Research Article

A cognitive key management technique for energy efficiency and scalability in securing the sensor nodes in the IoT environment: CKMT

A. B. Feroz Khan[1] · G. Anandharaj[1]

Received: 13 August 2019 / Accepted: 2 November 2019 / Published online: 8 November 2019

Abstract
IoT Consist of interrelated devices such as digital devices, mechanical devices, computer system, sensors, etc. for transceiving the information over the network. The scalability is essential to enhance the energy utilization and the performance of the environment, while mobility enhances the coverage of the system. Key management is vital for encrypted information transmitted over the network. Although many key management techniques are available, still it is a challenging issue concerning energy and computational cost, so robust key management technique is required that guarantees the required security requirements. In this paper, we proposed a cognitive key management technique (CKMT). The CKMT mechanism is helpful for key management and maintenance in a cluster-based mobile environment that reduces the rekeying process which is required for the mobile node when it enters the new location area, thereby reducing the computational overhead and enhances the scalability to large size network which makes our scheme more robust because strong key management technique is important while providing security services. Initially, we form a cluster, then Cluster-Head (CH) will be elected, it's a coordinator node that acts as a key manager. We also made an assumption that the sensors and Cluster-Heads are mobile, they can be able to shift from one position to another. This CH oversees and keeps up the private keys of sensors. When CH changes its location, it will hand over its responsibilities to the other CH in the network for uninterrupted communication. Our scheme uses local keys and foreign keys for each node and pairwise key is used only for the common nodes among the cluster so our scheme reduces the computational cost in the network under mobility condition. The results show that our proposed algorithm lowers the overhead in terms of computational costs, energy consumption, and delay.

Keywords Cognitive key management technique (CKMT) · Cluster-Head (CH) · Key management · Mobility · Cluster based key management scheme (CMKMS)

1 Introduction

The sensor nodes are used in the mobile environment to react to input and to pass the data to the intended recipient through various sensor nodes with the ability to carry data over the network. These sensor nodes are used in many fields of applications such as consumer, industrial and mission critical applications. The requirements for the mission critical applications involve the characteristics of security and mobility. The security concern is important to safeguard the network from malicious attacks and mobility is important to increase the performance and reachability of the network. Sensor networks are prone to various kinds of denial of service (DoS) attacks, e.g. replay attack Security of a sensor network is critical due to its mobility characteristics. Even though many works have been

✉ A. B. Feroz Khan, abferozkhan@gmail.com | [1]PG and Research Department of Computer Science, Adhiparasakthi College of Arts and Science, Vellore, Tamilnadu, India.

SN Applied Sciences (2019) 1:1575 | https://doi.org/10.1007/s42452-019-1628-4

done for the security of IoT networks, all have focused on encryption which depends on key management strategy. Many key management techniques are evolved for wsn for the past few years [1–3], there are still key management issues in encrypted solutions because of the sharing of keys between sensor nodes. Many of the key management mechanisms [1–3] are based on probabilistic during the sharing of the key in the cluster based environment. Probabilistic key management scheme cannot guarantee that the two nodes in the different clusters can establish a shared key, if some of the neighbour nodes could not establish a shared key then they cannot not able to participate in the network. Also, the same key is reused by more than one node so if any of the nodes is compromised then the network would be under greater risk. Hence we proposed an improved cluster-based key management technique in which we use three keys, local key, pair-wise key and foreign key, pair-wise key, and local key are established during the initialization phase and no further set up is required. So communication overhead is minimized, foreign key is established between Cluster-Head (CH) for secret communication when the cluster node moves to a different region. This cluster based mechanism has lower the computational cost and increases the efficiency of energy in the network. We consider a cluster based scheme because cluster based environment works well in reducing the energy consumption and increasing the scalability [3].

Currently available sensors are more prone to a scarcity of energy resources due to the computational cost during the transmission of data [4–6]. Efficient key management help combat huge energy consumption by lowering the computational cost that is required during data transmission [7–9]. We have examined lots of challenges in this issue and they can be overcome by the new key management scheme that we proposed here. This algorithm goes into two phases set up phase and maintenance phase, the former will perform clustering and key distribution, and Later will take the responsibility of maintaining keys during mobility condition. The algorithm we used here is ECDSA: Elliptic Curve Digital Signature Algorithm which is an encryption algorithm that can be used to make the encryption faster with less energy consumption and delay [10]. Our work is compared with existed key management scheme the CMKMS, Cluster based key management scheme and performance analysis done based on energy level, computational overhead, and delay.

The rest of this paper is organized into 6 sections. In Sect. 2, we discuss the related work in cluster based key management techniques with their pros and cons. In Sects. 3 and 4 we proposed our cognitive key management technique (CMKT) with cluster initialization, key management, and maintenance to secure the network and in Sects. 5 and 6, we discuss the security analysis and the simulation of the proposed mechanism. Finally, we conclude the work in Sect. 7.

2 Related work

This section discusses some recent key management techniques based on cluster based environment. The key management techniques are generally used for establishing and maintaining keys for authorized parties. A secure application must implement strong key management strategies so that the algorithm can reduce computational cost [1]. The various key management techniques are examined and compared by considering their scalability nature, shared knowledge of connected nodes in the network, and the authentication.

Nabavi et al. [2] describes a key management scheme for cluster based wireless sensor networks which uses ECDSA algorithm it improvise the probability of sharing of keys between sensor nodes and CH. It improves the performance of key management in the sharing of keys but the computational cost is greatly increased by introducing rekeying process.

Sachin et al. [3] proposed a hash key-based key management mechanism for cluster-based wireless sensor network for wsn which uses random key pre-distribution. The work considered three performance measures, packet loss rate, energy consumption, and delay. The work improves the performance with the establishment of a secured link for one hop and multi hop communication but the cluster heads in the network are not mobile.

Dilip et al. [11] proposed a cluster based mobile key management scheme which uses cluster based approach for the improvement of scalability. The research focuses on the management of keys under mobility conditions. The work shows that it lowers energy consumption. However, it uses RC-5 algorithm which has lower efficiency compared to our proposed scheme.

Robision et al. [12] proposed an enhanced cluster based key management scheme which uses high power nodes to form clusters and it balances the load among the cluster. The work achieves high throughput and it reduces delay. However, the nodes in wsn may suffer from congestion and no proper scheduling methods involved.

Xing et al. [13] describe an energy-efficient distributed deterministic key management scheme (EDDK) scheme which uses the elliptic curve digital signature algorithm. It lowers the energy consumption, but the computational cost might increase because it considers local keys and pairwise key for each cluster and so computation cost get increased.

From these studies, we came to know that all the algorithms are well defined, but the performance degradation is due to mobility characteristics of the node and so they cannot scale well in the network [14]. Here we propose an improved key management algorithm that uses local keys and foreign keys for each node and pairwise key is used only for the common nodes among the cluster so our scheme reduces the computational cost in the network under mobility condition. In this work, the performance of our algorithm is compared with cluster based key management scheme (CMKMS) algorithm using NS-2 simulator environment and the results are produced based on the energy consumption, time and computational cost. The main idea of our work is to produce an efficient key management method which is scalable and with less computational overhead under mobility condition.

3 Technology used

3.1 CKMT: a cognitive key management technique

This algorithm the CKMT is proposed here for reducing the computational cost and to increase the energy efficiency as well as scalability of the sensor networks. Figure 1 shows the cluster formation by grouping the sensor nodes into clusters. Clustering can be done to develop an efficient key management technique because clustering helps to increase the efficiency of energy as well as reducing the transmission time [15–18]. In cluster based technique we perform clustering then we elect CH node which will act as a coordinator or key manager in the first phase then the maintenance of key is performed in the next phase. The clustering is performed by dividing the nodes in the network into clusters. The CH node is elected based on the efficient reachability of the node and the maximum distance between all the nodes in the network. The CH node will have all the information about the interconnected nodes, it will get updated when any changes in the topology and this information is further passed to all the nodes in the network. The primary benefit of clustering is that the aggregated information is sent to the Base Station thereby reducing energy consumption. As mentioned earlier, we made a speculation that CH and the nodes in the environment can able to move from one position to another.

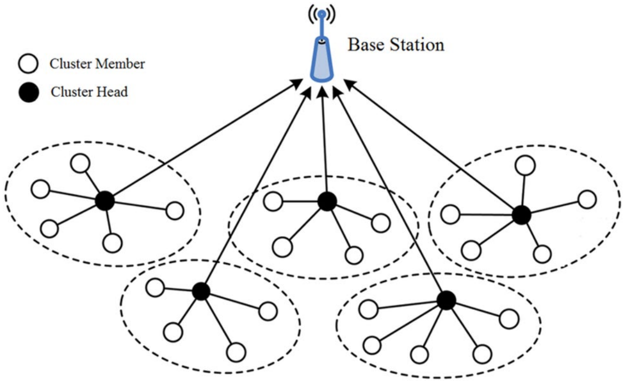


Fig. 1 Cluster formation

The sensors belong to the same cluster can be communicated to each other via CH, CH will collect the quantum of information received from the sensors and it transfers the same to another CH in the other cluster or to BS for the purpose of transmitting the data to the requested node. Here communication between CN–CN is considered as intra-cluster communication, the CH–CH or CH–BS are considered as inter-cluster communication. The algorithm CKMT considers the CH node as a key manager because CH and BS are fixed while sensors and CH are mobile. The node–node communication is happening via CN–CN link and CH–CH or CH–BS. The inter-cluster communication is done via CH–CH link and intra cluster communication is done via CN–CN link.

The algorithm considers the CH as a key manager which manages the key of all connected nodes with in the clusters.

When any of the nodes changes its position, the key management process will get started.

4 Proposed work

This CKMT algorithm consists of two phases, Phase1 is cluster initialization. In this phase, the requirements for partitioning the network into several clusters, the formation of the cluster then CH node selection are all done. Phase2 is cluster maintenance, here adding and removing the internal nodes, changing CH node if required are done.

4.1 Cluster initialization

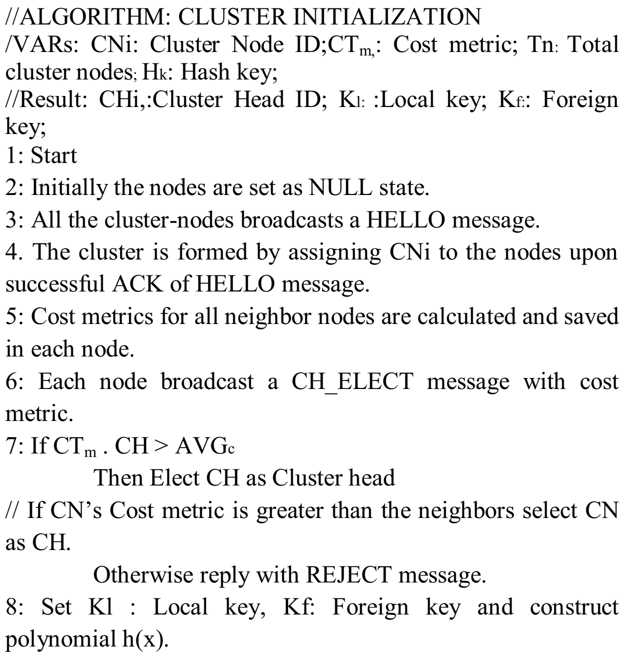
Before the start of the cluster initialization phase, all nodes in the network are in the NULL state. Once the cluster initialization phase is started, HELLO message will be broadcasted from each node in the environment just to gather the knowledge of the internal nodes, which can be used to calculate its cost metric. (Cost metric is the cost value used by routers to determine the best path to a destination network).

Later, every cluster node (CN) in the network will broadcast a CH_ELECT message with cost metric value and the degree of stability of all the interconnected nodes. After receiving the value, each node will compare the cost metric with itself, and the node having the largest value and associativity stable can be elected as a CH node. Finally, the elected CH node will broadcast the elected message (CH-CLAIM) to the nodes in the network. After receiving the CH_CLAIM message, the neighbor nodes will send RTJ message to the CH for the purpose of joining under the coordination of CH, and in turn, CH will send ATJ message back for confirming the joining request of each node. Here maximum efficiency is considered for the elected CH is based on the degree of stability so that CH might scarcely accomplish link disconnection.

The parameters used in CKMT algorithm are as follows:

- CN_i: Cluster node ID;
- CHi: Cluster Head ID;
- CT_m:Cost metric
- Kl: local key;
- Kf: foreign key;
- AVGc: average cost metric;
- Tn : Total nodes in a cluster
- Tc: Total cluster in the network;
- NC: average number of cluster neighbor;
- Hk: hash function.

The following steps are involved in the cluster initialization.



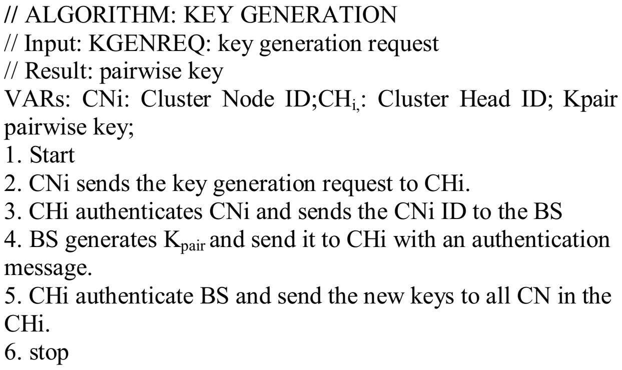
```
//ALGORITHM: CLUSTER INITIALIZATION
/VARs: CNi: Cluster Node ID;$CT_{m,}$: Cost metric; Tn: Total cluster nodes; Hk: Hash key;
//Result: CHi,:Cluster Head ID; Kl: :Local key; Kf:: Foreign key;
1: Start
2: Initially the nodes are set as NULL state.
3: All the cluster-nodes broadcasts a HELLO message.
4. The cluster is formed by assigning CNi to the nodes upon successful ACK of HELLO message.
5: Cost metrics for all neighbor nodes are calculated and saved in each node.
6: Each node broadcast a CH_ELECT message with cost metric.
7: If $CT_m$ . CH > $AVG_c$
        Then Elect CH as Cluster head
// If CN's Cost metric is greater than the neighbors select CN as CH.
        Otherwise reply with REJECT message.
8: Set Kl : Local key, Kf: Foreign key and construct polynomial h(x).
```

The cluster initialization phase is split into two sections, the formation of clusters and set up cluster ID for the unique identification of each cluster in the network. This phase is also used to set up the secure path between the clusters for secure communication of each node in the network. Here each node in the network maintains two keys K_l: Local key and K_f, Foreign key. As mentioned earlier local key is used for intra communication between cluster nodes and CH, and the foreign key is used for intercommunication between clusters (CH–CH/CH–BS) in the network.

4.2 Key establishment

Our proposed protocol consists of two keys, local key and foreign key that should be maintained by each node in the cluster. *Kh* is the notation for the home key which is used for intra cluster communication with its CH. K_f is the notation for the foreign key which is used for inter communication during mobility that is when the nodes belong to the home network enter the new location area (v). These keys are used for secret communication among the nodes in the cluster. After the cluster formation phase is over, the next step is to secure the wsn with the help of the pair wise key K_{pair}. During intra cluster communication, BS will generate pair wise key using one-way hash function and send it to each CH with an authentication message upon the request received from the CHi. Here BS will encrypt the message with K_{pair} using local key K_h as an authentication response and send the response to the legitimate CHi. CHi authenticates BS and send the new keys to all nodes in the same cluster. If CNi is compromised then the corresponding CH will pass this information to all the nodes in the cluster then it removes the compromised node from the network. If CH itself is compromised, then the CNi in the corresponding CH will be distributed to the uncompromised neighbour CH. If the CH is shifted to another position from its current position then the corresponding CH will transfer its responsibilities to other CH by initiating CH selection process.

The following steps are involved in the key generation process.



```
// ALGORITHM: KEY GENERATION
// Input: KGENREQ: key generation request
// Result: pairwise key
VARs: CNi: Cluster Node ID;$CH_{i,}$: Cluster Head ID; Kpair pairwise key;
1. Start
2. CNi sends the key generation request to CHi.
3. CHi authenticates CNi and sends the CNi ID to the BS
4. BS generates $K_{pair}$ and send it to CHi with an authentication message.
5. CHi authenticate BS and send the new keys to all CN in the CHi.
6. stop
```

4.3 Cluster maintenance

After the completion of the clustering initialization phase, CH and CNs must exchange periodic messages to maintain their relationship. As mentioned earlier, the CH and the internal nodes can be dynamically changed. It is the responsibility of CH to transfer its key manager role to the other CH in the cluster when it is leaving the network.

Since the nodes are in a dynamic environment, it has to adapt to the topological changes for the following cases,

(1) *Deleting or adding nodes* Any cluster node can be disconnected from the connected cluster if there is no periodic broadcast message after a period of time from the CH.

When a new node wants to join a cluster, it sends an RTJ packet to the CH through CN, and the cluster head will send an ATJ message back with K_l and K_f. (local key and foreign key).

(2) *Replacing the Cluster Head Position* Once a cluster head is broken, the node belonging to this cluster would return to the NULL state and new CH is ***elected.***

(3) *Change of Position* Once any CN changes it cluster to another cluster, the leaving node will inform the CH of its previous cluster. This information is passed to all the CNs including other CH in the boundary. Then the node will be provided with K_f from its new CH.

5 Security analysis

The proposed algorithm CKMT is compared with some related schemes and then analyze CMKT in terms of both energy consumption and computational cost. Table 1 surveys the comparison of key management techniques. Key management techniques are compared with the consideration of scalability, node authentication, and their deployment knowledge. In this survey, we came to know that all the listed techniques have proper functionality, but the performance degradation is due to the mobility characteristic of nodes and they do not perform well in scalable environments [19–23]. The main reason for the low performance of EDDK is that it calculates the pair-wise keys when the node changes its neighborhood it affects the calculation of keys, pair-wise keys and foreign keys, which would show the wrong instance of pair-wise keys and recalculation of pairwise key is required. The idea behind this is to propose an efficient scalable key management mechanism that we call it as CKMT which properly works under mobility of wsn.

6 Simulation results

The implementation of our proposed work is done by NS-2 by considering the parameters as in Table 2.

The proposed work simulates the CKMT algorithm using Network Simulator (NS-2). Figures 2, 3 and 4 show

Table 2 Network simulation parameters

Parameters	Value
Area	100 × 100 m
Network interface type	IEEE802.15.4
Number of nodes	100
BS position	(50,250)
Initial energy (j)	0.5
Sensing range (m)	30
Packet size (bits)	1024
MAC	IEEE802.15.4
Threshold distance (m)	87
Energy receive	40 nJ
Energy transmit	40 nJ
Routing protocol	Energy aware
Traffic	CBR

Table 1 Key management schemes comparison

Reference	Mechanism used	Scheme	Advantages	Disadvantages
[2]	Probabilistic key pre-distribution	ECDSA	Improved performance against sharing of keys	Computational overhead
[3]	One way hash key based scheme associated with pair-wise key	One-way hash	Improve network performance by establishing secured link for one-hop and multi-hop	Based on symmetric cryptography
[11]	Random key pre-distribution	CMKMS	Improve network in terms of energy, scalability and mobility	Based on RC-5 algorithm
[12]	Random pair wise key distribution	ECKMS	Enhance security by guaranteeing high throughput	Increase network overheads and no proper scheduling methods
[13]	Pair-wise key as well as local cluster keys	EDDK	Improve security by lowering the energy consumption	Increase the computational overhead

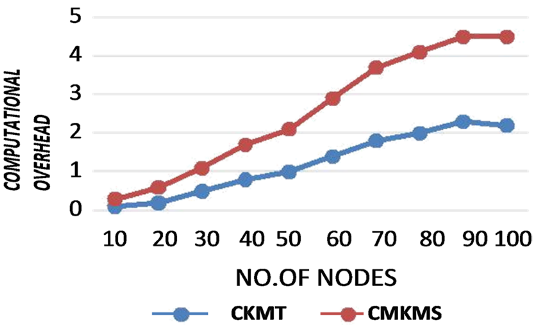


Fig. 2 Computational overhead of CKMT versus CMKMS

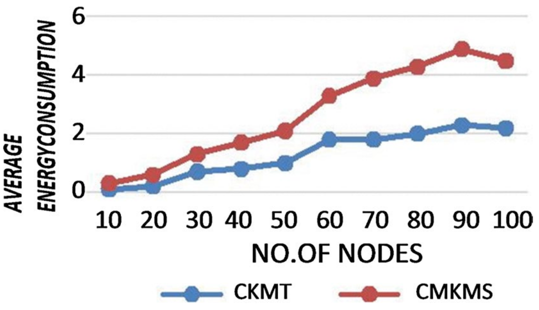


Fig. 3 Average energy consumption of CKMT versus CMKMS

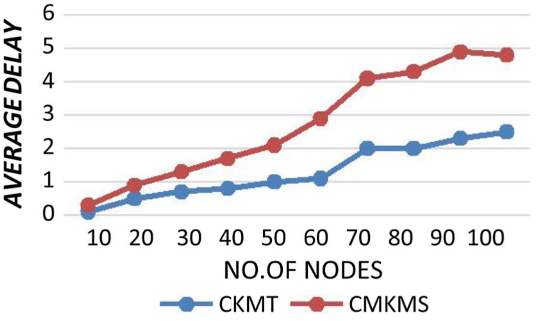


Fig. 4 Average delay of CKMT versus CMKMS

the graphs of computational cost, the average energy consumption and delay by varying the nodes in the network. The result shows that the proposed algorithm CKMT improves energy consumption, delay and reduce computational cost when compared with an existing CMKMS algorithm. One primary reason for the efficiency of our proposed algorithm is that it uses functions with lower complexity than in CMKMS.

6.1 Performance by varying the nodes

The computational cost of encryption and decryption, authentication of data packets and hashing often require additional costs in the setup phase while working with key management techniques [3].

The important reason for the improved efficiency of our algorithm is that it considers local key if the nodes are in the home network and foreign key for the CNs when the nodes are leaving the network and move to other clusters, whereas in existing scheme only local key and pairwise key are used which requires additional cost for generating keys due to the computational overhead that is occurring during the change of the cluster [24–27]. So based on the comparative analysis that we have done here our proposed algorithm CKMT improves the energy consumption, average delay and decreases the computational cost than the existing algorithm.

6.2 Performance of CH under mobility condition

In the existing work, when the CH moved from its current position, the computational cost gets increased because of the new configuration requirements that are necessary to connect into the new network, also rekeying process is required for the nodes in the new environment for secure communication, hence shifting of cluster impact on the calculation of all the required keys [2]. In our scheme, two keys are assigned when the nodes connect in the network local key for inside communication and foreign key is used for communication of nodes beyond the home network thereby reducing the computational cost when the CN moves into another network.

6.3 Performance by varying traffic interval

Figures 2, 3 and 4 shows the average energy consumption, delay, and computational cost by differing the traffic interval respectively. It shows that our proposed scheme improves the energy consumption, delay and computational cost under mobility condition. The main reason for reducing energy consumption is that our scheme uses local keys and foreign keys for each node and pairwise key is used only for the common nodes among the cluster so it reduces the computational cost in the network under mobility condition.

7 Conclusion

There are various key management mechanisms evolved in the field of sensor networks for efficient utilization of network in terms of scalability and mobility but still, it is challenging because of the larger energy consumption and storage overhead. The primary cause of the computational cost incurred because of the lack of scalability in the wireless networks. In this paper, we proposed a Cognitive key management technique in which pair wise key, local key and foreign key of cluster node can be established and maintained securely. The proposed CKMT algorithm increases the performance of the network by reducing the larger computational cost, energy consumption and delay during the key management process. In future, we will concentrate on finding a more efficient solution to improve the performance percentage of mobility effects.

Compliance with ethical standards

Conflict of interest The authors declares that there is no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Vijayakumar P, Azees M, Chang V, Deborah J, Balusamy B (2017) Computationally efficient privacy preserving authentication and key distribution techniques for vehicular ad hoc networks. Cluster Comput J Netw Softw Tools Appl 20(3):2439–2450
2. Nabavi SR, Mousavi SM (2016) A novel cluster-based key management scheme to improve scalability in wireless sensor networks. IJCSNS Int J Comp Sci Netw Secur 16(7):150–156
3. Babar SD, Mahalle PN (2016) A hash key-based key management mechanism for cluster-based wireless sensor network. J Cyber Secur Mobil 5(2):75–78
4. Ma Z, Zhang Z, Ding Z, Fan P, Li H (2015) Key techniques for 5G wireless communications: network architecture, physical layer, and MAC layer perspectives. Sci China Inf Sci 58(4):1–20
5. He D, Zeadally S, Xu B, Huang X (2015) An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad-hoc networks. IEEE Trans Inf Forensics Secur 10(12):1681–2691
6. Diop A, Qi Y, Wang Q (2014) Efficient group key management using symmetric key and threshold cryptography for cluster based wireless sensor networks. IJCNIS 6:9–18
7. Hamsha K, Nagaraja GS (2019) Threshold cryptography based light weight key management technique for hierarchical WSNs. Ubiquitous Commun Netw Comput 276:188–197
8. Naser SM, Croock MS (2018) Proposed simulator based on developed lightweight authentication and key management protocol for wireless sensor network. Int J Comput Digit Syst 7(4):251–260
9. Singh UR, Roy S (2019) Survey on key management schemes and cluster based routing protocols in wireless sensor network. Int J Comput Intell IoT 2(3):576–594
10. He X, Niedermeier M, de Meer H (2013) Dynamic key management in wireless sensor networks: a survey. J Netw Comput Appl 36(2):52–69
11. Dilip Babar S, Prasad N, Prasad R (2014) CMKMS: cluster-based mobile key management scheme for wireless sensor network. Int J Pervasive Comput Commun 10:196–211. https://doi.org/10.1108/IJPCC-04-2014-0029
12. Robinson H, Balaji S, Rajaram M (2016) ECBK: enhanced cluster based key management scheme for achieving quality of service. Circuits Syst 7(8):2014–2024
13. Zhang X, He J, Wei Q (2011) EDDK: energy-efficient distributed deterministic key management for wireless sensor networks. EURASIP J Wirel Commun Netw 2011:765143. https://doi.org/10.1155/2011/765143
14. Qin D, Jia S, Yang S, Wang E, Ding Q (2016) A lightweight authentication and key management scheme for wireless sensor networks. J Sens 2016:1547963
15. Vijayakumar P, Azees M, Kannan A, Deborah LJ (2016) Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks. IEEE Trans Intell Transp Syst 17(4):2149–2151
16. da Silva E, dos Santos AL, Albini LCP, Lima MN (2008) Identity-based key management in mobile ad hoc networks: techniques and applications. IEEE Wirel Commun 15(5):46–52
17. Bayat M, Barmshoory M, Rahimi M, Aref M (2015) A secure authentication scheme for VANETs with batch verification. Wirel Netw 21:1733–1743
18. Shnaikat KN, AlQudah AA (2014) Key management techniques in wireless sensor networks. Int J Netw Secur Appl (IJNSA) 6(6):49–64
19. Kim B, Song J (2019) Energy-efficient and secure mobile node reauthentication scheme for mobile wireless sensor networks. EURASIP J Wirel Commun Netw. https://doi.org/10.1186/s13638-019-1470-9
20. Nabavi SR, Mousavi SM (2018) A review of distributed dynamic key management schemes in wireless sensor networks. J Comput 13(1):77–89
21. Mardini W, Yassein MB, Khamayseh Y, Ghaleb BA (2014) Rotated hybrid, energy-efficient and distributed (R-HEED) clustering protocol in WSN. WSEAS Trans Commun 13:275–290
22. Bala S, Sharma G, Verma AK (2013) Classification of symmetric key management schemes for wireless sensor networks. Int J Secur Appl 7(2):117–138
23. Babar SD (2015) Security framework and jamming detection for internet of things. Aalborg University, Department of Electronic Systems, Aalborg
24. Ali A, Irum S, Kausar F, Khan FA (2013) A cluster-based key agreement scheme using keyed hashing for body area networks. Multimed Tools Appl Int J 66(2):201–214
25. Chakavarika TT, Gupta SK, Chaurasia BK (2017) Energy efficient key distribution and management scheme in wireless sensor networks. Wirel Personal Commun 97(1):1059–1070
26. Shainika M, Hema C (2015) Cluster based mobile key management scheme to improve scalability and mobility in wireless sensor networks. In: National conference on research advances in communication, computation, electrical science and structures, internationaljournal.org, pp 22–26
27. Ferozkhan AB, Anandharaj G (2019) A new framework and defensive techniques for DDOS detection on IoT. Int J Recent Technol Eng (IJRTE) 8(1):680–686

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.