ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20170807075015
Original scientific paper

Polynomial Based Dynamic Key Management for Secure Cluster Communication in Wireless Mobile Sensor Network

Eid REHMAN, Muhammad SHER, Syed Husnain Abass NAQVI, Anwar GHANI

Abstract: For inter and intra cluster communication, member nodes jointly build a mutual session key called cluster key to allow secure communication. Most existing schemes for cluster key management use messages exchange among the member nodes within a cluster for the new cluster key establishment when a node leaves or joins a cluster. This causes significant communication and computation costs. Furthermore, the secure distribution of cluster keys among member nodes in frequently changing environments is a difficult task without encryption and decryption operations. For secure cluster key management, we utilized polynomial (*P*) to accomplish effective intra-cluster key management and produced polynomial for making an inter-cluster key distribution. The main contribution is to generate polynomials and broadcast to nodes whenever a change occurs in a network or demanding nodes for secure key management. The presented scheme supports scalability for an increasing number of nodes using polynomials. The proposed scheme increases the lifetime of the network by decreasing the key pool size.

Keywords: cluster security; key management; polynomial; wireless mobile sensor network

1 INTRODUCTION

Because of random harsh deployment and wireless communication, the researcher has concentrated on the security of WMSN. In light of the resource constraint condition of the WMSNs, usual security systems are not viable because they devour energy excessively. Subsequently, scientists are presenting novel less expensive security techniques for each conceivable security part of WMSNs. WMSNs comprise some little, low-cost, movable, self-representing closes sensor nodes having little capacity to process information [1] and with constraint processing, memory, and energy. MICA2 Motes sensor node utilizes 8-bit instruction in16 MHz processors using 4 KB of EEPROM and 128 KB of reprogrammable memory [2]. For better network management and resource utilization, WMSN is organized into clusters [3]. Many researchers give an argument that cluster mobile sensor network can reasonably outperform network effectiveness, operational implementation, and increase the network lifetime. Conventional security systems using public key cryptography causes overhead with respect to computational and transmission costs. Key management is required for security purposes but is very difficult in such a resource constraint network of WMSN [4].

In conventional networks, end-to-end security is conceivable in light of the fact that it is a bit much for the center node to have entry to the substance of data packets. Nevertheless, in WMSNs, intermediate node specifically gets to the substance of messages; along these lines, it is harder to deploy basic end-to-end security. The intermediate nodes can without much of a stretch read or, then again adjust the information of a source towards the target node. The operation of a system can be effortlessly disturbed by modifying the packets through the intermediate nodes effectively. Besides, any malicious intruder node can effectively peruse confidential packets, if there is no scheme for securing packets. Intruders can inject false data or resend old put away information from real node to work the data aggregation handle, coming about a man-in-the-middle attack.

Different nameless security schemes have been proposed for WMSN in later past which shows differing natures and additional levels of security assurance at different prices. In this section, we talk about the current cryptography schemes that tended to the issue of key management for secure communication in WMSNs. Because of the resources limitation in WMSNs, a comprehensive harmony between energy usage overhead and the security level is expected to moderate the safety threats. Some symmetric metrics, for example, Node-ID, message authentication code (MCA), nonce-number, and time-stamps. These are energy efficient parameters for cluster key management techniques. Additionally, this keeps away from the distinctive kind of attacks from a suspected node and stay away from compromised node attacks. Various security schemes presented for WMSNs utilize symmetric encryption, because of the simplicity of its execution [5]. Other than this, single node authentication has turned out to be not able to take care of the increasing transmission demand. When the services request is growing up day by day, a multiparty calculation is fundamental for nodes verification (authentication) concurrently and safely. Similarly for inter and intra cluster communication, member nodes jointly build a mutual session key called cluster key to allow secure exchange of messages [6].

The whole cluster member authentication at once is extremely useful because it can authenticate all nodes of cluster simultaneously. This can be utilized for node authentication to distinguish between trustable supporters and non-supporters as well. For secure communication within the same cluster and among various clusters, we utilized a polynomial (*P*) to accomplish effective intra-cluster key management and produced a polynomial for making an inter-cluster key distribution. Thus, our presented scheme can be exceedingly successful in every authentication and key formation since it might give the best broadcast communication. In addition, for computation, each node requires a polynomial assessment and key-hash highlight. This is less demanding for performing encryption and decryption. The main objective is to develop a secure and efficient scheme for key distribution, calculation, and renewal Prior to the nodes deployment in WMSNs, every node is pre-designed with an arrangement of symmetric keys shared to the various

nodes of the cluster to transfer its Id safely to the CH. When the nodes are deployed, each node specified the predetermined symmetric keys used for the communication with the CH. The CH keeps up all the symmetric keys shared to the sensor nodes having a place with its cluster. The fundamental goal for utilizing these keys is to encourage the multi-hop correspondence while communicating secret information mainly, the individual proportion dissemination and the exchange fact between the CH and BS. The execution of the polynomial that is connected for determining an intra-cluster key can decrease the cluster (session) key storage overhead at the member nodes and their CH. When the intra-cluster key is procured, the CH self-creates a polynomial function, which is vital for improving an inter-cluster key organization. This encourages the decrease in the communication cost of the CH. In the cluster key creation for intra-clusters, our proposed method reduces the amount of communicated messages as a whole in the inter-cluster communication. The proposed scheme enhances the WMSN security and life-time by lessening the quantity of exchange messages during node mobility while diminishing the scope range of CH and the requirement for a node to move and join the intense scope region of other CH. Our work also presents a safe node migration in which dependable handoff happens and new connections are built up amongst CHs and member nodes. The main contribution of the proposed scheme is to generate polynomials whenever a change occurs in a network or demanding nodes. Polynomials are generated dynamically when change occurs in a cluster to create a new cluster key (session key). Therefore, nodes must hold only a few pre-assign data. Additionally, the presented scheme supports scalability for an increasing number of nodes because the polynomials used for computation of cluster keys are dynamically generated after the deployment. Our scheme frequently refreshes the cluster keys (session key) because of the easy new polynomial generation. The proposed scheme increases the life-time of WMSN by decreasing the key pool size.

The rest of the paper is organized as follows. Section 2 provides the literature review of some well-known cluster key management algorithms for WMSN. Section 3 describes the network and threat model. Section 3.2 describes the System model. Section 4 presents a proposed polynomial based cluster key management scheme. Section 5 describes security analysis of the proposed scheme with other scheme and section 6 discusses the simulation performance of the proposed scheme.

2 LITERATURE REVIEW

After stabling the clustering and CH selection, it is necessary to established secure communication between the member node and CH of a cluster for data collection. A lot of work has been done in static WSN, but in WMSN still faces research achievement in establishing secure communication in frequently changing topology.

By and large, security in WSN has been broadly researched in recent times [7, 8]. A large portion of the security arrangements has been composed either to protect WSNs from some known attacks (e.g., particular sending, dark gap) or cautious procedures: for example, intrusion detection system [9] is proposed.

Prevention mechanism such as key management scheme is presented [10]. The key messages transmitted through an intermediate node ought to likewise be secure [11]. In any case, they are designed for static WSN [12] which requires a vast number of messages to set up and maintain update key over the system. In addition, the dynamic nature of WMSN (frequent mobility) requires keys to be refreshed when needed. This causes immense communication overhead on nodes with less energy and henceforth decreases their lifetime.

To build up secure keys among the member node of a cluster, the scheme in [13] proposed a Logical Key Hierarchy (LKH) where the whole cluster is represented like a tree. Leaf nodes i.e. member nodes share symmetric keys. The cluster key is allocated to CH. Jen - Chiun Lin et al. presented One-way Key derivation (OKD) [14] that used the idea of one-way hash function like Dini et. al. LKH scheme was additionally enhanced by Je et. al. to consider the resources of each node during tree development. In these schemes, indirect path keys between leaf sensor nodes over the cluster are set up using the CH node of a neighbouring cluster. Similarly, another tree-based cluster key management is presented [15] in which a leaf node can calculate keys toward CH.

One key exchange scheme is Localized Encryption and Authentication Protocol (LEAP) [16], which was proposed to secure the inter-cluster communication of WSNs. LEAP organized communication messages and presented four sorts of keys inside a network for security. Every one of the four keys was shared between individual nodes of the WSN. This scheme is very costly because large keys are used.

The paper [17] utilized two polynomial pools, common mobile and common static, on which they executed three-level architecture to pick up an improved level of security for WSNs. The pools have a sensor node with getting to focus and movable sinks. Keys are conveyed by the access point and the portable sinks. Pairwise key pre-distribution techniques are utilized for authentication of a node with the assistance of polynomial keys.

One of the key management schemes [18] is presented for heterogeneous WMSNs using asymmetric key pre-distribution and hash function. It utilizes a seed key and hash capacity to understand the authentication of a mobile CH, however, it just allows CH mobility, and the entire members are static.

A key pre-distribution algorithm where BS provides seeds to sensor nodes to compute another key, which gives satisfactory security, was depicted in [19]. It permits secretly appropriating a secret to an arrangement of beneficiaries with just a single multicast correspondence [20]. A less expensive XOR-based re-keying scheme is presented in [21], which does not need message exchange in WSN for key distribution.

A dynamic polynomial based key management scheme is presented in [22] where master node is used for secure communication during cluster key establishment. In this scheme, some advance nodes are used called H- sensor nodes responsible for key management. Every time H-Sensor nodes generate polynomial when change occurs in a cluster. It enhances the left time of the sensor network by

reducing the key pool size but this needs advance node which increases the cost of a network.

In [23] the author presents an energy efficient distributed deterministic key management algorithm (EDDK) for WMSN. EDDK concentrates on the establishment and updating of the pairwise keys, also the inter cluster keys and can settle a few imperfections in some current key management schemes. Construction of neighboring table during key establishment not only gives the security to key support and information exchange, but it can likewise be utilized to adequately deal with the storage and refresh of the keys. By utilizing the elliptic curve digital signature algorithm in EDDK, both new and movable sensor nodes can join or leave or rejoin a sensor network safely. The real reason for the low performance of EDDK is that it calculates the pairwise keys and changes in neighborhood impact on the estimation of pairwise keys, which may give the wrong example of pairwise keys and needs the recalculation of pairwise keys.

In [24] a new scheme called cluster based mobile key management system (CMKMS) considers two stages, first stage for key maintenance which sets up two private keys, home key for its own cluster and foreign key when a node moves starting with one cluster then onto the next. The second stage keeps up the keys when CH moves starting with one group then onto the next. The proposed scheme enhances the efficiency of key management as far as security, energy saving, mobility, and network scalability. This scheme has efficiency because of using the RC4 algorithm for encryption and decryption.

Naqvi presents a novel key management scheme [25] to enhance the energy efficiency, security and scalability prerequisites by diminishing the computational complexity of the scheme. This scheme keeps running in two stages; in the first stage, it sets up the cluster and appoints the home and foreign keys to every node. The second stage keeps up the key update during the node and CH mobility. Besides, to improve energy efficiency, reduce computational overhead and to enhance encryption speed, the ECDSA encryption algorithm has been used.

Similarly, many schemes [26-30] have been presented for dynamic key generation in cluster WMSN for a heterogeneous network, where advance nodes are used for key generation and maintenance.

Most of the existing schemes for cluster key creation require messages exchange between member nodes within a cluster for the new cluster key establishment when a node leaves or joins a cluster. This causes significant communication costs.

3 THREAT AND SYSTEM MODELS

In the threat model, initially, the nodes are not compromised at the time when networks are deployed. After the nodes deployment, attackers can launch attacks.

Adversaries try to get important data of nodes or network by executing some advanced functions. The adversary can be in the form of either active or passive attacks. Physical attacks can be launched by an adversary to compromising node and get secret data for future use. Inside and outside attacks can distribute the vital thing, caught data, to interrupt with the security collusively, and are called collusive physical attacks. For example, the repudiated node and the newly joined nodes can dispatch tricky attacks over the cluster key no longer having a place with them. Capturing node physical attacks is exceptionally destructive to the system if over the top cryptographic keys still available inside a node. By and large, attacks influencing safe key distribution are eavesdropping and node capturing [31]. The presented key management scheme is safe from nodes capturing and eavesdropping attacks by using effective key distribution using the dynamic generation of a polynomial. CH sends an expanded polynomial without any encryption and decryption for session key to its member nodes. It is very difficult to guess the intra-cluster key (session key) key management for polynomial because polynomial factorization is NP - hard [32].

3.1 System Model

Our network consists of two types of nodes as shown in Fig. 1. One type is a normal node which forms cluster using some well known algorithm [33]. The second type of node are these which are selected as a cluster head (CH) using an algorithm [34]. These nodes have limited computation, communication, and storage capacity. Here we have assumed that CH cannot be compromised. The second type of node is BS having a lot of resources and powerful computation. To establish mutual authentication among the sensor nodes (sensors, cluster heads), each sensor needs to accomplish a shared authentication mechanism and create a dynamic shared cluster prior to the communication. At last, every node can confirm their authenticity using this shared with another. During adversary observation, the member nodes of the cluster are easily compromised. So a key refreshing mechanism can play a crucial capacity to shield from several forthcoming deceptive moves of the compromised node. In addition, new node addition in network enhances the scalability of the system.

4 PROPOSED POLYNOMIAL BASED CLUSTER KEY MANAGEMENTSCHEME

This paper presents an efficient key management scheme for cluster WMSN. First each sensor node including CH, member nodes are preload with unique ID, network secret key Nk, hash function (h), encryption and decryption function along with unique ID by the BS.

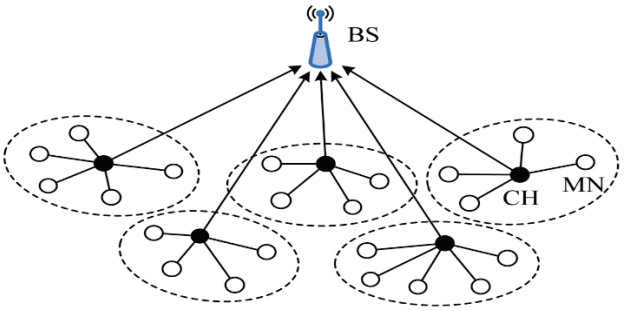

Figure 1 Cluster besed Network Model of WMSN

CH has also preload with same and additionally with polynomial calculation function. Here we have assumed that the CH and BS formed secure authentication

mechanism including ECC, which is very difficult to inverse. After clustering and CH selection, every member node sends its join request message ENk (ID_i||MAC(ID_i)) having ID of CH member nodes and MAC of their ID. We assumed that the maximum hop between CH and member nodes is one as shown in Fig. 4. After receiving encrypted message for every member node, CH decrypts the messages, takes ID and computes hash value using hash function for every member node to generated polynomial using Eq. (2). List of Notation used for the proposed scheme is given in Tab. 1.

$$P(x) = \mathrm{e}^{\log\left(x-h(ID_1)\right)\left(x-h(ID_2),\ldots\left(x-h(ID_\alpha)\right)\right)+Ck} \quad (1)$$

It is expanded as:

$$P(x) = \mathrm{e}^{\log\left(x^\alpha - x^{\alpha-1} + x^{\alpha-2} - x^{\alpha-\varepsilon},\ldots,\pm Z\right)} \quad (2)$$

where α is the size of cluster (member nodes in cluster), and x is predefined significant value and hash value of every sensor nodes identities (ID) of all cluster member is calculated. Ck is the cluster key generated by base station and sent to CH. After computing hash values of all member nodes ID, CH makes a polynomial using Eq. (1) and then expands this polynomial to Eq. (2). The encrypted expanded polynomial with NK, ENK (P(x)||MAC(P(x))) is distributed to the member nodes of cluster.

After receiving this, each cluster member node decrypts it and substitutes its ID value in the polynomial to derive the cluster key (Ck).

$$\rightarrow \mathrm{e}^{\log\left(x^\alpha - x^{\alpha-1} + x^{\alpha-2} - x^{\alpha-\varepsilon},\ldots,\pm Z\right)}$$
$$\rightarrow x^\alpha - x^{\alpha-1} + x^{\alpha-2} - x^{\alpha-\varepsilon},\ldots,\pm z$$
$$\rightarrow \text{Where: } \mathrm{e}^{\log(x)} = x$$
$$\rightarrow Ck \text{ Where: } x = ID_i$$

Table 1 List of Notations

Notations	Description
$P(x)$	Polynomial
H	Hash function
ID	Id of sensor node
NK	Network secret key
ENk	Encrypted with network secret key
MAC	Message authentication code
Ck	Cluster key (session key)
A	Size of cluster (number of nodes)
MAC_AddId	MAC of list of all ID
CHId	Cluster head ID
Nonce	Nounce
KCb	Key between CH and base station

When some node joins or leaves the cluster, the CH generates a new polynomial $P(x)$' for the rest of member nodes of cluster to provide new Ck. Similarly, when new CH is selected in cluster it will repeat the same above process to generate new polynomial.

4.1 Addition of Node

In the proposed scheme, the entire cluster is not involved in rekeying process. Because of mobility in WMSN the node may be moved from one cluster to another or new node can be added to the network. A scalable key management scheme needs the capacity of adding new node to the network. These new nodes require to build up shared cluster key (Ck) with existing nodes for authentication. When a new node tries to join cluster, it sends join request message containing ID to the correspondence CH. The BS plays an important role in authentication process. CH checks it in its look up table to confirm whether the node is a new one or already existing node has moved. When BS knows that this node is a new node, then it performs a new node addition process using algorithm 1.1 in Fig. 2. If the node already exists in the network, but has moved from one cluster to another cluster then a node migration algorithm 1.2 shown in Fig. 3 is performed.

4.1.1 New Node Addition

New node addition may occur in a cluster. When node is added to a network, it must become part of one cluster. This node sends joint request to CH. Node is already preloaded with Nk, hash function and unique ID, ENk(ID||MAC(ID)). CH forwards a request to the base station by adding CHId encrypted with KCB. Base station checks the status of this node using some intrusions detection algorithm, which is not addressed in this work. After confirming its validity, BS broadcasts encrypted message containing LIdi (list of members ID), MAC_AddIdi and nonceN to all CH's as shown in algorithm 1.2. The CH multicast the cluster key (Ck), ID list of sensor nodes and MAC in encrypted format.

This guarantees the freshness and reliability of the node joining procedure. Fig. 2 shows a new node addition, where CH gets authentication for BS. Base station maintains a list of abnormal nodes, checks legitimacy of every node using some intrusion detection mechanism, which is out of scope.

4.1.2 Migration of Node

Because of mobility in node, it is possible that some node moves from one cluster to another cluster in WMSN. CH node maintains a list of those nodes rejected at joining because they did not fulfil the security requirement as needed per BS. This protects from intruders to become part of the network during node migration from outside the networks.

Before the node migration, it sends leave message to the existing CH_i. CH_i sends this migration information with valid ID of sensor node encrypted with KCb that wants to migrate. CH_i removes this node from the cluster by generating new polynomial $P(x)$' and distributes it to its member nodes. At the same time this node sends join request message along with preload unique ID and Nk to another CH_j to join their cluster. This CH_j checks this node in their maintaining black list of rejected nodes, if it exists then rejects their request. Otherwise, CH_j sends verification request to base station for confirming verification. The base station also checks their legitimacy by comparing this node ID in the whole black list nodes. If it does not exist, then BS computes authentication of encrypted message with KCb and sends it to CH_j. Before the addition of this

migrated node in cluster, CH_j confirms the message credibility parameters value and keys. After the confirmation of verification process, CH_j sends a join success message to this node and informs BS through sending an acknowledgement. Fig. 2 shows the node migration algorithm for existing node of network. After allowing joining to this node CH_j generates new polynomial to new cluster key (session key).

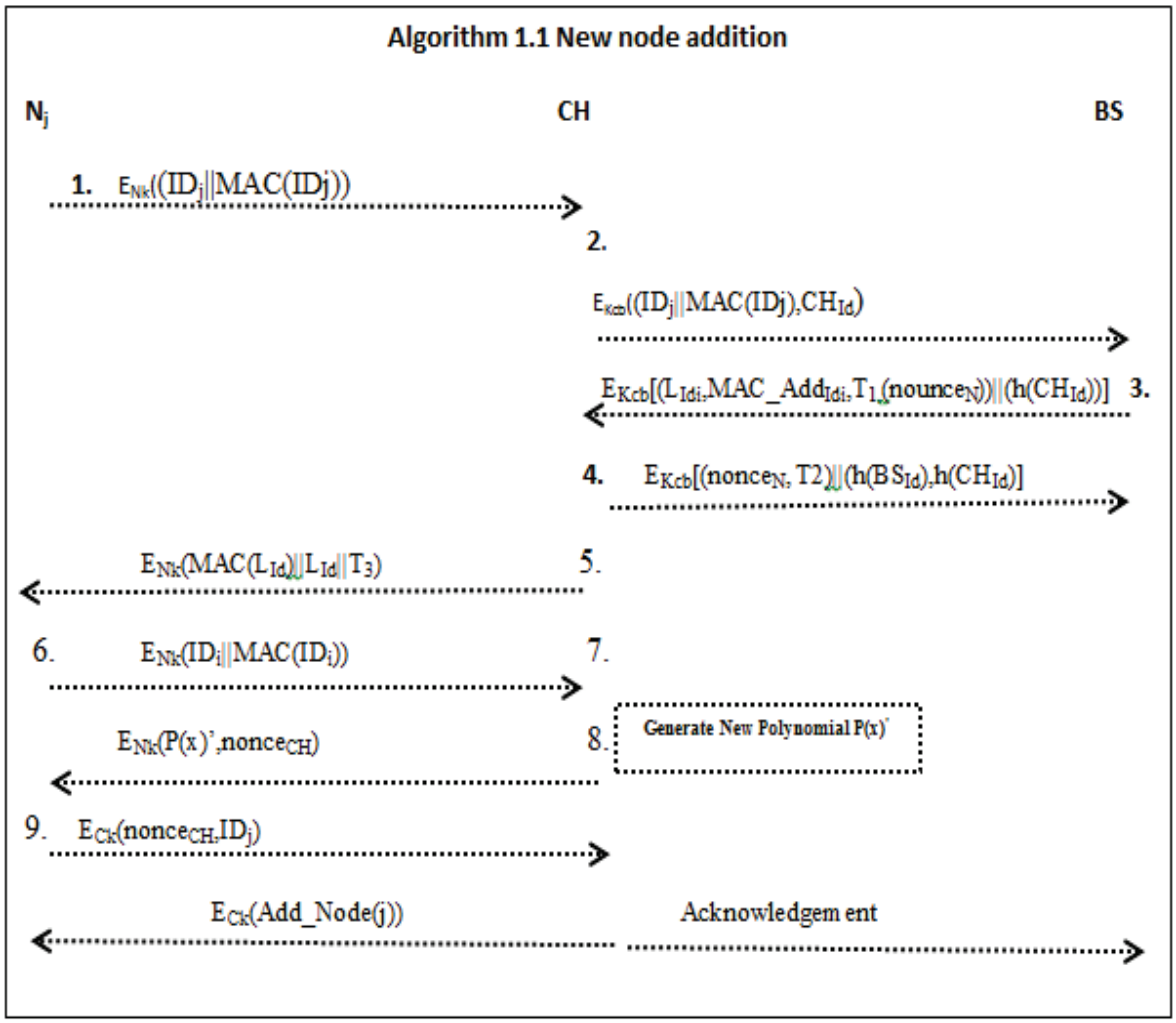

Figure 2 Node Addition Algorithm

5 SECURITY ANALYSIS

This section analyses the security features of the proposed scheme. The threats and attack models try to affect the key management in cluster communication of two types, one is inside attacks and second is outside attacks. The cluster key is established through distribution of polynomial, which provides secure establishment of key between member nodes. Sharing of cluster key through polynomial between member node of restrict attacker to know about cluster key. Without knowing the cluster key, attacker cannot intercept member communication individual and be able to modify it. The proposed scheme ensures integrity and confidentiality through resistance against insider attacks.

The non-member nodes or old member of cluster can eavesdrop the cluster information though outsiders attack. When the CH sends the expanded polynomial without encryption then outsider attacker attempts to eavesdrop that communication. Without knowing the cluster key *Ck* and even with no encryption, it is very hard to recover the cluster key. On the off chance that the adversary tries experimentation approach, the exponential log implements extra complexity. This approach may occupy the attacker to discover the log value for the separate x value. Deriving the cluster key using polynomial factorization is additionally exceptionally hard. To guess the cluster key, an extended polynomial needs really O ($n$log$n$) solutions for a polynomial extension problem. Here, the presence of cluster key in the polynomial forces difficult in the polynomial factorization and making the polynomial factorization with a specific end goal to break the proposed scheme is non-deterministic polynomial-time (NP) hard.

Whenever a member node leaves the cluster, CH delete their Id and generate new polynomial containing new cluster key *Ck* and send to the remaining member nodes. Hence, the cluster member nodes key is refreshed by receiving new one. Thus, departed member is prohibited to use their previous key for communication in cluster.

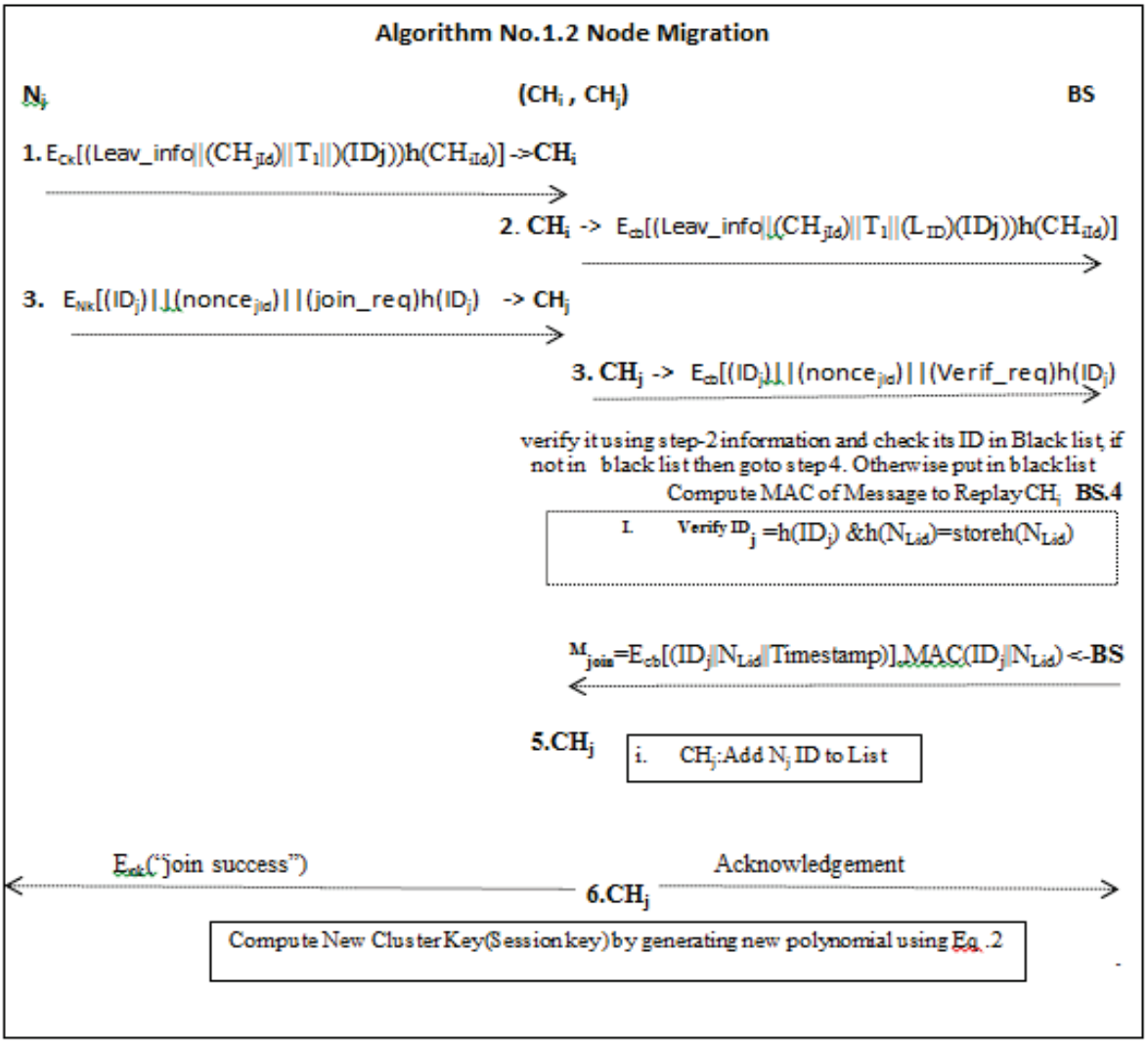

Figure 3 Node migration algorithm for cluster key management

6 RESULT AND ANALYSIS

The proposed solution has been validated through simulation and comparing its performance with the EKMS [23], EDDK [24] and scheme [25]. The result comparison among the proposed scheme and the rest of schemes has been carried out using the following simulation parameters shown in Tab. 2. The proposed system efficiency is analysed based on the cost effectiveness by using these parameters: communication overhead, computational overheads, storage overhead, energy consumption and average latency.

Table 2 Simulation Parameters

Parameters	Values
Mobility model	Random way point
Number of sensor nodes	100
Length of Data packet	256 Bit
Length of control packet	50 Bit
Initial Energy	1 Joule
Interface Queue type	Proposed scheme
Communication model	Bi-directional
Simulation Area	1000 × 1000 m^2
Node Speed	1 - 30 m/second
Maximum Queue	50 packets
Simulation Time	300 Second

Communication overhead is the measurement of number of bits transmission for the establishment of cluster key in case of new node addition or existing nodes migration between CH and member nodes of cluster. Fig. 4 shows the communication overhead for establishing new cluster key between all participating nodes in case of number of new nodes addition to cluster. The main reason of lower performance of EDDK is the usage of pairwise key and local cluster key for each node in cluster. In case of new node addition first pairwise keys are established, then cluster key will be established which increases communication cost. Similarly, EKMS and scheme [25] used local cluster key and foreign key for the establishment of new cluster key, which also increases communication overhead. Additionally when the number of new nodes increases, the communication overhead also increases in CH. Our proposed scheme has low communication overhead because of only one message exchange between new nodes and CH for addition, while the rest of messages for addition are exchanged between BS and CH. There is no need to exchange any messages except new polynomial after BS authentication between other member nodes of cluster. Thus, the communication overhead between the CH and member remains constant i.e. only polynomial is exchanged between CH and member node for new cluster key generation. Hence, the proposed scheme has superior

performance in communication cost compared with the other three schemes for secure new node addition in cluster.

Similarly, communication overhead in case of existing nodes migration from one cluster to another cluster is shown in Fig. 5. Every node sends leave and join request to their alternative CH for leaving and joining new one. CH should update their cluster key for their member as authentication process is completed by BS. In EDDK, node has exchanged two keys for leaving and two for joining which increases communication overhead. Similarly EKMS and scheme has higher communication cost when existing node moves from one cluster to another cluster. Our proposed scheme performed better then because only one leave request and one joint request are generated for existing node. CH generated new polynomial after confirming their authentication from BS. Therefore, member nodes do not need to communicate each other for establishing a new secure cluster key for their secure communication in cluster. At last, the communication overhead between the CH and each member node is constant because polynomials are exchanged when new cluster key is established.

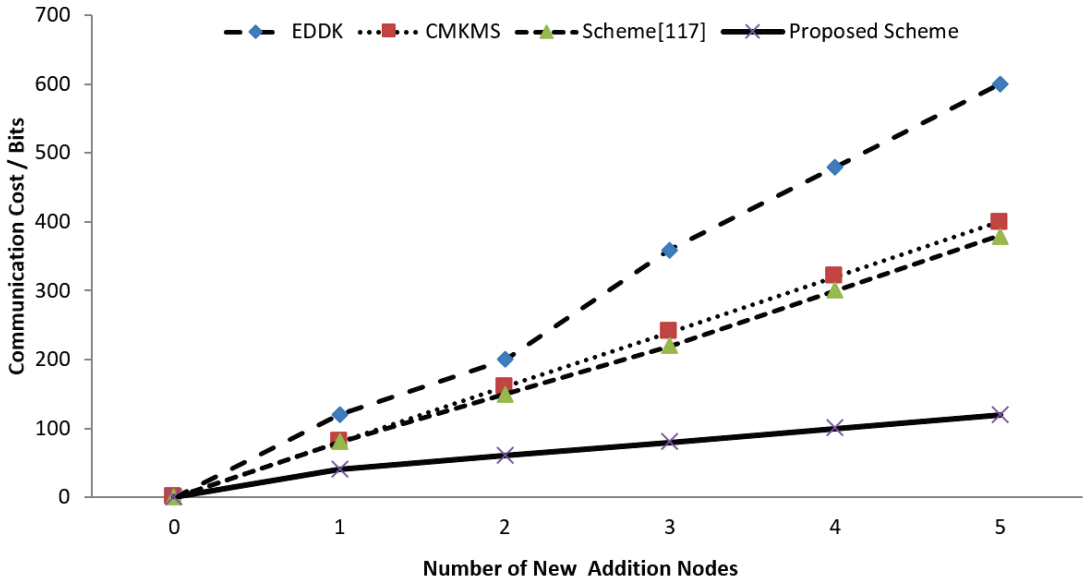

Figure 4 Communication cost in case of new node addition

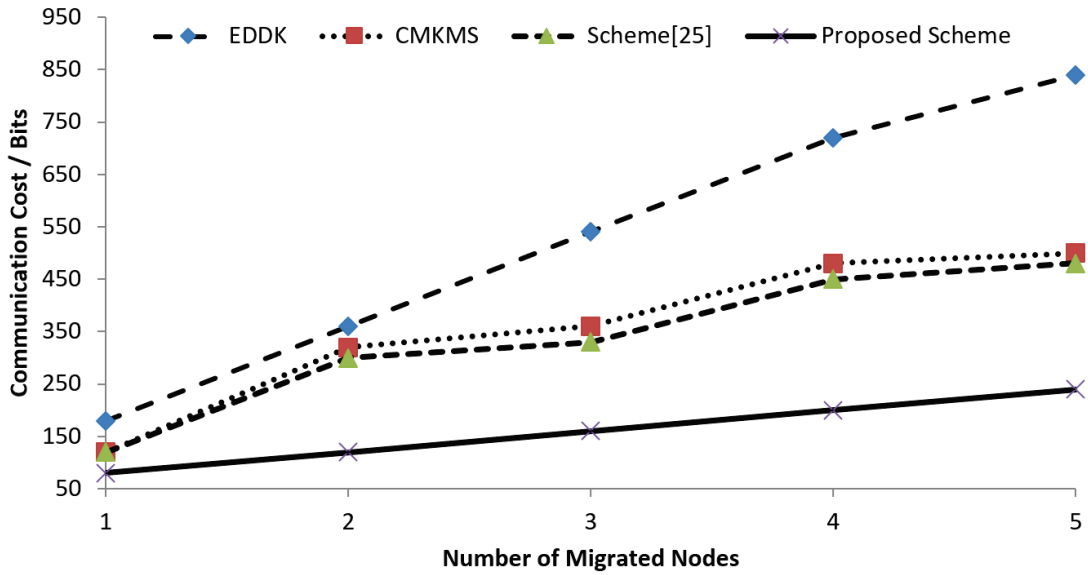

Figure 5 Communication Overhead in case of Node Migration

Tab. 3 shows the Storage overhead of the proposed scheme, EDDK, EDMS and scheme [25]. The amount of storage capacity required to store security parameters for cluster key generation considered is storage overhead. EDDK has the worst storage due to the fact that each sensor node requires for storing a neighbor table because each node has to generate new common key for the new nodes addition. Additionally CH need more storage for storing all member nodes keys and tables. EKMS and scheme has also high storage overhead because of usage of two keys i.e. one is home key and the other is foreign key for the generation of cluster key. The proposed scheme has less storage overhead because CH store only one polynomial and n ID of member nodes of cluster. At the other side, member nodes store only one key along with one polynomial.

Table 3 Storage overhead Comparison

Name of Scheme	No. of Key stored in CH	No. of key stored Member nodes
EDDK	$n + n \times$ Neighbor table	$2 + 3 \times$ Neighbor table
CMKMS	$2 \times n$ + polynomial	2 keys + polynomial
Scheme [25]	$2 \times n$ + polynomial	2 keys + polynomial
Proposed Scheme	$n + 1$ + polynomial	1 key + Polynomial

Table 4 Computational cost Comparison

Name of Schemes	Encryp / Decryp in CH	Encyp / Decryp in member node
EDDK	Encryp + Decryp + 1 Pseudo random function	Encryp + Decryp
CMKMS	Encyp + Decryp	Encypt + Decryp
Scheme [25]	Encyp + Decryp	Encyp + Decryp
Proposed scheme	0	0

Tab. 4 and Fig. 6 show the computation overhead comparison of proposed scheme with the rest of three schemes for establishing a cluster key. Encryption and decryption plus one pseudo random generation function execution operations are required in order to established secure cluster key in EDDK. Similarly, in EKMS and scheme [25] also need an encryption and decryption operations for establishing cluster key for entire cluster. In the proposed scheme no encryption and decryption operation are required for establishing cluster key. For any number of nodes, CH sends an expanded polynomial without encryption. From CH point of view, the computational cost of computing polynomial is less than an encryption and decryption. The number of operations and time required to compute polynomial by the cluster in linear. Thus, the proposed scheme performed better performance in light of computation cost.

The amount of energy consumed during cluster key establishment when member node leaves or joins a cluster in EDDK, EKMS and scheme [25] is compared with the proposed scheme. EDDK has lower performance in terms of energy consumption because the change in neighboring nodes effects the calculation of key, which may give wrong instance of key and re-calculation is needed. The main reason of EKMS having lower energy efficiency is the usage of RC5 algorithm for encryption and decryption involved during node leaving or joining of cluster. Similarly the scheme [25] also used ECDSA algorithm for encryption and decryption in case of leaving or joining of cluster. The total energy required to compute a polynomial of cluster is linear, while the rest of three schemes have n multiple and n shows the number of nodes in cluster. The energy consumption of computing a secure hash of sensor node id is 5,6 nJ per byte [32]. For cluster of n nodes, the hash computation of member node n consumes $n \times 5{,}6$ nJ. Fig. 5 shows the energy consumption of cluster key establishment of the proposed scheme compared with the rest of three schemes.

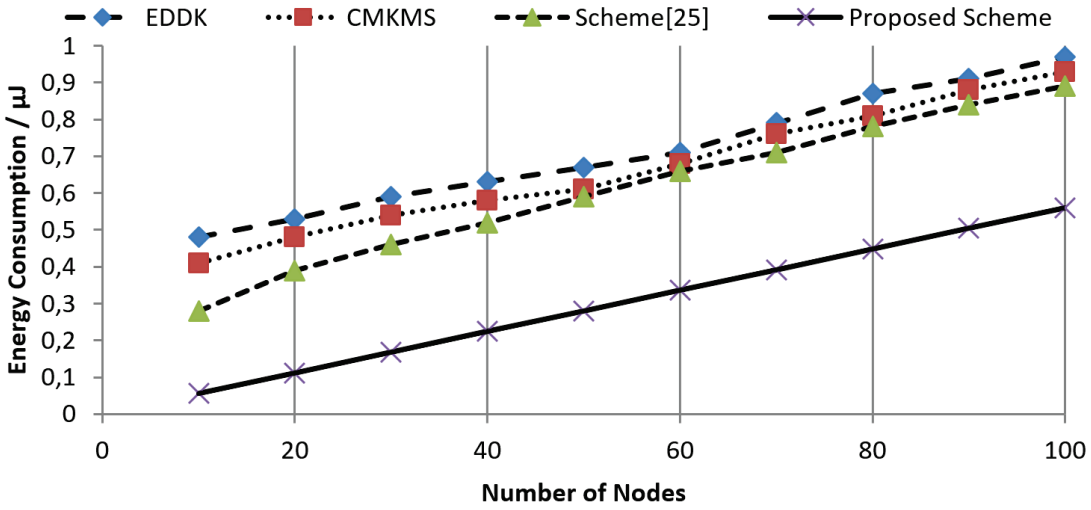

Figure 6 Energy consumption of various Nodes

7 CONCLUSION

This work proposed a safe node migration with dependable handoff happening and fresh links are built up amongst CHs and member nodes. The proposed scheme generated polynomials whenever they are needed by nodes. Polynomials are generated dynamically when change occurs in cluster to create new cluster key (session key). Therefore, sensor nodes need to store only few preloaded messages. Additionally, the presented scheme has improved the scalability of number of nodes in network because of using polynomials for the computation and distribution of cluster keys having dynamic generation after the deployment or change in network. Our scheme frequently refreshes the cluster keys (session key) because easy new polynomial generation CH generated a new cluster key and used the polynomial for the secure distribution of cluster key without encryption and decryption. The proposed key management scheme is secure against eavesdropping and node capturing by using an efficient key management based on dynamic generation of polynomial. The proposed scheme has low communication, storage, and computation without compromising the security of key management. The amount of keys stored in CH is considerably reduced and provides resistance against insider and outsider attacks.

8 REFERENCE

[1] Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer networks, 52*(12), 2292-2330. https://doi.org/10.1016/j.comnet.2008.04.002

[2] Karlof, C., Sastry, N., & Wagner, D. (2004, November). TinySec: a link layer security architecture for wireless sensor

networks. *In Proceedings of the 2nd international conference on Embedded networked sensor systems ACM*, 162-175. https://doi.org/10.1145/1031495.1031515

[3] Abbasi, A. A. & Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks. *Computer communications, 30*(14-15), 2826-2841. https://doi.org/10.1016/j.comcom.2007.05.024

[4] Djenouri, D., Khelladi, L., &Badache, N. (2005). Security issues of mobile ad hoc and sensor networks. In *IEEE Communications Surveys Tutorials, 7*(4), 2-28. IEEE Communications Society. https://doi.org/10.1109/COMST.2005.1593277

[5] Venkatraman, K., Daniel, J. V., & Murugaboopathi, G. (2013). Various attacks in wireless sensor network: survey. *International Journal of Soft Computing and Engineering (IJSCE), 3*(1), 208-212.

[6] Diop, A., Qi, Y., & Wang, Q. (2013, December). An Efficient and Secure Session Key Management Scheme for Cluster Based Wireless Sensors Networks. *In Joint International Conference on Pervasive Computing and the Networked World*, (33-44). Springer, Cham. https://doi.org/10.1007/978-3-319-09265-2_5

[7] Tomić, I. & McCann, J. A. (2017). A survey of potential security issues in existing wireless sensor network protocols. *IEEE Internet of Things Journal, 4*(6), 1910-1923. https://doi.org/10.1109/JIOT.2017.2749883

[8] Al Ameen, M., Liu, J., & Kwak, K. (2012). Security and privacy issues in wireless sensor networks for healthcare applications. *Journal of medical systems, 36*(1), 93-101. https://doi.org/10.1007/s10916-010-9449-4

[9] Derhab, A., Bouras, A., Senouci, M. R., & Imran, M. (2014). Fortifying intrusion detection systems in dynamic Ad Hoc and wireless sensor networks. *International Journal of Distributed Sensor Networks, 10*(12). https://doi.org/10.1155/2014/608162

[10] Zhang, J. & Varadharajan, V. (2010). Wireless sensor network key management survey and taxonomy. *Journal of network and computer applications, 33*(2), 63-75. https://doi.org/10.1016/j.jnca.2009.10.001

[11] Ghafoor, A., Sher, M., Imran, M., & Derhab, A. (2015). Secure Key Distribution Using Fragmentation and Assimilation in Wireless Sensor and Actor Networks. *International Journal of Distributed Sensor Networks*. https://doi.org/10.1155/2015/542856

[12] Saleem, K., Khalil, M. S., Fisal, N., Ahmed, A. A., & Orgun, M. A. (2013, July). Efficient random key based encryption system for data packet confidentiality in WSNs. *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications,* 1662-1668. https://doi.org/10.1109/TrustCom.2013.206

[13] Wong, C. K., Gouda, M., & Lam, S. S. (2000). Secure group communications using key graphs. *IEEE/ACM transactions on networking, 8*(1), 16-30. https://doi.org/10.1109/90.836475

[14] Lin, J. C., Lai, F., & Lee, H. C. (2005, November). Efficient group key management protocol with one-way key derivation. *In The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)* l, 336-343.

[15] Estrin, D., Govindan, R., Heidemann, J., & Kumar, S. (1999, August). Next century challenges: Scalable coordination in sensor networks. *In Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking* 263-270. https://doi.org/10.1145/313451.313556

[16] Zhu, S., Setia, S., & Jajodia, S. (2006). LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. ACM *Transactions on Sensor Networks (TOSN), 2*(4), 500-528. https://doi.org/10.1145/1218556.1218559

[17] Agrawal, C. G. & Kulkarni, J. B. (2014). Enhancing the security in WSN using three-tier security architecture. *International Journal of Innovative Research in Information Security (IJIRIS),1*, 40-47.

[18] Yan, X., Li, B., & Ye, X. (2014). A key management scheme for mobile heterogeneous sensor networks*. J. Naval Univ. Eng, 29*(1), 48-52.

[19] Banihashemian, S. & Bafghi, A. G. (2010, February). A new key management scheme in heterogeneous wireless sensor networks. *The 12th International Conference on Advanced Communication Technology (ICACT), 1*, 141-146. https://doi.org/10.1109/CNSR.2010.46

[20] Naranjo, J. A. M., Antequera, N., Casado, L. G., & López-Ramos, J. A. (2012). A suite of algorithms for key distribution and authentication in centralized secure multicast environments. *Journal of Computational and Applied Mathematics, 236*(12), 3042-3051. https://doi.org/10.1016/j.cam.2011.02.015

[21] Ghafoor, A., Sher, M., Imran, M., & Saleem, K. (2015, April). A lightweight key freshness scheme for wireless sensor networks. *12th International Conference on Information Technology-New Generations*, 169-173). https://doi.org/10.1109/ITNG.2015.32

[22] Li, M., Long, J., Yin, J., Wu, Y., & Cheng, J. (2010, April). An efficient key management based on dynamic generation of polynomials for heterogeneous sensor networks. *2nd International Conference on Computer Engineering and Technology, 5*, V5-460.

[23] Zhang, X., He, J., & Wei, Q. (2011). EDDK: Energy-efficient distributed deterministic key management for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking, 2011*(1). https://doi.org/10.1155/2011/765143

[24] Dilip Babar, S., Rashmi Prasad, N., & Prasad, R. (2014). CMKMS: Cluster-based mobile key management scheme for wireless sensor network. *International Journal of Pervasive Computing and Communications, 10*(2), 196-211. https://doi.org/10.1108/IJPCC-04-2014-0029

[25] Nabavi, S. R., & Mousavi, S. M. (2016). A Novel Cluster-based Key Management Scheme to Improve Scalability in Wireless Sensor Networks. *IJCSNS, 16*(7), 150.

[26] Li, M., Long, J., Yin, J., Wu, Y., & Cheng, J. (2010, April). An efficient key management based on dynamic generation of polynomials for heterogeneous sensor networks. *2nd International Conference on Computer Engineering and Technology, 5*, V5-460.

[27] Zeng, Y., Zhao, B., Su, J., Yan, X., & Shao, Z. (2007, December). A loop-based key management scheme for wireless sensor networks. *In International Conference on Embedded and Ubiquitous Computing*, 103-114. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-77090-9_10

[28] Kausar, F., Hussain, S., Yang, L. T., & Masood, A. (2008). Scalable and efficient key management for heterogeneous sensor networks. *The Journal of Supercomputing, 45*(1), 44-65. https://doi.org/10.1007/s11227-008-0184-2

[29] Kausar, F., Hussain, S., Park, J. H., & Masood, A. (2007, December). Secure group communication with self-healing and rekeying in wireless sensor networks. *In International Conference on Mobile Ad-Hoc and Sensor Networks,* 737-748. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-77024-4_67

[30] Sun, Y., Trappe, W., & Liu, K. R. (2002, April). An efficient key management scheme for secure wireless multicast. *IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No. 02CH37333), 2*, 1236-1240

[31] Saied, Y. B., Olivereau, A., & Zeghlache, D. (2011, October). Energy efficiency in M2M networks: A cooperative key establishment system. *3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 1-8.

[32] Gao, S., van Hoeij, M., Kaltofen, E., & Shoup, V. (2006). The computational complexity of polynomial factorization. *American Institute of Mathematics, 364*.
[33] Rehman, E., Sher, M., Naqvi, S. H. A., Badar Khan, K., & Ullah, K. (2017). Energy efficient secure trust based clustering algorithm for mobile wireless sensor network. *Journal of Computer Networks and Communications, 2017*.
https://doi.org/10.1155/2017/1630673
[34] Rehman, E. & Naqvi, S. H. A. (2018). Ensuring Quality of Service Using Multi-Criteria Quadrant Based Clustering (MCQC) Protocol for Wireless Sensor Networks. *Journal of Information Communication Technologies and Robotic Applications*, 18-29.

Contact information:

Eid REHMAN, PhD student
(Corresponding author)
Department of Computer Science & Software Engineering
International Islamic University Islamabad Pakistan,
Department of Software Engineering,
Foundation University Islamabad, Rawalpindi Campus,
E-mail: eidrehmanktk@gmail.com
E-mail: eid.rehman@fui.edu.pk

Muhammad SHER, Professor, Dr.
Department of Computer Science & Software Engineering,
International Islamic University Islamabad Pakistan
E-mail: m.sher@iiu.edu.pk

Syed Hussnain Abbas NAQVI, Assist Professor Dr.
Department of Computer Science & Software Engineering,
International Islamic University Islamabad Pakistan
E-mail: syed.hussnain@iiu.edu.pk

Anwar GHANI, Dr.
Department of Computer Science & Software Engineering,
International Islamic University Islamabad Pakistan
E-mail: anwar.ghani@iiu.edu.pk