SKMT: A SMART AND EFFICIENT KEY MANAGEMENT TECHNIQUE FOR SECURING THE INTERNET OF THINGS

V.N.Vasanthi[1], S.Perumal[2]

[1]Research Scholar, Department of Computer Science, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, Tamil Nadu, India-600117.
[2]Professor, Department of Computer Science, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, Tamil Nadu, India-600117.
Email: vnvasanthi78@gmail.com[1], perumal.scs@velsuniv.edu.in[2]

Article History: **Received:** 18.04.2023 **Revised:** 03.05.2023 **Accepted:** 10.06.2023

Abstract: *Efficient key management is crucial for secure communication among smart devices in the Internet of Things (IoT). While existing security mechanisms can protect the network from various attacks, efficient key management schemes are necessary to ensure optimal network performance. This research proposes a smart key management technique that utilizes pairwise, foreign, and home keys for intra-cluster communication, communication outside the cluster, and establishment of communication, respectively. The proposed cluster-based scheme is suitable for mobile devices, as it reduces energy consumption and enhances scalability. The proposed protocol also minimizes the computational overhead incurred in the key management process by reducing the rekeying process required for nodes in different networks through the use of foreign keys. The performance of the proposed technique was evaluated in Ns 3 and it is compared with other state-of-the-art models, including TSRM and CKMT, using metrics such as energy consumption, delay, and throughput. The results showed that the proposed technique outperformed other models in terms of these metrics, highlighting its potential for enhancing the security and performance of IoT networks.*

Key words: *Key management; security; IoT; cluster*

DOI: 10.48047/ecb/2023.12.10.507

1. INTRODUCTION

The sensor nodes are used in different applications such as military, medical, farming, water management, home automation, smart city, smart garbage, etc. In IoT, sensor nodes are playing a key role in the transmission of information. This information could be private or sensitive information since most IoT applications comprise sensitive information. Most of the sensor nodes are left unattended which induces the attacker to perform some attacks such as eavesdropping or extracting confidential information. The security of sensor nodes is a critical issue because of their dynamic nature [1-4]. Many kinds of research have been evolved to protect the sensor network. However, all those are focused on encryption that uses a key management strategy to secure the environment. Secure communication is an important concern to prevent the network from catastrophic attacks such as DDoS attacks. To provide secure communication among the sensor nodes in IoT, key establishment is a challenging task when the key is distributed among the nodes. There are many attacks possible such as node capture attacks that may destroy the security of the entire network due to the exposure of the key during its distribution to other nodes inside or outside the network. The network can be saved from such disasters if a reliable and secure key management technique is employed in it. There are many key management schemes available, but most of those schemes are probabilistic for the distribution of keys in the cluster-based environment [5]. The probabilistic-based schemes could not guarantee the security of a key when it is shared between different clusters. A node cannot participate in the network if it could not establish a shared key. Alternatively, if more than one node uses the same key then the network may be prone to severe attack if any one of the nodes is compromised. The current key management mechanisms incurred large computational costs which leads to performance degradation mainly the energy consumption increases due to the increase of the computational costs. This

is because the currently available schemes cannot cope with the dynamic nature of the sensor network. To overcome these issues, an efficient key management technique is proposed in this section. The work proposes 3 keys local key, pairwise key, and foreign key to be established. The foreign key is required when the nodes are moved to other clusters. The local key is used when the nodes in a similar cluster are communicated. Pairwise keys are established and shared with all the nodes to ensure the authenticity of the device. Hence the proposed scheme reduces the computational costs when the mobile nodes are changing location.

2.RELATED WORK

The latest cluster-based key management techniques are discussed here. We observed that most of the techniques are based on probabilistic and are not scalable. Hence they cannot be adapted to resource-constrained nodes that are dynamic. Mousavi et.al 2016 [6] introduces a unique key management mechanism that uses the ECDSA scheme for the distribution of shared keys. The results show that the network performs well with less energy and delay. However, the network is experiencing increased computational costs during the mobility of CN. Sachin et.al 2016 [7] proposed a hash-based key establishment technique for the establishment of the secret key. The work uses pre-distribution of random keys during communication which increases the performance with reduced energy, time, and delay. The results show the increased achievement of the proposed work but the CH used in this scheme is not mobile. Dilip et.al 2017 [8] described a mobile key management scheme based on the RC-5 algorithm which is well scalable and the network performance increased. However, the proposed method is less efficient than our proposed method because of using the RC-5 algorithm. Robinson et.al 2016 [9] proposed an efficient key management method based on the cluster network. The proposed algorithm selects high-power nodes to create clusters and it effectively manages the load among cluster nodes. The performance of the proposed work is enhanced and the network achieves high throughput and efficient energy. Zhang et.al 2019 [10] describes a novel key management strategy for the distribution of shared key securely in the network. The authors proposed an EDDK scheme based on ECDSA which uses only two key local key and pairwise key. The proposed method lowers the energy but the computational cost increases during the mobility of CN.

Lin You et al. (2015) [11] propose a novel technique for key management based on DKH-KD. The proposed work establishes an efficient pairwise key well using hash chains. When mobile nodes change location, however, the rekeying procedure incurs a higher computational cost.

Danyang et al. 2016 [12] presented an efficient AKMS key management system. Using the Lightweight authentication scheme to authenticate sensor nodes during communication, the proposed method effectively improves the efficacy of the network.

Ehdai et al. 2016 [13] implement a 2D hash key management mechanism that can effectively prevent the node capture attack. The proposed work improves the network's efficacy by decreasing delay and energy consumption, but the computational cost increases when sensor nodes are relocated to other locations.

LA. Ahlawat et al. (2017) [14] propose an innovative defense scheme based on the random pre-distribution of keys. The proposed method effectively detects the path's vulnerability and improves network performance by increasing energy efficiency and delay.

Priyanka et al. (2015) [15] described a mobile key management scheme that exploits the node capture attack based on a hash chain. The proposed work analyses the probability of each cluster being compromised and creates a hash chain based on this analysis. The algorithm effectively reduces node capture attacks, but it increases computation overhead and the re-keying process across clusters.

Anita et al. 2015 [16] propose a hybrid key management technique to provide a secure channel during sensor node network communication. The proposed work decreases computational overhead and improves network efficacy. However, implementing the algorithm in a real-time setting is difficult.

From the preceding observation, we can conclude that all related schemes are reliable and effective and that the low performance is due to the mobility of cluster nodes. All of the designs have the significant drawback that they are not scalable to accommodate the mobility of nodes. The primary cause of this issue is that they only provide a secure connection for one-hop nodes, which is unsuitable for large networks [17-20].

A further essential finding of the study is that none of the currently available schemes are suitable for the IoT environment because the secure link is only established between nodes with a single hop.

The proposed key management mechanism is a hash-based system for single-hop and multiple-hop nodes. The proposed work is evaluated in the presence of a DDoS attack using a network simulator, and the outcome demonstrates that the work improves network performance concerning energy, latency, throughput, and computational cost. The proposed plan is contrasted with all of the related plans that were discussed. We observed that the proposed work safeguards the network from jamming issues with a secure and efficient key management technique and that the network's performance improves compared to all related schemes discussed in this section. The performance evaluation of the proposed work takes into account three parameters: energy, delay, and computational cost. The computational cost was incurred during authentication and encryption, as well as the establishment of secret keys during mobility. Energy consumption is the average amount of energy consumed by network elements.

3.THE PROPOSED WORK

This section introduces a secure key management scheme called a smart key management technique (SKMT) which effectively establishes the shared key among all the nodes in clusters. The proposed work is well applicable to cluster-based networks where all the nodes are divided into clusters. Each cluster has a coordinator node called Cluster Head (CH) and one or more sensor nodes in the same proximity.

The maximum distance node in the border proximity region of different clusters is declared as CH. All the nodes in a cluster need to communicate with CH to transfer their messages to BS. CH will maintain the information about the connected nodes in the cluster and any changes in the path need to be informed to CH and the CH will transform the information in the network and all the nodes will be aware of it. Clustering helps to save energy by sending the aggregated information to BS. Figure 1 shows the design of the cluster network. In this model, a network consists of sensor nodes CNs, BS, and CH. The network is clustered into many forms based on the proximity of the nodes in the network and the nodes are grouped into similar clusters if they are in the same ranges.

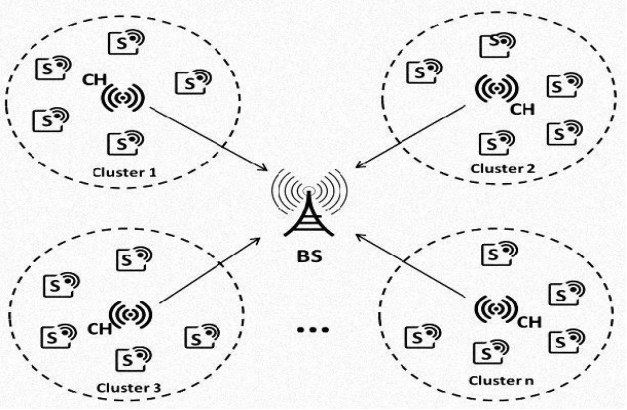

Figure 1: Cluster formation

All the communications between sensor nodes cannot be direct and it should occur via CH [21,22]. A CH will act as a central node in the cluster, it will receive the messages from the nodes and transfer the aggregated data to the BS and thus saving energy. The key management scheme proposed in this section act as a key manager for the connected nodes in the cluster. The algorithm undergoes two phases for cluster initialization and setup. Phase 1 will first divide the network into different clusters and helps in the formation of clusters. Then CH node will be elected based on the maximum distance of the node. Phase 2 is used for the maintenance of the network. This will take the responsibility of adding and removing the nodes in the network and changing the CH node upon communication failure.

3.1 Cluster Initialization

All nodes are initially set to a NULL state during the cluster initialization phase. After the cluster has been configured, each node in the network will broadcast a HELLO message to gather information about the other nodes connected to it. This information is needed subsequently to determine the optimal path for transmitting data to the destination node [23-25]. After collecting information about its neighboring nodes, each node will send a CH_EL message to designate a cluster head node. This message comprises information about the cost metric and stability level of a node. Ultimately, CH will be selected based on the stability and cost metrics. The node with the highest cost metric value and a high degree of stability is known as a CH node.

Now, the CH will send this elected information to all cluster nodes via CH_CLAIM. As a joining request in response to CH_CLAIM, each node in the cluster will send an RTJ (Request To Join) message. After obtaining the RTJ, CH will confirm the joining request with ATJ (Affirm to join). Once all nodes within clusters have been initialized.

3.2 Establishment of Secret key

Two keys (Kl, Kf) need to be maintained by each node in the network. Kl is the local key and Kf is the foreign key. Another key used in this algorithm is Kpair: the pairwise key that needs to be established during intra-cluster communication. The foreign key is used during inter-cluster communication when the nodes are mobile for secure communication. The pairwise key is generated through one way hash function with the node's device ID. The nodes in different clusters will use a pairwise key to authenticate each other. Malicious nodes can be easily identified during the authentication process of the attacker device.

When the CHi requests BS for a pairwise key, BS will send it by encrypting the message with Kpair using Kl. CH will send the keys to all CNs after authenticating the BS.

So initially all the nodes are assigned with key Ki during the node initialization process. Next, the master key Km is established after establishing the initial key ki. When a node in a network wants to communicate, it requests a pairwise key using its master key Km. This hash chain length is fixed with the compromised probability of every cluster in the network.

For node A → $h_{xa}(h_{ya}(k_i))$ (1)

For node B → $h_{xb}(h_{yb}(k_i))$ (2)

The following formula is generated for node A as a pair-wise key

$h_{xb}(h_{yb}(k_i))$ iff $X_a <= X_b$ & $Y_a <= Y_b$ (4)

Node B's pair-wise key is

$h_{x}a(h_{ya}(k_i))$ iff $X_a >= X_b$ & $Y_a >= Y_b$ (5)

the pairwise key generated for both nodes is

$h_{max}(h_{xa(}h_{ya}(k_i)))$ (6)

/Algorithm 1: Master Key Establishment Algorithm

Input: Set of nodes N in the network, cluster head CH, one-way function f()

Output: Master key MK for the cluster

1. Divide the sensor network into clusters, with each cluster having a cluster head node CH.
2. For each node, i in the network, generate a random key RK_i and assign a unique identifier ID_i.
3. Each node i sends its ID_i and RK_i to the cluster head CH.
4. CH collects the IDs and RKs of all nodes in the cluster.
5. CH computes the master key MK using the one-way function f() as follows:
   MK = f(CH.ID, {ID_i, RK_i} for i in N)
6. CH distributes the MK to all nodes in the cluster using secure communication.
7. Each node stores the MK and its own RK securely.
8. To communicate securely with another node j in the same cluster, node i derive a pairwise key PWK_ij as follows:
   PWK_ij = f(MK, RK_i, RK_j)
9. To communicate with node k in another cluster, node i requests the MK and RKs of the nodes in k's cluster from CH.

10. Using the received MK and RKs, node i derive a pairwise key PWK_ik to communicate securely with node k as follows:

PWK_ik = f(MK, RK_i, RK_k)

The following are the notations used in algorithm 1

N: The set of nodes in the network.

CH: The cluster head node.

RK_i: The random key generated by node i.

ID_i: The unique identifier assigned to node i.

MK: The master key is generated by the cluster head.

f(): A one-way function used to generate the master key and pairwise keys. This function takes one or more inputs and generates a fixed-length output that is computationally infeasible to reverse. The specific function used will depend on the requirements of the system and may vary between implementations.

PWK_ij: The pairwise key generated by node i and node j to communicate securely within the same cluster.

PWK_ik: The pairwise key generated by node i and node k to communicate securely with a node in another cluster.

Algorithm 2: Cluster Initialization Algorithm

Input: Set of nodes N in the network, cluster size s, one-way function f()

Output: Cluster formation and master key MK for each cluster

1. Divide the sensor network into clusters of size s, with each cluster having a cluster head node CH.
2. For each node, i in the network, generate a random key RK_i and assign a unique identifier ID_i.
3. Each node i sends its ID_i and RK_i to the cluster head CH of the cluster it belongs to.
4. CH collects the IDs and RKs of all nodes in the cluster.
5. CH computes the master key MK for the cluster using the one-way function f() as follows:
6. MK = f(CH.ID, {ID_i, RK_i} for i in N within the cluster)
7. CH distributes the MK to all nodes in the cluster using secure communication.
8. Each node stores the MK and its own RK securely.
9. Repeat steps 2-7 for each cluster in the network.

This algorithm follows a similar structure to the master key establishment algorithm, but it initializes the cluster formation and master key establishment process for the entire network. The algorithm divides the sensor network into clusters of a specified size and generates random keys and unique identifiers for each node. Each node sends its ID and RK to its respective cluster head, which collects the IDs and RKs of all nodes in the cluster. The cluster head computes the master key for the cluster using a one-way function and distributes it to all nodes in the cluster using secure communication. Each node stores the master key and its random key securely. The process is repeated for each cluster in the network.

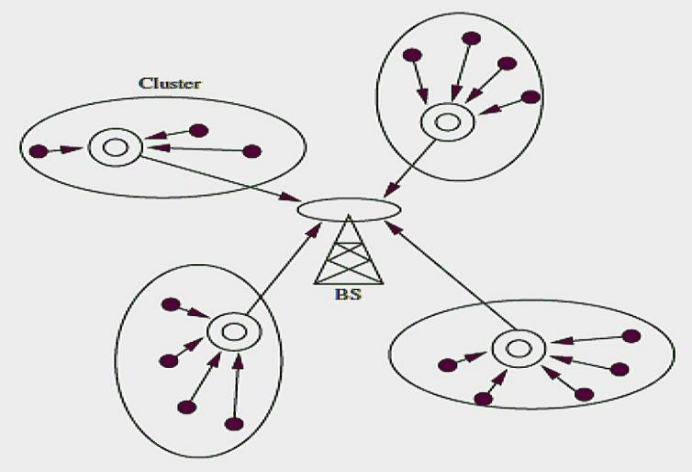

Figure 2: One-hop nodes and multi-hop nodes

After the setup phase, a maintenance phase follows where the Cluster Head (CH) and Cluster Nodes (CNs) exchange HELLO messages periodically to maintain connectivity. The CH assumes the key manager role, which is handed over to another CH when the current CH leaves the network. As nodes are mobile, the network needs to adapt to changes in topology. If the CH does not receive any communication from a node after a certain time, that node can be removed from the cluster. To add a new CN, it sends a Request to Join (RTJ) to the CH, which replies with an Accept to Join (ATJ) using the local and foreign keys. The nodes can be added or removed from the network accordingly. When a CH in a cluster disconnects, a new CH is elected to coordinate CNs in the cluster. If a CN changes its location, the home cluster generates a foreign key for the leaving node to forward incoming messages to their destination.

Algorithm 2 is used for the cluster initialization phase, providing step-by-step instructions on how to set up the cluster. The key generation algorithm above is used for pairwise key generation when the CH requests the Base Station (BS) to generate Kpair.

For authentication of one-hop nodes, an Auth-Req packet is sent to the BS. Figure 2 shows the one-hop nodes and multi-hop nodes in the network. The node requesting to connect sends a request to the BS for authentication, and the BS replies with an Auth-Res using the requested node's public key. After authentication, the node can participate in the network and must send periodic HELLO messages to inform of its availability.

For authentication of multi-hop nodes, the same procedure is followed as in single-hop authentication for each cluster in the network. Multi-hop nodes can establish their pairwise key using a simple one-way hash function, generating a secure link between one-hop and multi-hop nodes.

4. RESULTS DISCUSSION

The proposed work is implemented using discrete event simulator NS-2. The simulation parameters are given in Table 1. The algorithm is implemented in the simulation in the presence of jamming attacks with an increasing number of malicious nodes to analyze the performance of the network. The work considers the following performance metrics to evaluate the proposed work.

- ✓ Performance by increasing attacker nodes
- ✓ Performance by considering energy
- ✓ Performance by considering the delay
- ✓ Performance by considering packet loss rate
- ✓ Performance under mobility scenario
- ✓ Performance without mobility
- ✓ Performance with mobile CH
- ✓ Performance with one-hop and multi-hop nodes

The proposed scheme is compared with the modern scheme such as EDDK and CMKMS. The results are compared in terms of energy, delay, and computational work. The proposed algorithm SKMT is well applicable to the IoT environment after evaluating its performance over the existing schemes. Even though all the key management schemes in related work are efficient they are not well suitable for mobility conditions and are performing well when combined with other schemes. Simulation parameters are given in Table 1.

Table 1: Simulation Parameters

Parameters	Values
Network simulator	NS-2
Simulation Area	100 x 100
Range	100 m
TH value	1.8
Size of data	150 bytes
Rate	256 Kbps

Malicious terminals	35
Traffic rate	1s-10s
Mobility Speed	1-25 km/hr
Number of Nodes	100
Traffic Intervals	1s-10s

In the first set of simulations, the algorithm's performance is tested by varying traffic intervals, attacker nodes, and the random mobility of sensor nodes to measure crucial factors such as energy consumption, latency, throughput, and packet loss rate. The traffic rate varies from 1 to 10 seconds. The fastest traffic rate is measured in 1s, while the slowest pace is measured in 10s. The simulation employs 100 nodes, and the number of attacker nodes introduced into the network varies from 2 to 25. The mobility behavior is monitored at 1 to 25 km/hr. increments.

4.1 Experimental evaluation

Figures 3, 4, and 5 show how the suggested work performs in terms of energy consumption, latency, and computing cost as the number of attacker nodes increases. After implementing the suggested technique, the simulation environment enables jamming assaults with 100 nodes and 5, 10, 15, 20, and 25 attacker nodes. However, the processing overhead spent during authentication, encryption, and hashing as a result of the algorithm's inefficiency leads to an increase in energy usage. The suggested solution tackles these difficulties by using two keys: a local key for internal network communication and a foreign key for communication in a foreign network. As a result, when compared to current techniques, our suggested solution improves energy usage, decreases latency, and minimizes computing overhead.

The suggested method was tested for performance by adjusting traffic intervals, and the results were analyzed for energy, delay, and computing cost. The suggested approach makes use of local and foreign keys for communication inside and across clusters, resulting in increased energy efficiency. Even as the number of attacker nodes increased, the simulated findings revealed a drop in packet loss rate.

The proposed algorithm also efficiently detects and eliminates jamming from the network even when the nodes are randomly mobile. The proposed scheme demonstrated a significant improvement in packet loss rate, energy consumption, and delay compared to existing schemes.

However, when the CH is mobile, the proposed work incurs overhead in terms of energy, delay, and computational cost. In contrast, existing schemes require a rekeying process for establishing the master key and pairwise key whenever a node changes its location, resulting in increased computation cost, energy consumption, and delay. The proposed scheme overcomes this limitation by using a foreign key when a node leaves the home network. The proposed work establishes three keys, namely, the local key, the pairwise key, and the foreign key. The foreign key is required when nodes move to other clusters, the local key is used when nodes in the same cluster communicate, and pairwise keys are established and shared with all nodes to ensure the authenticity of the device.

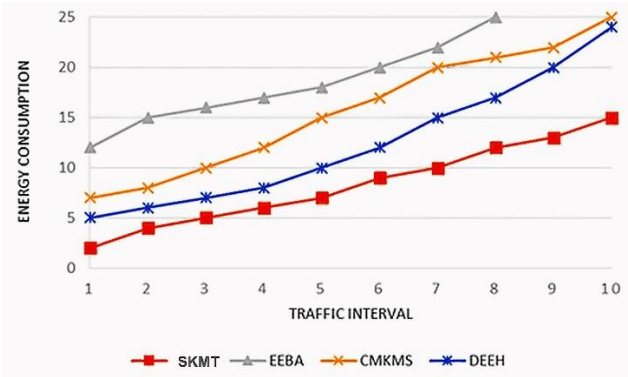

Figure 3: Performance under traffic interval considering energy

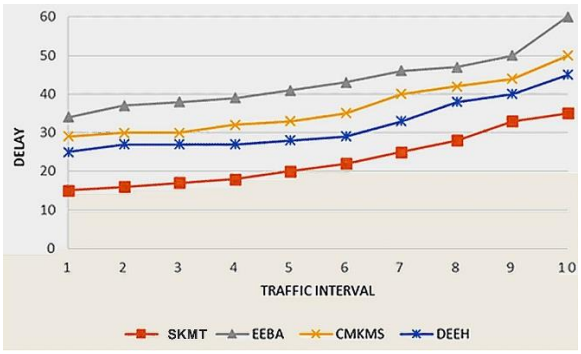

Figure 4: Performance under traffic interval considering Delay

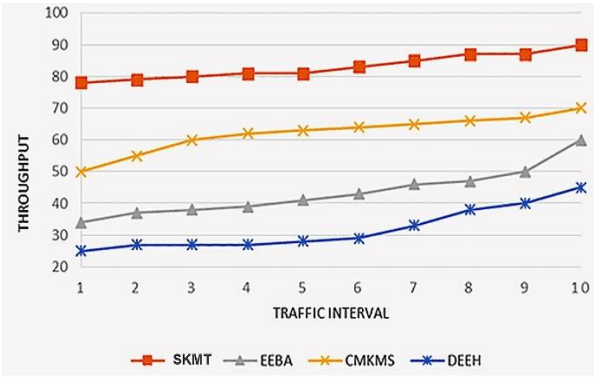

Figure 5: Performance under traffic interval considering Throughput

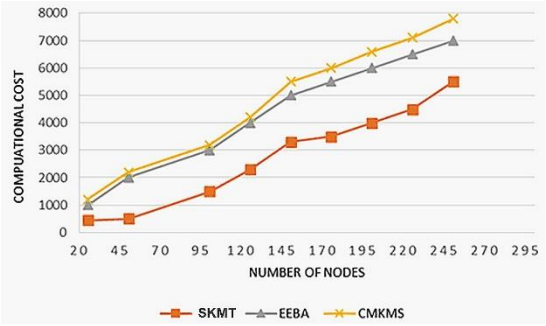

Figure 6: Performance in Mobile CH considering Computational cost

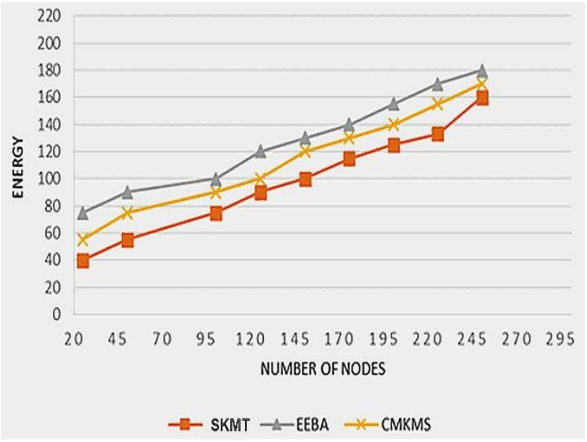

Figure 7: Performance in Mobile CH Considering Energy

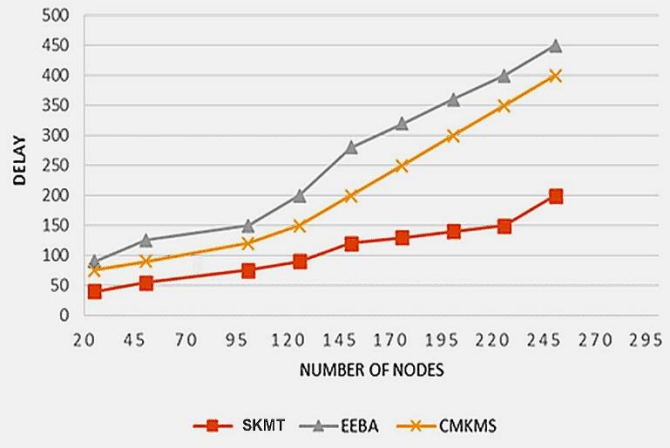

Figure 8: Performance in Mobile CH Considering Delay

5. CONCLUSION

The current key management mechanisms in wireless sensor networks are known to incur large computational costs, which can lead to performance degradation, particularly in terms of energy consumption. This is because the existing schemes are often unable to cope with the dynamic nature of the sensor network, as nodes move around and change location. To address these issues, a new hash-based key management technique for cluster-based wireless networks is proposed in this section. The proposed technique involves establishing three types of keys: local keys, pairwise keys, and foreign keys. Local keys are utilized to communicate between nodes in a similar cluster, while pairwise keys are shared among all nodes to ensure device authenticity. Foreign keys are required when nodes are moved to other clusters. The proposed scheme aims to reduce computational costs when mobile nodes change location, which can help to mitigate scalability issues that can arise due to the dynamic characteristics of sensor nodes. In addition to the proposed hash-based key management technique, future improvements to wireless sensor networks could also involve the use of deep learning algorithms. Deep learning can be used to analyze large amounts of data collected by sensor nodes and can help to identify patterns and anomalies that may be useful for improving network performance. For example, deep learning could be used to detect and prevent security breaches or to optimize network routing. By incorporating deep learning into key management and other aspects of wireless sensor networks, it may be possible to achieve even greater improvements in network performance.

REFERENCES

1. García-Guerrero, E.; Inzunza-González, E.; López-Bonilla, O.; Cárdenas-Valdez, J.; Tlelo-Cuautle, E. Randomness improvement of chaotic maps for image encryption in a wireless communication scheme using PIC-microcontroller via Zigbee channels. Chaos, Solitons Fractals 2020, 133, 109646
2. Kandi, M.A.; Lakhlef, H.; Bouabdallah, A.; Challal, Y. A versatile Key Management protocol for secure Group and Device-to-Device Communication in the Internet of Things. J. Netw. Comput. Appl. 2020, 150, 102480.
3. Hsiao, T.C.; Chen, T.L.; Chen, T.S.; Chung, Y.F. Elliptic Curve Cryptosystems-based Date-constrained Hierarchical Key Management Scheme in Internet of Things. Sensors Mater. 2019, 31, 355–364.
4. Tabassum, T.; Hossain, S.A.; Rahman, M.A. A Secure Key Management Technique Through Distributed Middleware for the Internet of Things. In Intelligent Computing; Arai K., Kapoor S., Bhatia R., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1128–1139.
5. Tlelo-Cuautle, E.; Díaz-Muñoz, J.D.; González-Zapata, A.M.; Li, R.; León-Salas, W.D.; Fernández, F.V.; Guillén-Fernández, O.; Cruz-Vega, I. Chaotic Image Encryption Using Hopfield and Hindmarsh–Rose Neurons Implemented on FPGA. Sensors 2020, 20, 1326.

6. 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.
7. Sachin D, Mahalle PN (2016) A hash key-based key management mechanism for cluster-based wireless sensor network. J Cyber Secur Mobil 5(2):75–78.
8. Dilip Babar S, Prasad N, Prasad R 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, 2014.
9. 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.
10. 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/10.1155/2011/76514 3
11. Lin You, Younan Yuan, et al. A Key Distribution Scheme for WSN Based on Hash Chains and Deployment Knowledge. Int J Distrib Sens Netw 2015;11 (7):640792. doi: https://doi.org/10.1155/2015/640792
12. Danyang Qin, Shuang Jia, Songxiang Yang, et al. A Lightweight Authentication and Key Management Scheme for Wireless Sensor Networks. J Sens;2016:1–9. doi: https://doi.org/10.1155/2016/1547963, 2016
13. Ehdaie M, Alexiou N, Ahmadian M, Aref MR, P.. Papadimitratos 2D hash chain robust random key distribution scheme. Inform Process Lett 2016;116(5):367–72.
14. Ahlawat P, Dave M. A hybrid approach for path vulnerability matrix on random key predistribution for wireless sensor networks. Wireless Pers Commun;94(4):3327–53. 2017.
15. Priyanka Ahlawat, Mayank Dave. An attack model based highly secure key management scheme for wireless sensor networks. ICSCC 2017:7–8.
16. Mary Anita EA, Geetha R, et al. A Novel Hybrid Key Management Scheme for Establishing Secure Communication in Wireless Sensor Networks. wireless personal communications 2015;82(3):1419–33.
17. Singh, S.R.; Ajoy, K.K. Key Management Scheme for Internet of Things Using an Elliptic Curve. J. Comput. Theor. Nanosci. 2020, 17, 115–121.
18. Malani, S.; Srinivas, J.; Das, A.K.; Srinathan, K.; Jo, M. Certificate-Based Anonymous Device Access Control Scheme for IoT Environment. IEEE Internet Things J. 2019, 6, 9762–9773.
19. Sengupta, J.; Ruj, S.; Bit, S.D. A comprehensive survey on attacks, security issues, and blockchain solutions for IoT and IIoT. J. Netw. Comput. Appl. 2020, 149, 102481.
20. Al-Rubaye, S.; Rodriguez, J.; Fragonara, L.Z.; Theron, P.; Tsourdos, A. Unleash Narrowband Technologies for Industrial Internet of Things Services. IEEE Netw. 2019, 33, 16–22
21. Sarenche, R.; Salmasizadeh, M.; Ameri, M.H.; Aref, M.R. A secure and privacy-preserving protocol for holding double auctions in smart grid. Inf. Sci. 2021, 557, 108–129. Khan, A.A.; Kumar, V.; Ahmad, M.; Rana, S. LAKAF: Lightweight authentication and key agreement framework for smart grid network. J. Syst. Archit. 2021, 116, 102053
22. Ani, U.D.; Watson, J.M.; Carr, M.; Cook, A.; Nurse, J.R.C. A review of the use and utility of industrial network-based open source simulators: Functionality, security, and policy viewpoints. J. Def. Model. Simul. 2020, 1–24
23. Airehrour, D.; Gutierrez, J.A.; Ray, S.K. SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of things. Future Gener. Comput. Syst. 2019, 93, 860–876
24. Abosata, N.R.A.; Kemp, A.H.; Razavi, M. Secure smart-home application based on IoTCoAP protocol. In Proceedings of the 2019 Sixth International Conference on Internet of Things: Systems, Management, and Security (IOTSMS), Granada, Spain, 22–25 October 2019
25. Mirshahjafari, S.M.H.; Ghahfarokhi, B.S. Sinkhole+ Cloneid: A hybrid attack on RPL performance and detection method. Inf. Secur. J. Glob. Perspect. 2019, 28, 107–119.

26. Khan, G. A cognitive key management technique for energy efficiency and scalability in securing the sensor nodes in the IoT environment: CKMT. SN Appl. Sci. 1, 1575 (2019). https://doi.org/10.1007/s42452-019-1628-4