send request.

Steps (5) – (8): The node N_i replies to CH_n in response to send message request. This reply message is encrypted and contains ID$_j$, nonce and time stamp (Ts’). Step 6 describes that CH_n now calculates time stamp and verify the equality of nonce. If the computation not fulfills both conditions, then join process becomes failed and message is discarded as a result. In opposite case the protocol performs these tasks. The node N_i is added to the member list of CH_n and the encrypted join success message is sent to N_i. At the same time CH_n also forwards table update request to BS. The request is encrypted and has ID$_n$ and ID$_i$. After the successful join procedure the node N_i becomes able to communicate data securely in the network through $CH_{j.}$. Next, the node migration procedure is not completed yet, some updates are still there. Now the BS station sends an encrypted message with ID$_i$ to previous cluster head CH_j to update its member list. Step 8 shows that a message is received to BS as reply of update request from CH_j. The reply message is encrypted and has ID$_i$ along with update success message.

V. FORMAL MODELING AND ANALYSIS

In this section, we have performed formal modeling for our LT-SMM scheme to verify and analyze it using Non-Monotonic Cryptographic Protocol (NCP) which is also known as Rubin Logic [31]. It is a standardized formal mechanism to benchmark and verify the essential requirements of security protocols and cryptographic operations. It can ensure the mandatory steps required to perform a certain security function at sender and receiver as well. Formal modeling assists to figure out the deviating steps in the proposed protocol scenario. It also helps to identify the potential outcomes concerning security attack scenarios by comparing with intrinsically standardized steps. It is near to the actual implementation in a programming language. A global set is defined that contains entities, their roles along with globally accessible variables of the modeled protocol. It keeps the information in sets and refreshes the states on the clients after each updateable operation. A local set is maintained at every element that also contains subsets including possession set POSS(), belief set BEL(), seen and behavior list BL(). A structure of local Set for proposed LT-SMM scheme is presented in Table 3.

It includes the detailed stepwise description during message exchange between Sender N_i, CH_j and BS. The fundamental implementation level operations including concatenation of parameters, hash values, encryption and decryption are performed before sending a message. Update operations are performed after sending the message to memorize the newly calculated values. On receiving side, the basic operations are decryption of cipher text, check message freshness using timestamp, compare and verify hash for message integrity. A possession set like POSS (N) maintains the list of commonly used variables associated with message encryption, decryption and related operations at N$_\mathrm{i}$. Similarly, sets are maintained at other entities including POSS(CH$_\mathrm{j}$) and POSS(BS). A Behavior List BL(N_i) holds details regarding cryptographic operations and information exchanging operations that are performed in close to execution schemes at N_i. Similarly these sets are maintained including BL(CH_j), and BL(BS) at participating entities CH_j and BS respectively.

TABLE 2. Local set at sender, CH_j and BS.

1. Sender (N_i)
POSS(N_i) = { ID_{N_i}, K_M, $K_{N_i-CH_i}$} BEL(N_i) = { #(ID_{N_i}), #(K_M) #($K_{N_i-CH_i}$)} BL(N_i) = Concat(ID_{Ni}, $Join_{REQ}, ts_{Ni}, P_{Code}$) →$P_{N_i}$ Hash(h(.); P_{N_i}) → H_{N_i} Encrypt({ ID_{N_i}, $Join_{REQ}, ts_{N_i}, P_{Code}, H_{N_i}$ }$K_{N_i-CH_j}$) → C_1 Send(CH_j, { ID_{N_i}, C_1}) → M_1, Update ($M_{ID=1}$)
2. Group Head (CH_j)
POSSCH_j () = { ID_{CH_j}, $K_{CH_{j-N_i}}$, $K_{CHj\text{-}BS}$ } BEL(CH_j) = {#(ID_{CH_j}), #($K_{CH_{j-N_i}}$ -), #($K_{CHj\text{-}BS}$) } BL(CH_j) = Receive(CH_j, { ID_{Ni}, C_1}) and Split({ ID_{Ni}, C_1}) Decrypt({ C_1 }$K_{CH_i-N_i}$) to get [ID_{Ni}, $Join_{REQ}, ts_{Ni}, P_{Code}, H_{N_i}$] MsgFreshness ($ts'_{Ni} - ts_{Ni}$) ≥ Δt if true then Msg is aborted LookUpRejectedList(ID_{Ni}) if true then Msg is aborted Hash({Concat(ID_{Ni}, $Join_{REQ}, ts_{Ni}, P_{Code}$)}) →$H^*$ Verify(H_{N_i}, H^*) if mismatch, then abort Concat($ID_{CH_j}, ID_{Ni}, ts_{CHj}, PCode_{REQ}$) →$P_{CH_j}$ Hash(h(.); P_{CH_j}) → H_{CH_j} Encrypt({$ID_{CH_j}, ID_{Ni}, ts_{CHj}, PCode_{REQ}, H_{CH_j}$}$K_{CHj-BS}$) → C_2 Send(ID_S, { ID_{CH_j}, C_2}) → M_2 Update($M_{ID=2}$) Receive(CH_j, { ID_{BS}, C_3}) and Split({ ID_{BS}, C_3}) Decrypt({ C_3 }K_{CH_j-BS}) to get [$ID_{BS}, K_M, ts_{BS}, P_{Code}', H_{BS}$] MsgFreshness ($ts'_{BS} - ts_{BS}$) ≥ Δt if true then Msg is aborted Hash({Concat($ID_{BS}, K_M, ts_{BS}, P_{Code}'$)}) →$H^+$ Verify(H_{BS}, H^+) if mismatch, then abort Verify(P_{Code}', P_{Code}) if mismatch, then abort $Get\ K_{CH_{j-N_i}} = h(ID_{Ni}) \oplus h(P_{Code}) \oplus h(n_{CHj} \Vert ts_{Ni})$ Concat($ID_{CH_j}, KeyGen_{REQ}, n_{CHj}$) →$P^{\wedge}{}_{CH_j}$ Hash(h(.);$P^{\wedge}{}_{CH_j}$) → $H^{\wedge}{}_{CH_j}$ Encrypt({$ID_{CH_j}, KeyGen_{REQ}, n_{CHj}, H^{\wedge}{}_{CH_j}$}$K_M$) → C_4 Send(ID_S, { ID_{CH_j}, C_4}) → M_4 Update($M_{ID=4}$)
3. Server (BS)
POSS(BS) ={ ID_{BS}, $K_{BS–}CH_j$, $K_{BS–}CH_n$ } BEL(BS) ={#(ID_{BS}),#($K_{BS–}CH_j$),#($K_{BS–}CH_n$)} BL(BS) = Receive(BS, { ID_{CH_j}, C_2}) and Split({ ID_{CH_j}, C_2}) Decrypt({ C_2 }K_{BS-CHj}) to get [$ID_{CHj}, ID_{Ni}, ts_{CHj}, PCode_{REQ}, H_{CH_j}$] MsgFreshness($H'_{CH_j} - H_{CH_j}$) ≥ Δt if true, then Msg is aborted Hash({Concat($ID_{CHj}, ID_{Ni}, ts_{CHj}, PCode_{REQ}$)}) →$H^{\sim}$ Verify(H_{CH_j}, $H^{\sim}$) if mismatch, then abort Hash(h(.); Concat(ID_{BS}, K_M, P_{Code}')) → H_{BS} Encrypt({ $ID_{BS}, K_M, ts_{BS}, P_{Code}', H_{BS}$}$K_{BS-CHj}$) → C_3 Send(CH_j, { ID_{BS}, C_3}) → M_3, Update($M_{ID=3}$)

LT-SMM scheme is analyzed for inter-group key establishment scenario where mobile Node N_i authenticates with CH_j by sending joining request message. BS distributes the mobile nodes' information to CHs for new node joining in any region of the network. CH_j receives message from N_i and transmits to BS, the state of possession set is given as follows;

POSS(CH_i) = {ID_{CH_j}, $K_{Ni-}CH_j$, K_{CHj-BS}, ID_{N_i}, C_1, $Join_{REQ}$, ts_{Ni}, P_{Code}, $H_{N_i}ts_{Ni}$, H^*, P_{CH_j}, H_{CH_j}, C_2, M_2 }.

After the completion of the process, the forget operation identifies and removes the out of scope variables including ID_{Ni}, C_1, ID_{Ni}, $Join_{REQ}$, ts_{Ni}, P_{Code}, $H_{N_i}ts_{Ni}$, H^*, P_{CH_j}, H_{CH_j}, C_2, and M_2. New state of possession set is POSS(CH_i) ={ID_{CHj}, $K_{N_{i-CH_j}}$, K_{CHj-BS} } after removing the temporary values.

Similarly, the other entities maintain their sets. It follows secure communication mechanisms for secure key generation between BS − CH_j and BS−CH_n. After message transmission, the Update (M_{ID}) operation saves the identity of message in memory for future use when the other parties like receiver replies back. Finally, all the sets at participating entities are refreshed after the completion of inter-group key establishment between member nodes of CH_j.

TABLE 3. Simulation parameters.

Parameters	Values
Network Field	1500 x1500 meters
Node Count	100~500
Cluster radius	400 m
Sensing radius	100 m
Initial energy	1000 J
Transmission Power at Node	0.819 µJ
Receiving Power	0.049 µJ
Channel Type	Wireless
Propagation Model	Two Ray
Transmission Power at AN	0.5819 J
Receiving Power	0.049 J
Mac Protocol Type	Mac/802-11
Queue Type	DropTail/PriQue
Antenna Type	Omni Antenna
Max Packet in Queue	50
Router Trace	ON
Mac Trace	OFF
Agent Trace	ON

VI. RESULTS AND ANALYSIS

In this section simulation results are presented for energy consumption, resilience and rekeying operations. We have compared results with base schemes GKM [13], RiSeG [15] and LNT [20]. The proposed scheme is simulated using NS2.35 with C language on Fedora core-16. TCL is used for nodes deployment, configuration and messaging. H-Sensors are distinctly configured for an initial energy of 10000 Joules, message transmission and receiving a cost of 0.5809J and 0.049J respectively. Transmission radii of H-Sensor and L-Sensor are set at 400 meters and 100 meters, respectively. We deploy 10 clusters with varying cluster size from

10 to 100 in a region of 1500 × 1500 meters. A list of simulation parameters is shown in Table 3.

A. MESSAGE EXCHANGE ANALYSIS

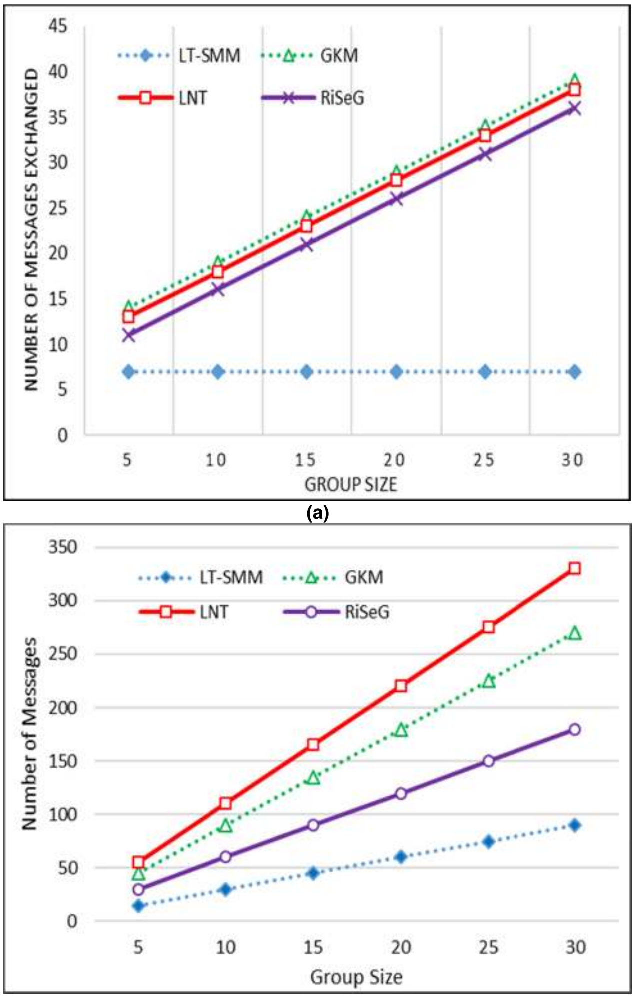
Message exchange procedure has significant importance in the communication of nodes either they are transferring data within or outside the group, joining or leaving the group and as well as forming new groups. In this procedure number of messages may vary through the protocol applied. Increasing and decreasing the quantity of messages has a direct impact over communication and computation cost.

1) GROUP DEPLOYMENT PHASE

In LT-SMM, an optimal approach is adapted that all nodes, cluster heads are deployed with a pre-loaded key installed by base station. That is why the scheme is optimal while deploying the groups in the sense exchanged messages that has a direct impact over communication cost and energy consumption. Only three messages are communicated between CH and each individual node. In this way, numbers of messages are directly proportional to group size. We analyzed number of messages for different phase as the quantity of communication messages exchanged from $\mathrm{N_i}$ to CH_j and Back from CH_j to $\mathrm{N_i}$. In the same fashion, it exchanges from CH_j to BS and back from BS to CH_j. First messages are counted for CH_j and $\mathrm{N_i}$ communication and the messages for CH_j to BS. Then this quantity is multiplied with the size of group as Nmsg = [(Tx + Rx)BS, CH, NxG] where, Nmsg shows the total number of messages exchanged, Tx denotes transmitted messages, Rx represents received messages, G represents group size. In this regard (Tx + Rx) BS, CH, N is the sum of transmitted and received messages between CHj and $\mathrm{N_i}$, same as the total number of messages exchanged between CH_j and BS. Figure 4(a) illustrates the messages exchanged during group deployment phase. The bottom line represents the smallest numbers of messages are exchanged with each group size in LT-SMM. Results show that for a group size of 25, the number of messages exchanged are 31, 33, 34 for RiSeG, LNT and GKM respectively. Our proposed LT-SMM dominates by sharing only 7 messages during group deployment due to involvement of CH. Number of exchanged messages during node joining phase are linear with increasing group size and on the other hand number of exchanged messages increasing rapidly in existing schemes.

2) NODE JOINING PHASE

We have observed that number of messages exchanged has great impact on communication, computation costs and energy consumption. During node joining phase, multiple messages are passed for authentication and/or key exchange purposes. In LT-SMM, total six messages are exchanged for each individual node to join the group between node, cluster head and base station. In other schemes, multiple messages are broadcasted within the group whenever a new node joins but in LT-SMM communication only occurs between node, cluster head and base station. There is no need to update the key to all nodes in the group for new node joining because cluster head is responsible for communication among nodes. Total number of messages Nmsg = [(Tx)BS,CH_j, N + G] include the sent and received messages over network. G denotes size of the cluster. For first group in LT-SMM, 2 messages are sent by node, cluster head sends 4 message in which 2 to node and 2 to base station. Moreover, one message is sent by BS. In this case, Nmsg = (2)N + (4)CH + (1)BS + 0 = 07. In above example, G has 0 value because no message is broadcasted in node joining process in LT-SMM. While calculating other schemes, each scheme broadcasts a key update message to all group members whenever a new node joins. It means that in LT-SMM, the number of messages remains constant for each group size either that is small or large. Figure 4(b) elucidates the total number of messages processed during node joining process i.e., 150, 225 and 275 messages for RiSeG, GKM and LNT respectively. LT-SMM dominates by exchanging 75 messages in node joining process, and other schemes have a little difference among them with different group sizes.



(a)

(b)

FIGURE 4. Number of messages exchanged in (a) Group deployment phase and (b) Node joining phase.

The main difference is broadcast message in other schemes which is not required in LT-SMM.

B. ENERGY UTILIZATION

Energy is the factor that makes these to live long. Whenever any type of communication takes place, then energy is consumed. Energy consumption is directly proportional to the messages exchanged during communication. There is variation while transmitting and receiving messages and as well as variation regarding type of node. The consumed energy is computed on each sent message that adds the consumed energy on transmission and energy consumed on reception, because each sent has an obvious reception on other end. Number of sent messages are calculated on each device and multiplied with the energy consumed earlier as given in (1).

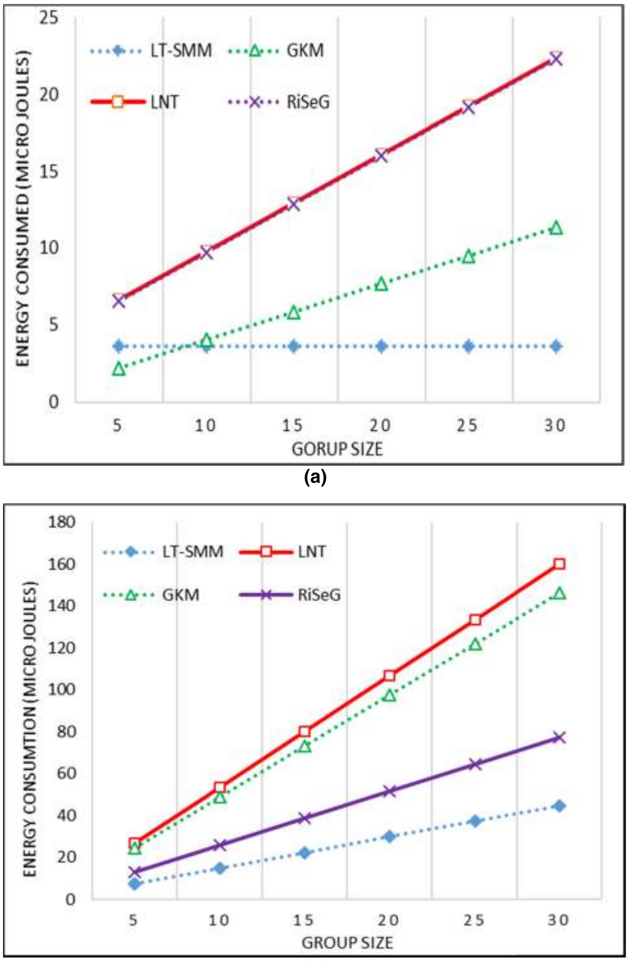
$$\mathrm{E}_T = [(\mathrm{M}_N \mathrm{x} \mathrm{E}_N)+(\mathrm{M}_g \mathrm{x} \mathrm{E}_g)+(\mathrm{M}_{\mathrm{CH}} \mathrm{x} \mathrm{E}_{\mathrm{CH}})+(\mathrm{M}_{\mathrm{BS}} \mathrm{x} \mathrm{E}_{\mathrm{BS}})]\mathrm{xG} \tag{1}$$

Here, E_T denotes total energy consumed, M shows total number of messages sent and E is to energy consumed on specified node. Moreover, G is representing group size. In the specifications, N is for simple node, g is for neighbor node or cluster head, CH and BS are for cluster head and base station respectively. Regarding values of these factors, in different phases of communication number of messages may or may not be 0 for some specific device including any of them. Total energy computation totally depends upon the number of messages sent.

1) ENERGY UTILIZATION FOR GROUP DEPLOYMENT

In Group deployment phase different types of nodes are involved in different schemes like simple nodes, neighbor nodes, cluster heads, neighbor cluster heads and base station. In LT-SMM, total 3 messages are exchanged between node and cluster head.

Next, 2 messages are exchanged from cluster head and 1 message from node. No messages are communicated for base station and neighbor node or neighbor cluster heads. Total energy calculation for single node in group deployment phase is calculated as EN, g, CH, BS = ETx + ERx i.e., 0.2309 = 0.1819 + 0.049 for simple node or neighbor node and 0.6309 = 0.5819 + 0.049 for cluster head or base station assumed. The energy for said group size is E = [EN + Eg + ECH + EBS] × G = [0.2309 + 0 + 1.2618 + 0] × 5 = 7.4635 μJ. Figure 5(a) elucidates energy consumption for group deployment and Figure 5(b) for node joining phase. We have compared LT-SMM with existing schemes. For a group size of 25, energy consumption is 64.63 μJoules, 121.95 μJoules and 133.49 μJoules for RiSeG, GKM and LNT respectively. LT-SMM dominates by consuming only 37.31 μJoules. LT-SMM is more efficient for the deployment of clusters because of CH based selection procedure and key verification for group members from base station.



(a)

(b)

FIGURE 5. Energy consumed in (a) Group deployment and (b) Node joining phase.

2) ENERGY UTILIZATION FOR NODE JOINING PHASE

In this research we also analyzed energy consumption in node joining procedure and make a comparison with some previous schemes such as GKM, LNT and RiSeG. As we have discussed earlier that energy consumption totally depends upon the number of messages sent and received through a node. On the contrary to deployment, the consumed energy is not multiplied with group size but added with the energy consumed over cluster head due to broadcast. And there is just the difference that LT-SMM has constant number of messages for different group sizes at new node joining while other schemes send a broadcast message to all group member to update key for new joining node, therefore, other schemes consume more energy than LT-SMM. By using eq., for example we calculate energy in LT-SMM for new node joining in the already deployed group as given in (2).

$$\mathrm{E}_{\mathrm{Total}} = [\mathrm{E}_{\mathrm{N}} + \mathrm{E}_{\mathrm{g}} + \mathrm{E}_{\mathrm{CH}} + \mathrm{E}_{\mathrm{BS}} + (\mathrm{E}_{\mathrm{CH}}\mathrm{xG})] \tag{2}$$

If no neighbor nodes are involved in joining process and no message is broadcasted and node transmit 2, cluster head 4 and base station 1 message as in LT-SMM, then total energy consumed at the time of joining is as follows; $E_{Total} = [(2 \times 0.2309) + 0 + (4 \times 0.6309) + (1 \times 0.6309) + 0] = 3.6163\mu j$. Figure 5(b) illustrates that LNT and RiSeG have almost same energy consumption as $19.2197\mu j$ and $19.1579\mu j$ whereas GKM consumes 9.52405 μj for a group size of 25. It has been observed for smaller group sizes of 5 nodes, GKM consumes 2.23505 μj which is lesser than LT-SMM's energy consumption. LT-SMM consumes 62% less energy as compared to GKM and 257% better than LNT and RiSeG. In case of LT-SMM, energy consumption line is horizontally straight at bottom along x-axis showing that there is no effect on energy consumption if the group size increases. It also represents that the scheme is consuming a tiny amount of energy that makes the node lives long in the group along with *CH*. Energy consumption for both group deployment phase and node joining phase to compute security measures is very economical as compared to existing schemes.

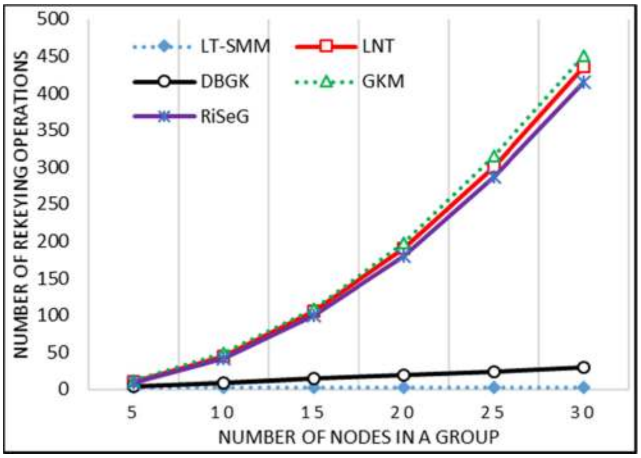
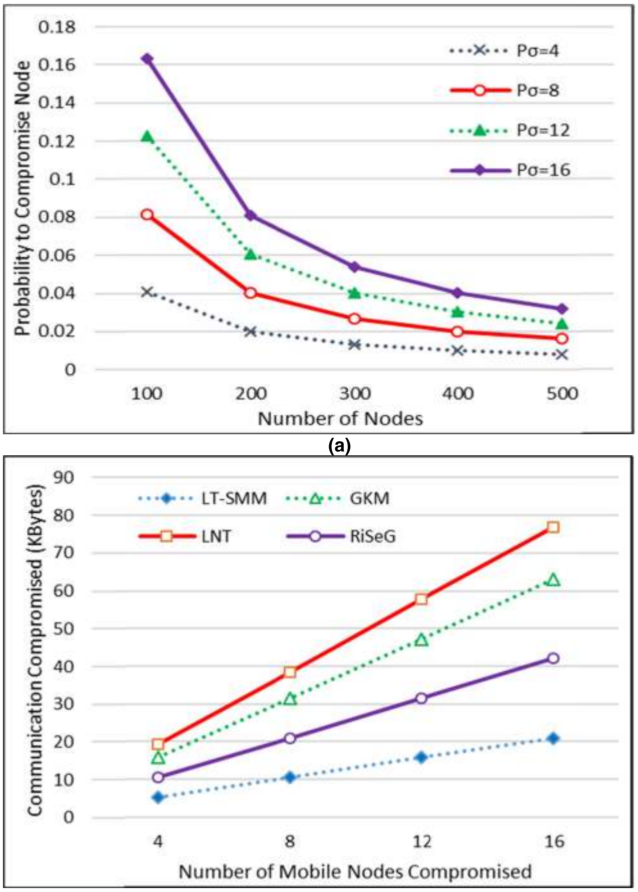


FIGURE 6. Rekeying cost of node joining/leaving.

C. REKEYING

Rekeying provide security but it also increases computation and communication cost. Unnecessary rekeying has a drastic effect on the life of sensor node. In tree based network models, if a network is represented as graph $G = (N, L)$ where N represents total number of nodes in the network and L represents established links. We can find maximum links as $\text{Maxlinks} = (|N| \times |N - 1|)/2$. For $N = 30$, the $\text{Maxlinks} = 435$ where 304 rekeying operations will be required when 70% links are active. It becomes worst when rekeying is performed multiple times in short duration. Figure 6 depicts that in LNT, GKM and RiSeG, more rekeying operations are required as compared to DBGK and proposed LT-SMM. Larger the group size then more rekeying operations are performed on each join or leave. Center line represents DBGK rekeying that is also equal to group size approximately. If we analyze deeply that in this rekeying is performed on all the nodes in the group except that corresponding node that is either joining or leaving the group. In this way we can say that total rekeying operations are Rekeying = Group Size – 1.



(a)

(b)

FIGURE 7. Probability to compromise the node is presented in (a), and the amount of data compromised in KBytes is shown in (b).

Bottom line the chart shows constant rekeying operation in LT-SMM for each group size. As we discussed above that total 3 rekeying operations are manipulated on each join of new node and there is no need for rekeying on leaving the node in this protocol. On leaving, just status is updated at cluster head.

D. RESILIENCE

During the group deployment, node's joining and migration phase, intruders can attack to breach the security and grab some portion of data. In this section, we have deduced the chances of compromised data in the network during setup phase when participants send an authentication request to respective CH. We have presented the probability for compromising the communication when a few nodes are compromised out of total participant nodes varying from 100 to 500. In this scenario, the probability P_σ as given in (3) predicts the chances that a participant is compromised. In this equation, N represents the total number of mobile nodes whereas σ represents number of nodes compromised. In this case, N-2 shows that sender and receivers are excluded from set of compromised mobile nodes. The term N-1 means to exclude the sender from total mobile nodes which is supposed to be uncompromised.

$$P_\sigma = 1 - \binom{N-2}{\sigma} \Big/ \binom{N-1}{\sigma} \qquad (3)$$

Figure 7(a) elucidates the probability to compromise a single neighboring mobile node when a number of other mobile nodes are compromised in the network. Results illustrate that for 300 mobile nodes, probability to compromise node is 0.01677, 0.03355, 0.0503 and 0.0671 for $\sigma = 4, 8, 12$, and 16 respectively. In case of compromising an intermediate device, the intruder can grab the data and security credentials stored in that device. By considering 25 mobile nodes per group with a probability $P_\sigma = 12$, the fraction of compromised for node joining scenario is illustrated in Figure 7(b).

VII. CONCLUSION

An efficient mobility management for logical tree oriented secure communication in group-based IoT enabled WSN is presented. Results of the protocol LT-SMM found efficient exactly according to the predicted results. There are many tree oriented schemes introduced but the problem is frequent rekeying on each node operation in the group. In this scheme majorly focused on to minimize extra rekeying. The scheme efficiently resolves the issue of un-necessary rekeying in tree based sensor networks. In this regard it provides secure group communication within the network. The scheme is also found to provide efficient secure communication for both inter-group and intra-group data exchange. This scheme is dedicated with the management of mobility of nodes in sensor networks. The protocol provides efficient backward and forward secrecy on leaving/joining of the node, and remains efficient if the operation is occurred frequently in the network. In future, the scheme can be observed with suitable modifications for secure routing in tree based sensor networks. This tree oriented schemes involve the networks where frequent mobility is involved. We have presented protocols for group deployment, node joining and migration. Schemes are simulated using NS 2.35 where TCL is used to configure and deploy the nodes, and message initiation. C language is used to write send, receive, encrypt, decrypt and hash functionalities. AWK files are used to extract results from trace files. Results shows that, LT-SMM reduces number of messages exchanged by 56% as compared to RiSeG during node joining phase. Moreover, 62% less energy is consumed as compared to GKM. Results are more dominating for other counterparts. In future, we shall analyze the performance of group based multicasting during sensing bottle neck and congestion scenarios.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[2] M. Chen, Y. Li, X. Luo, W. Wang, L. Wang, and W. Zhao, "A novel human activity recognition scheme for smart health using multilayer extreme learning machine," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1410–1418, Apr. 2019.

[3] A. Čolaković and M. Hadžialić, "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues," *Comput. Netw.*, vol. 144, pp. 17–39, Oct. 2018.

[4] I. U. Din, M. Guizani, S. Hassan, B.-S. Kim, M. K. Khan, M. Atiquzzaman, and S. H. Ahmed, "The Internet of Things: A review of enabled technologies and future challenges," *IEEE Access*, vol. 7, pp. 7606–7640, 2018.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[6] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How can heterogeneous Internet of Things build our future: A survey," *IEEE Commun. Surv. Tuts.*, vol. 20, no. 3, pp. 2011–2027, 3rd Quart., 2018.

[7] X. Luo, Y. Xu, W. Wang, M. Yuan, X. Ban, Y. Zhu, and W. Zhao, "Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy," *J. Franklin Inst.*, vol. 355, no. 4, pp. 1945–1966, Mar. 2018.

[8] S. Hameed, F. I. Khan, and B. Hameed, "Understanding security requirements and challenges in Internet of Things (IoT): A review," *J. Comput. Netw. Commun.*, vol. 2019, Jan. 2019, Art. no. 9629381.

[9] X. Luo, J. Sun, L. Wang, W. Wang, W. Zhao, J. Wu, J.-H. Wang, and Z. Zhang, "Short-term wind speed forecasting via stacked extreme learning machine with generalized correntropy," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4963–4971, Nov. 2018.

[10] X. Luo, C. Jiang, W. Wang, Y. Xu, J.-H. Wang, and W. Zhao, "User behavior prediction in social networks using weighted extreme learning machine with distribution optimization," *Future Gener. Comput. Syst.*, vol. 93, pp. 1023–1035, Apr. 2019.

[11] X. Li, F. Wu, M. K. Khan, L. Xu, J. Shen, and M. Jo, "A secure chaotic map-based remote authentication scheme for telecare medicine information systems," *Future Gener. Comput. Syst.*, vol. 84, pp. 149–159, Jul. 2018.

[12] P. Porambage, A. Braeken, C. Schmitt, A. Gurtov, M. Ylianttila, and B. Stiller, "Group key establishment for secure multicasting in IoT-enabled wireless sensor networks," in *Proc. IEEE 40th Conf. Local Comput. Netw. (LCN)*, Clearwater Beach, FL, USA, Oct. 2015, pp. 482–485.

[13] X. Luo, X. Yang, C. Jiang, and X. Ban, "Timeliness online regularized extreme learning machine," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 3, pp. 465–476, Mar. 2018.

[14] J.-H. Son, J.-S. Lee, and S.-W. Seo, "Topological key hierarchy for energy-efficient group key management in wireless sensor networks," *Wireless Pers. Commun.*, vol. 52, pp. 359–382, Jan. 2010.

[15] M. A. Kandi, H. Lakhlef, A. Bouabdallah, and Y. Challal, "An efficient multi-group key management protocol for Internet of Things," in *Proc. 26th Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, Split, Croatia, Sep. 2018, pp. 1–6.

[16] Y.-H. Kung and H.-C. Hsiao, "GroupIt: Lightweight group key management for dynamic IoT environments," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5155–5165, Dec. 2018.

[17] X. Luo, J. Deng, J. Liu, W. Wang, X. Ban, and J.-H. Wang, "A quantized kernel least mean square scheme with entropy-guided learning for intelligent data analysis," *China Commun.*, vol. 14, no. 7, pp. 1–10, Jul. 2017.

[18] G.-D. Zhou and T.-H. Yi, "Recent developments on wireless sensor networks technology for bridge health monitoring," *Math. Problems Eng.*, Oct. 2013, Art. no. 947867.

[19] J. Hernández-Serrano, J. Vera-del-Campo, J. Pegueroles, and C. Gañán, "Low-cost group rekeying for unattended wireless sensor networks," *Wireless Netw.*, vol. 19, no. 1, pp. 47–67, Jan. 2013.

[20] O. Cheikhrouhou, A. Koubâa, O. Gaddour, G. Dini, and M. Abid, "RiSeG: A logical ring based secure group communication protocol for wireless sensor networks," in *Proc. Int. Conf. Wireless Ubiquitous Syst.*, Oct. 2010, pp. 1–5.

[21] O. Cheikhrouhou, A. Koubâa, G. Dini, and M. Abid, "RiSeG: A ring based secure group communication protocol for resource-constrained wireless sensor networks," *Pers. Ubiquitous Comput.*, vol. 15, no. 8, pp. 783–797, Dec. 2011.

[22] N. Ferrari, T. Gebremichael, U. Jennehag, and M. Gidlund, "Lightweight group-key establishment protocol for IoT devices: Implementation and performance Analyses," in *Proc. 5th Int. Conf. Internet Things, Syst., Manage. Secur.*, Valencia, Spain, Oct. 2018, pp. 31–37.

[23] X. He, P. Szalachowski, Z. Kotulski, N. Fotiou, G. F. Marias, G. C. Polyzos, and H. Meer, "Energy-aware key management in mobile wireless sensor networks," *Ann. Univ. Mariae Curie-Sklodowska*, vol. 12, no. 4, pp. 83–96, Jan. 2012.

[24] M. Garcia, S. Sendra, J. Lloret, and A. Canovas, ''Saving energy and improving communications using cooperative group-based wireless sensor networks,'' *Telecommun. Syst.*, vol. 52, no. 4, pp. 2489–2502, Apr. 2013.

[25] H. Ning, Y. Fu, S. Hu, and H. Liu, ''Tree-code modeling and addressing for non-ID physical objects in the Internet of Things,'' *Telecommun. Syst.*, vol. 58, no. 3, pp. 195–204, Mar. 2015.

[26] O. Cheikhrouhou, A. Koubaa, G. Dini, H. Alzaid, and M. Abid, ''LNT: A logical neighbor tree secure group communication scheme for wireless sensor networks,'' *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1419–1444, Sep. 2012.

[27] L. Veltri, S. Cirani, S. Busanelli, and G. Ferrari, ''A novel batch-based group key management protocol applied to the Internet of Things,'' *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2724–2737, Nov. 2013.

[28] G. Dini and I. M. Savino, ''S2RP: A secure and scalable rekeying protocol for wireless sensor networks,'' in *Proc. IEEE Int. Conf. Mobile Adhoc Sensor Syst.*, Vancouver, BC, Canada, Oct. 2006, pp. 457–466.

[29] M. R. Abdmeziem, D. Tandjaoui, and I. Romdhani, ''A decentralized batch-based group key management protocol for mobile Internet of Things (DBGK),'' in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun., Dependable, Autonomic Secure Comput., Pervasive Intell. Comput.*, Liverpool, U.K., Oct. 2015, pp. 1109–1117.

[30] M. R. Abdmeziem, F. Charoy, ''Fault-tolerant and scalable key management protocol for IoT-based collaborative groups,'' in *Security and Privacy in Communication Networks* (Lecture Notes Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 239, X. Lin, A. Ghorbani, K. Ren, S. Zhu, and A. Zhang, Eds. Cham, Switzerland: Springer, 2018.

[31] A. D. Rubin and P. Honeyman, ''Nonmonotonic cryptographic protocols,'' in *Proc. Comput. Secur. Found. Workshop VII*, Franconia, NH, USA, Jun. 1994, pp. 100–116.

MUHAMMAD ARIF MUGHAL was born in Hyderabad, Sindh, Pakistan. He received the BCIT degree in computer science from the University of Sindh, Jamshoro, and the M.S. degree in computer sciences from the University of Science and Technology Beijing (USTB), China, where he is currently pursuing the Ph.D. degree in computer science. He is also with the Beijing Key Laboratory of Knowledge Engineering for Materials Science and Institute of Artificial Intelligence, USTB. His major research areas are wireless networks, key management techniques in wireless sensor network and security solutions for Internet of Things (IoT).

PENG SHI received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Science, in 2007. He is currently an Associate Professor with the University of Science and Technology Beijing, China. His research interests include social network analysis, knowledge engineering, WSN, and big data application in materials science.

ATA ULLAH received the B.S. and M.S. degrees in computer science (CS) from COMSATS Islamabad Pakistan, in 2005 and 2007 respectively, and the Ph.D. degree in CS from IIUI, Pakistan, in 2016. From 2007 to 2008, he was a Software Engineer with Streaming Networks, Islamabad, Pakistan. He joined NUML, Islamabad, in 2008, where he was an Assistant Professor and the Head of the Project Committee with Department of Computer Science, until 2017. From 2017 to 2018, he was a Research Fellow with the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. Since 2018, he has been with the National University of Modern Languages (NUML). He has supervised 112 projects at under graduate level and received One International and 45 National Level Software Competitions. He received ICT funding for the development of projects. He has published several papers in ISI indexed impact factor journals and international conferences. His research interests include WSN, the IoT, cyber physical social thinking (CPST) space, health-services, NGN, VoIP, and their security solutions. He is also a Guest Editor and a Reviewer for Journal and conference publications.

KHALID MAHMOOD received the M.S. degree in computer science from the Department of Computer Science and Software Engineering, International Islamic University, Islamabad (IIUI), in 2014. He is currently pursuing the Ph.D. degree with the University of Malaysia Pahang, Gambang, Malaysia. He holds a visiting position with the National University of Modern Languages (NUML). His current research interests include cryptographic security techniques, security of smart devices in the Internet of Things (IoT), and wireless sensor networks.

MUHAMMAD ABID received the Ph.D. degree in computer architecture from Tsinghua University, China, in 2012. He is currently a Principal Scientist with the Pakistan Institute of Engineering and Applied Sciences (PIEAS). He is also the Principal Investigator of the Nvidia GPU Education Centre and also a Co-Principal Investigator of the Data Science Lab, PIEAS. His current research interests include the IoTs, smart systems, wireless sensors networks, and intelligent embedded systems. One of his papers received a best paper award.

XIONG LUO received the Ph.D. degree in computer applied technology from Central South University, Changsha, China, in 2004. He is currently a Professor with the School of Computer and Communication Engineering, University of Science and Technology Beijing (USTB), Beijing, China. He is also with the Beijing Key Laboratory of Knowledge Engineering for Materials Science and Institute of Artificial Intelligence, USTB. He has published extensively in his areas of interest in several journals, such as the IEEE Transactions on Industrial Informatics, the IEEE Transactions on Human-Machine Systems, the IEEE Access, *Future Generation Computer Systems*, and the *Journal of the Franklin Institute*. His current research interests include machine learning, cloud computing, and computational intelligence.