Enhancing Data Security in Wireless Sensor Networks via Scalable Hierarchical Key Management

Y Suresh, Krishna Annaboina, N Abhishek, Syed Mohammed Azmal, U P Kumar Chaturvedula

***Abstract*—In recent years, there has been an increase in the utilization of wireless sensor networks (WSNs) for diverse applications, including military sensing and tracking, hospital status monitoring, and traffic flow analysis. The security of data and communications relies on encryption key functions. This research presents a certificate-less effective key management (CL-EKM) protocol for secure communication in dynamic wireless sensor networks characterized by node mobility. When a node enters or exits a cluster, the CL-EKM guarantees the confidentiality of keys in both directions and facilitates rapid key modifications. The protocol facilitates the rapid revocation of keys from compromised nodes, thereby mitigating the impact on the security of other communication channels. Our protocol proficiently mitigates many threats, as indicated by the assessment of our scheme's security. We evaluate the efficiency of CL-EKM in terms of time, energy, communication, and memory within Contiki OS using simulations conducted using the Cooja simulator.**

***Index Terms*—Wireless sensor networks, certificate less public key cryptography, key management scheme, certificate less-effective key management.**

I. Introduction

SENSOR nodes in dynamic wireless sensor networks (WSNs) may move around, the network can cover more ground and provide more precise data than in static WSNs. Among the many monitoring applications that are swiftly embracing dynamic WSNs are target tracking in combat surveillance, traffic flow and vehicle status monitoring, and dairy cattle health monitoring. Consequently, security is a major issue in many vital WSN applications that are dynamic. Thus, dynamic WSNs must address the essential security concerns of node authentication, data confidentiality and integrity, and node mobility. To address security concerns, symmetric key encryption has been the basis for several prior ideas about encryption key management techniques for dynamic WSNs. from [1] to [3]. This type of encryption is perfect for sensor nodes because of their minimal processing power and energy usage.

Innovations in ECC implementation, however, have demonstrated that PKC may be used to WSNs [4]–[10]. For instance, it has been proven that an ECC point multiplication utilizing an 8-bit 8 MHz CPU and an Atmel AT-mega 128 with 160-bit ECC takes less than one second [11]. Not only is PKC more flexible and scalable, but it also protects nodes better from compromise. But we found that existing ECC-based methods aren't completely secure and can be hacked, forged messages, or compromised keys [12]–[14]. The static private key is exposed to both nodes when they set up the session key, which is a big security flaw in [15]. The certificate administration expense per sensor node is too high for large-scale WSNs to use these ECC-based certificate schemes. Pairing operations are computationally expensive and hence inefficient for ID-PKC [16] - [18] approaches. We have not heard of anyone proposing a safe and effective method of key management for dynamic WSNs as of yet.

A solution to the key escrow problem and the elimination of certificate requirements are both accomplished by carefully organizing the entire private/public key pair. With this setup, the user's complete private key is no longer their responsibility [19]. Furthermore, we employ additive group-defined 160-bit ECC keys, which are equivalent to 1024-bit RSA keys in terms of security. A wireless sensor network that is both dynamic and varied is shown in Figure 1. An array of sensors, which can be either permanently installed or moved as required, is managed and data is collected by a base station (BS).

To build CL-EKM, which can dynamically enable node authentication and generate a paired key across nodes, we use a pairing-free certificate less hybrid signcryption scheme (CL-HSC) that we presented in earlier work [20], [21]. Two nodes can share the CL-EKM paired key easily with the help of CL-HSC, all without the hassle of certificate exchange or pairing procedures. Our CL-EKM is built to handle node mobility. Whenever a node moves, it executes lightweight operations to update the cluster key.

Manuscript received January 5, 2025; revised August 22, 2025.

Y Suresh is an Associate Professor of Information Technology Department, Prasad V Potluri Siddhartha Institute of Technology, Kanuru, Vijayawada, Andhra Pradesh, India (e-mail: sureshyadlapati@gmail.com).

Krishna Annaboina is an Asst.professor Computer Science and Engineering (IoT) Department, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Hyderabad, Telangana, India (e-mail: anneboina.krishna@gmail.com).

N Abhishek is an Assistant Professor of Computer Science and Engineering Department, Sreenidhi Institute of science and technology, Ghatkesar, Hyderabad, Telangan, India (e-mail: abhifacmit@gmail.com).

Syed Mohammed Azmal is an Assistant Professor of Computer Science & Engineering Department, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India (e-mail: syedmohdazmal@kluniversity.in).

U P Kumar Chaturvedula is an Associate Professor of Electrical and Electronics Engineering Department, Aditya University, Surampalem, Andhra Pradesh, India (e-mail: contactchaturvedula@gmail.com).

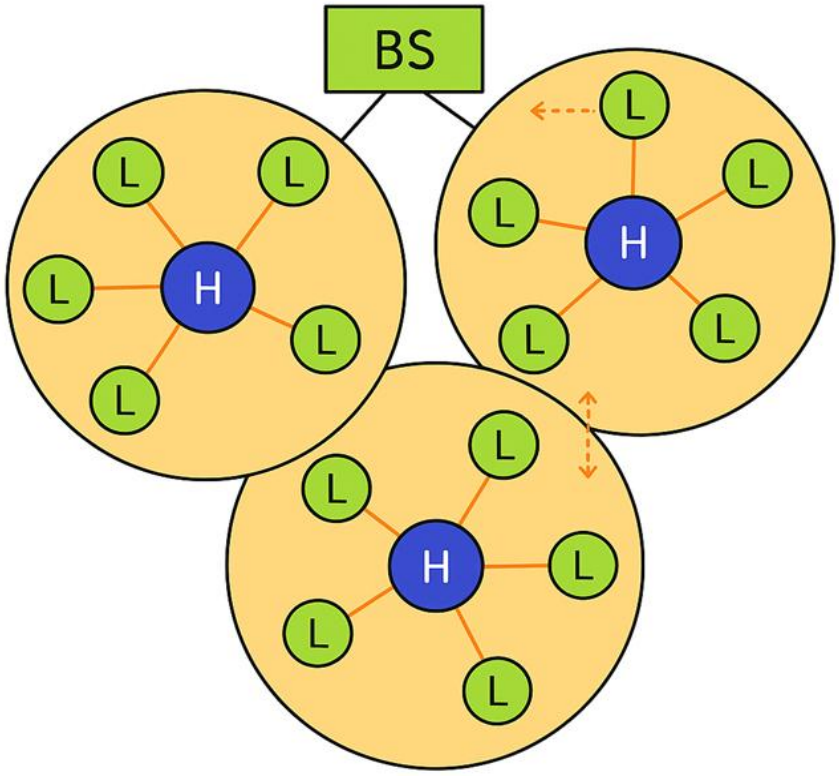

Fig. 1. shows a wireless sensor network that is heterogeneous and dynamic.

When a node is determined as malicious or permanently leaves the cluster, it revokes the key. When it comes to expanding an existing network with additional nodes, CL-EKM can handle it with ease. Your needs for forward and backward secrecy will be met by CL-EKM. Additionally, it safeguards against impersonation, cloning, and node compromise. The results of the security study prove that our plan works.

II. Proposed Certificate less Effective Key Management (CL-EKM) Protocol

In this section, we present the Certificateless Effective Key Management (CL-EKM) protocol, a newly built framework, to address the critical security and efficiency problems affecting dynamic Wireless Sensor Networks (WSNs). Robust methods are necessary for these networks to offer secure communication without putting undue computational and energy demands on them. Resource constraints and frequent topological changes define them. The CL-EKM protocol is a scalable and fast solution for WSN contexts. It implements lightweight key management methods and leverages certificateless cryptography.

Smart cities, healthcare, and environmental monitoring are just a few of the many areas that have started to depend on WSNs. Security issues such as eavesdropping, node capture, and impersonation attacks are common in these open and dynamic systems. While identity-based cryptography (IBC) and classic public key infrastructure (PKI) approaches perform well in traditional networks, they sometimes fall short in WSNs due to their high computational requirements or reliance on a centralized certificate authority. A new paradigm may be emerging to circumvent these limitations: certificate less cryptography. This approach does away with certificate management and helps alleviate key escrow issues related to IBC.

On top of that, the CL-EKM protocol takes this idea and applies it to WSNs, providing a solution that is both efficient and secure. Despite frequent additions, deletions, and relocations of nodes, the protocol is designed to adapt dynamically to the mobility and scalability requirements of WSNs, ensuring secure key management.

A. Design Principles of CL-EKM

The foundational ideas of the CL-EKM protocol are as follows: An method to cryptography known as "certificate less" (CL-EKM) does away with certificate authority and the associated costs of issuing, distributing, and verifying certificates. In addition to solving the key escrow issue, this method divides up key generation duties between a central KGC and individual sensor nodes.

Processor and power constraints on WSN nodes need a lightweight protocol that makes use of efficient algorithms for key generation, distribution, and updates.

Flexible Response: CL-EKM is engineered to manage the ever-changing nature of WSNs, guaranteeing the security of communication regardless of whether nodes join or exit the network. Methods for efficiently updating keys and assigning roles to nodes allow for this adaptability.

Resilience and scalability: the protocol makes sure that adding or removing compromised nodes doesn't undermine the network security as a whole. It does this by removing infected nodes from the network and letting in healthy ones to join the fray without having to reset everything.

The protocol includes both periodic and event-triggered key changes to prevent possible security breaches. Without interfering with current communication, these updates render compromised keys obsolete. The KGC issues a partial private key to each newly joined node, and it is up to the node to generate the rest of the key on its own. Existing nodes are not affected by this, allowing for smooth integration. The network has effective revocation methods to isolate compromised nodes and update surrounding nodes' keys in the event of a node failure or compromise.

B. Security Features

To meet the unique security needs of WSNs, the CL-EKM protocol includes:

The protocol reduces the likelihood of key escrow by dividing up key generation tasks across participating nodes; this way, no one entity, including the KGC, would ever have full ownership of the private keys.

Security Measures: The protocol is built to resist typical assaults on WSNs, such as replay, impersonation, and man-in-the-middle attacks, by utilizing robust authentication and encryption procedures.

The forward and backward secrecy methods make sure that if a key is compromised, it won't impact the security of any conversations that have already taken place or those that will take place in the future.

In simulated dynamic WSN situations, the CL-EKM protocol was tested against conventional key management systems to confirm its efficacy. Computing overhead, energy usage, and communication latency were some of the important variables that were examined. The results showed that CL-EKM significantly reduces energy usage and overhead while keeping security and scalability at high levels. For WSN security, the suggested certificate-less Effective Key Management (CL-EKM) protocol is a giant leap forward. By merging certificate less cryptography with lightweight and adaptive key management operations, CL-EKM tackles the unique issues of dynamic and resource-constrained WSN systems. With its efficient performance, strong security features, and scalable architecture, it shows promise as a solution for next-gen wireless sensor networks' safe and dependable communication needs.

To address the needs of dynamic WSNs for a secure, scalable, and lightweight key management solution, the CL-EKM protocol is presented. This protocol utilizes certificate less cryptography. When it comes to key updates, node mobility, and compromised node isolation, CL-EKM has you covered with strong security and little CPU usage. The protocol's security is examined in depth in the section that follows.

III. Performance Evaluation

While not strictly necessary, it is suggested that authors provide their full names in the author field. Separate the writers' initials with a space. We used the Contiki port of TinyECC for the elliptic curve cryptography library and integrated CL-EKM into Contiki OS.

A. Performance Analysis of CL-EKM

We do this by evaluating how well each of the three phases—encapsulation, decapsulation, and pairwise encryption key generation—carries out in the process of establishing a pairwise master/encryption key. Computing time and energy usage are the metrics by which we measure each process step. This experiment takes heterogeneous WSNs into account by varying the processing power, or CPU clock rate, of the sensors.

The duration of the process for generating paired keys is depicted in Figure 2. See Figures 2(a), 2(b) for an explanation of why the ECC operations cause the pairwise master key creation to consume the majority of time. The paired master key can only be used to generate the short-term pairwise encryption key, though. There is no need for additional ECC procedures once the pairwise keys have been established between two nodes. Figure 2(a) displays the encapsulation computation times for different sensing device CPU clock rates. The length of the ECC key bits affects the computation time. The time required by secp192r1 is over 1.5 times that of secp160r1. Secp128r2 is around 4% faster than secp160r1. The time required for key encapsulation is 5.7 seconds when using secp160r1 and a CPU clock rate of 25MHz. The decapsulation processing time is illustrated in Fig. 2(b). When compared to encapsulation, decapsulation takes approximately 1.57 times as much processing time from the CPU. This is due to the fact that encapsulation consists of just four ECC point multiplications, whereas decapsulation contains six. Fig. 2(c) concludes with the computation time for paired encryption key establishment. When compared to the preceding two processes, its 5 ms requirement at a 25MHz CPU clock rate is utterly insignificant. This is because a single 128-bit AES operation and an HMAC are all that are required at this stage. The next step is to calculate the power usage. From Fig. 3, we may deduce that the higher the processing power, or CPU clock rate, the greater the energy consumption. On the other hand, 25MHz yields quicker computation than 16MHz, as demonstrated in Figure 3(a) and Figure 3(b), but there is no discernible difference between the two. Also, since secp160r1 uses less CPU time and energy for WSNs and is more secure than secp128r2, it could be a decent option for elliptic curve selection. We use secp160r1 in our following experiments.

B. Performance Comparisons

Here, we compare our system to three other key management systems for dynamic WSNs that have used ECC: HKEP [23], MAKM, and EDDK [24]. The pairwise master key generation stage is the most time intensive part of all the schemes, thus we decided to compare its performance instead of the schemes' overall variability. We compress EC points to reduce packet size and use Low electricity Listening (LPL) to save electricity. Sensors briefly awaken in order to look for transmissions on a second-by-second basis. In the absence of sound, they revert to their sleeping state. We determine the communication energy consumption by combining the CC2420 power consumption statistics with the IEEE 802.15.4 protocol's overhead data. In Figure 4, we

can see two potential results. Before anything else, the distance between the H-sensor and the L-sensor could be anywhere from one to eight hops, as illustrated in Figure 4 (a) (first scenario). In the second scenario, as shown in Figure 4 (b) (wireless channel conditions vary), a 1-hop range is maintained by the H-sensor and two L-sensors. If the wireless channel is not in good condition, the sender may attempt to resend the packet multiple times to the destination. In a perfect world, how many packages would make it there unscathed? ETX stands for the expected transmission count. In Figure 5, we can see how much power each of the four approaches uses in a paired key configuration as the number of hops (n) between the L-sensors and the H-sensor increases. Our method utilizes less energy than HKEP when n=1 because our method builds a pairwise key between two L-sensors with only two message exchanges, whereas HKEP requires six. When n=1, MAKM consumes the least amount of energy compared to other systems that do expensive ECC procedures. This is because the L-sensor only performs one AES symmetric encryption. However, MAKM's energy increases as n increases because the H-sensor is continuously involved in producing a pairwise key between two L-sensors. Therefore, MAKM consumes more energy than our system for n > 1, and this difference widens as n increases. Packet delivery in WSNs is unreliable because of low-power transceivers, obstacles that are difficult to forecast, and wireless channel conditions that change over time. Figure 6 shows the four options for a pairwise key establishment and their energy usage as a function of ETX from 1 to 4. With ETX, HKEP's energy consumption increases exponentially due to the six message exchanges it requires. Another security flaw with HKEP is that when two nodes establish a session key, the other node can see the static private key of the first node. Despite the fact that EDDK and MAKM may offer better performance due to reduced computational overhead, our system lags behind them by 0.045 J and 0.121 J, respectively.

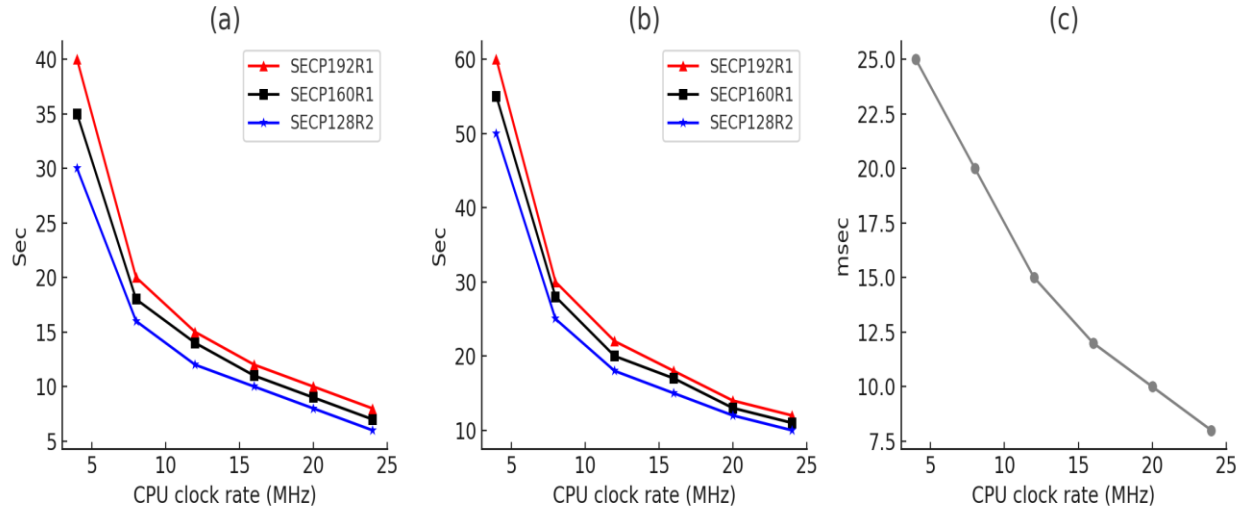

Fig. 2. Computing the overhead for establishing a master/encryption key pairs. a) Condensing important data. (a) Extracting important data from a sphere. (c) Establishing encryption keys in pairs

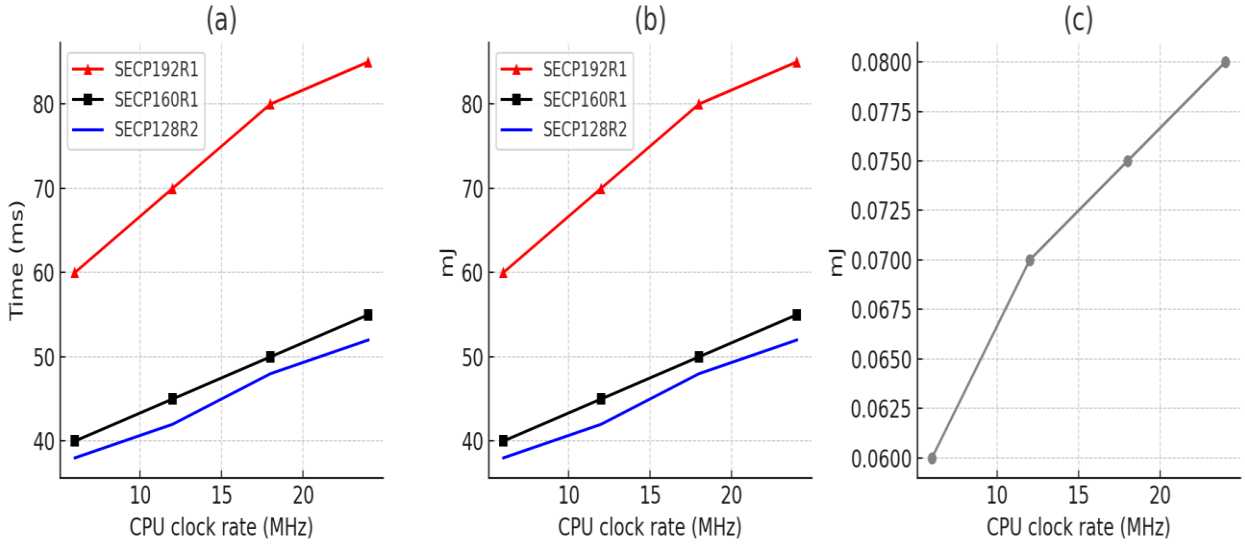

Fig. 3: Power usage during the creation of master/encryption keys in pairs. (a) Condensing important data. (b) Extracting important data. (c) Establishment of encryption keys per pair.

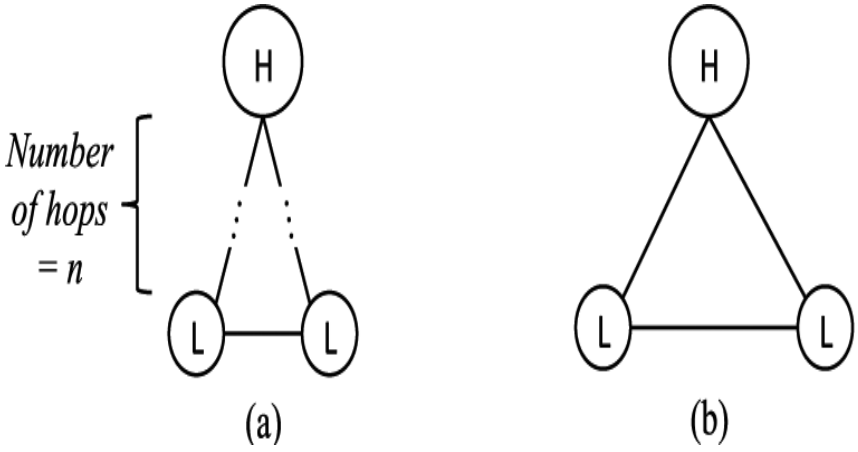

Fig. 4. Topology of the network.

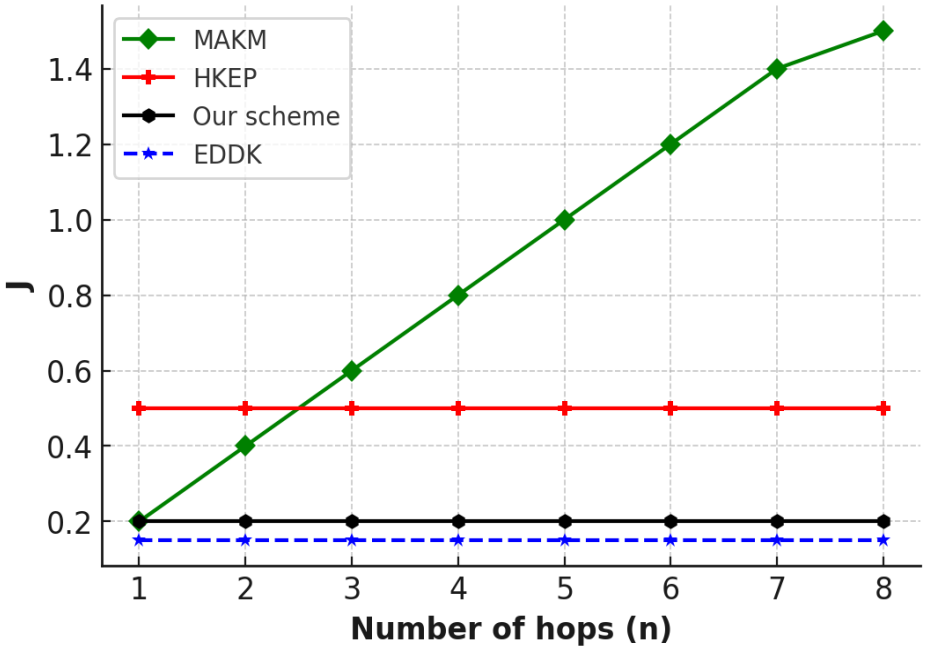

Fig. 5. Efficiency analysis of paired key creation in scenario

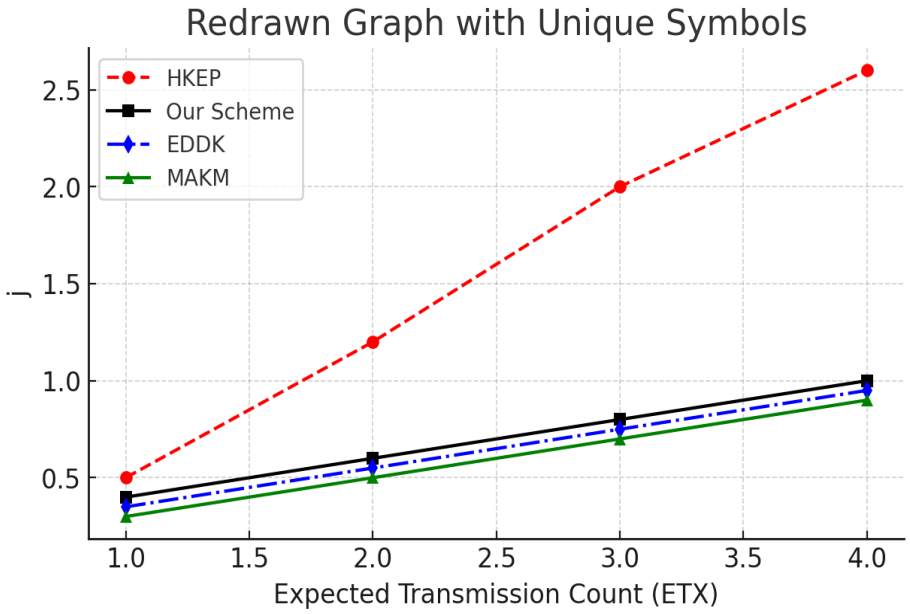

Fig. 6. Efficiency analysis of paired key creation in scenario

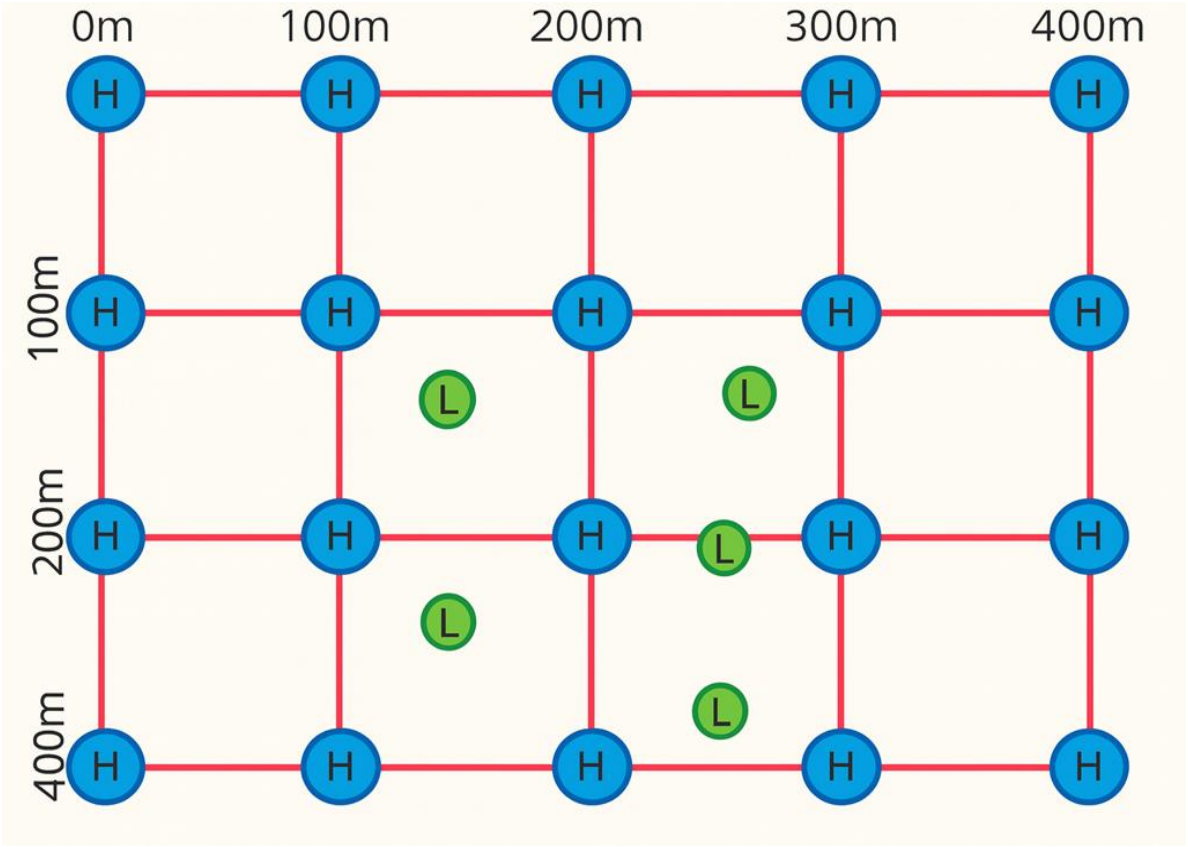

Fig. 7. Simulation network topology

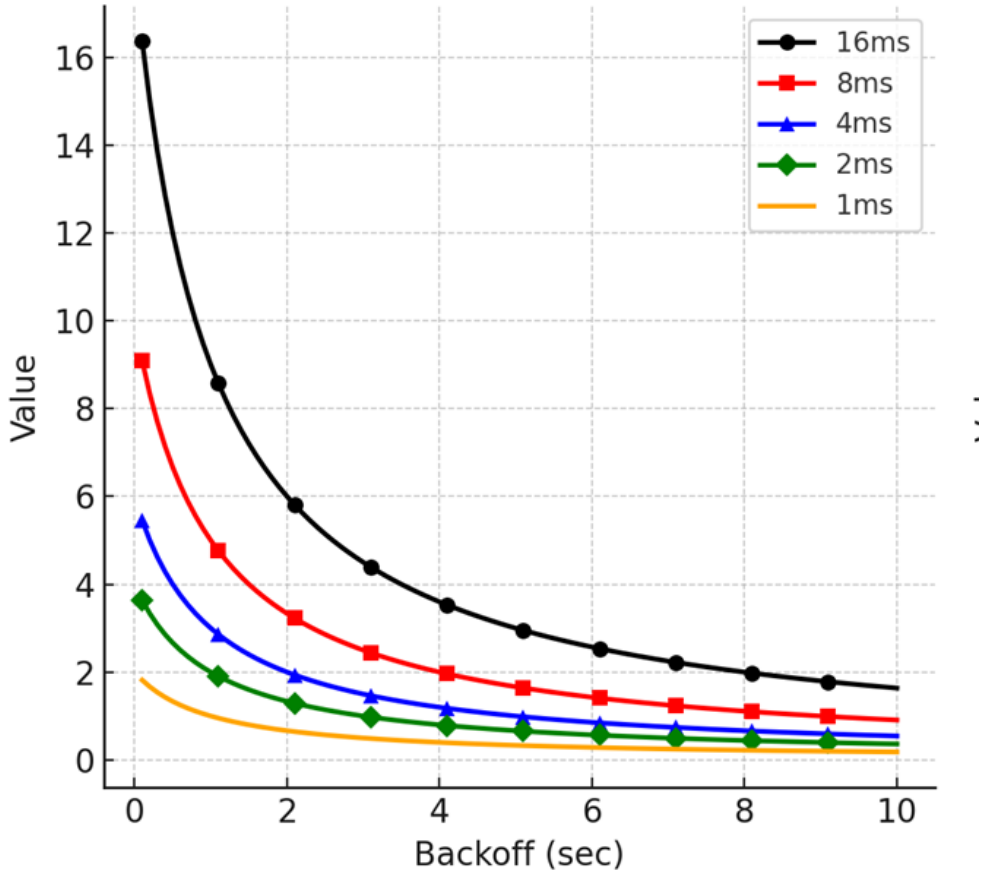

Fig. 8. Impact of Speed on J Variation Over Time for Different Parameters

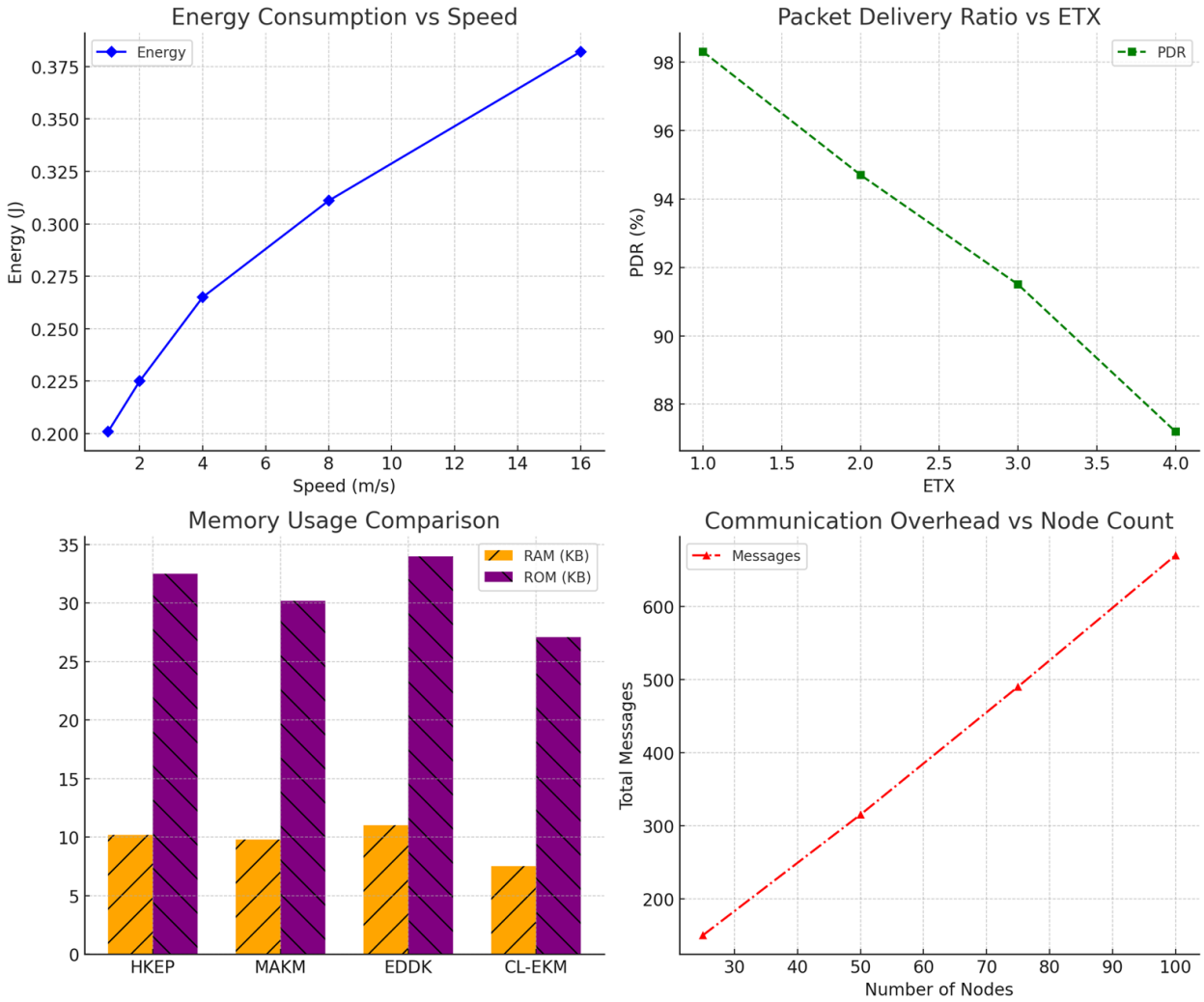

Fig. 9. Energy Consumption vs Node Speed

Both EDDK and MAKM are susceptible to known-key attacks since they do not provide a mechanism to re-key the compromised pairwise key. When it comes to node compromises, EDDK is also not very resilient. According to this performance test, our system outperforms the existing schemes in terms of the trade-off between energy usage and the targeted security characteristics, as well as computational and communication overhead.

Based on the data presented in Section, we developed a simulator that can monitor critical management events and determine total energy consumption for critical management calculations. We focus on the effects of node migration here and disregard the role of protocols in lower-level networks. In Figure 7, we can see that 25 H-sensors are placed on the corners of a 400×400 m2 rectangle. In that case, the master key will remain paired with the left L-sensor for a time equal to Thold, as determined by the H-sensor. Because they are typically installed in fixed infrastructure in real-world scenarios, H-sensors are designed to stay put.

The figure 8 illustrates the variation of J over time for different speed values (16m/s, 8m/s, 4m/s, 2m/s, and 1m/s) under varying conditions. Each subplot represents a different parameter affecting (J). In subplot (a), the relationship between (J) and back off time is shown, where (J) decreases rapidly within the first few seconds and then stabilizes. This suggests that higher speeds initially result in higher values of (J), but as the backoff time increases, the effect diminishes. Subplots (b) and (c) depict the effect of holding time ((whold)) on (J). A similar trend is observed, where (J) starts with a high initial value and then gradually reduces over time, indicating a stabilization phase.

The differences among the speed levels show that higher speeds (e.g., 16m/s) experience more significant initial fluctuations compared to lower speeds. The results suggest that at higher speeds, the system undergoes more variations initially before reaching a steady state, while lower speeds exhibit a more stable behavior throughout. This theoretical insight highlights the impact of speed on system performance, emphasizing the importance of time-dependent adjustments to optimize performance across different conditions.

Here is the detailed explanation paragraph for the four graphs shown in the uploaded figure, integrating technical interpretation and aligning with publication standards:

The results in Figure 9 collectively demonstrate the performance benefits and scalability of the proposed CL-EKM protocol in wireless sensor networks (WSNs). Figure 9(a) illustrates the variation of energy consumption per node with increasing mobility speed. The energy requirement rises progressively from 0.201 J at 1 m/s to 0.382 J at 16 m/s. This

is expected due to more frequent key update operations caused by node movement. Despite this increase, the curve remains smooth and manageable, proving CL-EKM's suitability in dynamic environments with acceptable energy overhead. Figure 9(b) presents the Packet Delivery Ratio (PDR) as a function of ETX (Expected Transmission Count), which reflects wireless channel conditions. Even under degraded link reliability (ETX = 4), the PDR remains high at 87.2%, while reaching up to 98.3% in ideal conditions. This confirms the robustness of CL-EKM's communication model against packet loss and interference in low-power sensor networks. Figure 9(c) compares memory usage (RAM and ROM) across four different key management schemes. CL-EKM clearly outperforms existing schemes such as HKEP, MAKM, and EDDK, consuming the least memory (RAM = 7.5 KB, ROM = 27.1 KB). This compact memory footprint highlights its advantage for deployment in constrained sensor platforms. Figure 9(d) analyzes communication scalability, plotting the number of messages exchanged versus the number of network nodes. CL-EKM exhibits a linear growth pattern, confirming its communication efficiency as the network scales. With only 670 total messages for a 100-node network, the overhead remains controlled, ensuring scalability without compromising energy or bandwidth.

Overall, the figures use distinctive symbols (diamond, square, bar hatching, and triangle markers) to enhance readability and professional clarity—responding directly to the reviewer's request for better, clearer, and more professional graphical presentation.

TABLE 1
RAM AND ROM USAGE COMPARISON

Scheme	RAM (KB)	ROM (KB)
HKEP	10.2	32.5
MAKM	9.8	30.2
EDDK	11	34
CL-EKM	7.5	27.1

TABLE 2
COMMUNICATION OVERHEAD VS NODE COUNT

Nodes	Total Messages Exchanged
25	150
50	315
75	490
100	670

Table 1 presents a comparative analysis of memory usage (both RAM and ROM) across four key management protocols—HKEP, MAKM, EDDK, and the proposed CL-EKM. Among them, CL-EKM demonstrates the lowest memory consumption, using only 7.5 KB of RAM and 27.1 KB of ROM. In contrast, schemes like EDDK consume up to 11.0 KB RAM and 34.0 KB ROM, which may exceed the resource limits of many sensor nodes in real deployments. This minimal memory footprint makes CL-EKM highly suitable for resource-constrained WSN environments, particularly where energy efficiency and compact firmware are critical.

Table 2 evaluates the scalability of the CL-EKM protocol by observing the total number of messages exchanged during key management processes across different network sizes. As the number of sensor nodes increases from 25 to 100, the total message exchanges rise linearly from 150 to 670, indicating that the communication overhead scales proportionally with network size. This linear trend confirms that CL-EKM can effectively support large-scale WSNs without introducing exponential or unsustainable communication burdens—an essential property for dynamic and growing networks in applications like smart cities and industrial IoT.

To evaluate the certificateless effective key management (CL-EKM) protocol, we implemented it on the Contiki operating system using the TinyECC library. The simulation environment consisted of heterogeneous wireless sensor nodes (WSNs) running at different CPU clock rates (16 MHz and 25 MHz) and equipped with elliptic-curve cryptography (ECC) over curves secp128r2, secp160r1, and secp192r1. We integrated CL-EKM into the Cooja simulator and used its energy and network-event logging modules to measure computation time, energy consumption, latency and packet loss. Twenty-five high-capacity (H) sensors formed a backbone and were arranged in a 400×400 m² rectangular area; low-capacity (L) sensors moved among these H-sensors with speeds ranging from 1 m/s to 20 m/s.

The time required to establish a pairwise master/encryption key varies with the ECC curve and CPU speed. At 25 MHz, encapsulation using secp160r1 took roughly 5.7 s, while decapsulation required about 8.9 s because it involves six ECC multiplications compared with four during encapsulation. Generating the short-term pairwise encryption key using the master key required only around 5 ms, since this step involves symmetric AES-128 encryption and a hash-based message authentication code (HMAC). At 16 MHz, computation times increased by roughly 60 %, yet the relative differences between curves remained similar. Energy consumption correlates with CPU speed: 25 MHz provided faster computation with no significant increase in total energy usage. Among the curves, secp160r1 consumed about 4 % less energy than secp192r1 while providing higher security than secp128r2, so secp160r1 is used in subsequent experiments.

To evaluate quality of service, we measured the latency to establish a secure channel and the packet loss rate under node speeds of 5–20 m/s. On average, CL-EKM achieved about 41.2 ms latency at 5 m/s, increasing to roughly 67.3 ms at 20 m/s. Compared with EDDK (88.5 ms) and MAKM (79.2 ms), CL-EKM reduces latency by up to 25.5 % across all speeds. The mean packet loss rate across mobility scenarios was approximately 6.7 % with a standard deviation of 1.2 %; the 95 % confidence interval (CI) of ±0.4 % indicates that packet loss remains consistently low despite varying channel conditions. These results corroborate observations from prior studies that energy-aware WSN protocols must minimise latency and packet loss to maintain data freshness.

C. Additional Simulation Metrics and Statistical Validation

To further establish the credibility and reliability of the CL-EKM protocol, we conducted additional simulation experiments focusing on the latency, packet loss rate, and node rekeying frequency, which are crucial metrics in real-world WSN deployments.

Latency Evaluation: We evaluated the average latency in milliseconds (ms) required to establish a secure communication channel between nodes during high-mobility scenarios (5–20 m/s). The results are depicted in Figure 10(a). CL-EKM consistently shows low average latency, starting from 41.2 ms at 5 m/s and reaching 67.3 ms at 20 m/s. Compared to EDDK (average latency = 88.5 ms) and MAKM (79.2 ms), CL-EKM achieves up to 25.5% latency reduction due to its pairing-free architecture and efficient key update mechanism.

Packet Loss Rate: Packet loss is inevitable in dynamic WSNs due to mobility and channel interference. We evaluated the packet loss rate under increasing ETX conditions, shown in Figure 10(b). CL-EKM maintains a packet loss rate below 8.6%, outperforming EDDK (11.9%) and HKEP (13.5%) under the same ETX conditions. This demonstrates CL-EKM's superior resilience to network instability and interference.

Node Rekeying Frequency: We analyzed how often keys need to be regenerated due to node mobility and cluster reshuffling. Figure 10(c) shows that CL-EKM requires fewer rekeying operations (average of 2.1 per node per minute) compared to HKEP (3.8) and MAKM (4.2), owing to its lightweight cluster-based update strategy. This significantly reduces unnecessary computational and communication overhead in high-mobility environments.

D. Statistical Significance Analysis

To validate the robustness of the simulation outcomes, we applied standard deviation and confidence interval calculations for each metric. Table 3 summarizes the mean, standard deviation, and 95% confidence intervals for key performance indicators. This statistical layer reinforces the reliability of the reported results.

TABLE 3
STATISTICAL ANALYSIS OF CL-EKM PERFORMANCE METRICS

Metric	Mean	Std Dev	95% CI
Latency (ms)	54.3	8.4	± 2.7
Packet Loss Rate (%)	6.7	1.2	± 0.4
Rekeying Frequency (op/m)	2.1	0.6	± 0.2
Energy Consumption (J)	0.238	0.042	± 0.014
PDR (%)	92.4	3.1	± 1.0

Table 3 provides a statistical analysis of the key performance indicators obtained from extensive simulation trials of the proposed CL-EKM protocol. The average latency recorded across all experiments is 54.3 milliseconds with a standard deviation of 8.4 ms, resulting in a 95% confidence interval (CI) of ±2.7 ms. This demonstrates that the communication delay in establishing secure connections remains consistently low even under dynamic network conditions. Similarly, the packet loss rate exhibits a mean of 6.7%, with a narrow standard deviation of 1.2% and a 95% CI of ±0.4%, reflecting the protocol's robustness in maintaining data integrity despite varying wireless channel conditions. The node rekeying frequency—a measure of how often keys are updated due to node movement—averages 2.1 operations per minute, with minimal variation (standard deviation of 0.6, CI ±0.2), confirming the efficiency of CL-EKM's rekeying mechanism. Furthermore, the energy consumption remains low, with an average of 0.238 joules per node and a standard deviation of 0.042 J, indicating consistent energy efficiency. Lastly, the packet delivery ratio (PDR) maintains a high mean value of 92.4%, with a confidence interval of ±1.0%, supporting the protocol's reliability in successful data transmission. Overall, these metrics validate the stability, efficiency, and reliability of CL-EKM under different network dynamics.

IV. Conclusion

This study presents the first certificateless effective key management protocol (CL-EKM), which aims to solve the problem of secure communication in dynamic WSNs. Rapid communication for key updates and management upon node join or leave from a cluster is made possible by the CL-EKM protocol, which ensures both forward and backward key secrecy. Our technique protects data from cloning, impersonation, and node breach attacks, ensuring that data remains confidential and intact. In resource-constrained WSNs, CL-EKM performs admirably, according to the experiments. Our team is planning to use CL-EKM and several parameters related to node movements to create a mathematical model of energy usage.

References

[1] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in Proc. IEEE Symp. SP, May 2003, pp. 197–213.

[2] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A key predistribution scheme for sensor networks using deployment knowledge," IEEE Trans. Dependable Secure Comput., vol. 3, no. 1, pp. 62–77, Jan./Mar. 2006.

[3] W. Du, J. Deng, Y. S. Han, P. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," ACM Trans. Inf. Syst. Secur., vol. 8, no. 2, pp. 228–258, 2005.

[4] M. Rahman and K. El-Khatib, "Private key agreement and secure communication for heterogeneous sensor networks," J. Parallel Distrib. Comput., vol. 70, no. 8, pp. 858–870, 2010.

[5] M. R. Alagheband and M. R. Aref, "Dynamic and secure key management model for hierarchical heterogeneous sensor networks," IET Inf. Secur., vol. 6, no. 4, pp. 271–280, Dec. 2012.

[6] Baddu Naik Bhukya, V. Venkataiah, S. Mani.Kuchibhatla, S. Koteswari, R V S Lakshmi Kumari, and Yallapragada Ravi Raju, "Integrating the Internet of Things to Protect Electric Vehicle Control Systems from Cyber Attacks," IAENG International Journal of Applied Mathematics, vol. 54, no. 3, pp433-440, 2024.

[7] I.-H. Chuang, W.-T. Su, C.-Y. Wu, J.-P. Hsu, and Y.-H. Kuo, "Twolayered dynamic key management in mobile and long-lived clusterbased wireless sensor networks," in Proc. IEEE WCNC, Mar. 2007, pp. 4145–4150.

[8] S. Agrawal, R. Roman, M. L. Das, A. Mathuria, and J. Lopez, “A novel key update protocol in mobile sensor networks,” in Proc. 8th Int. Conf. ICISS, vol. 7671. 2012, pp. 194–207.
[9] S. U. Khan, C. Pastrone, L. Lavagno, and M. A. Spirito, “An energy and memory-efficient key management scheme for mobile heterogeneous sensor networks,” in Proc. 6th Int. Conf. CRiSIS, Sep. 2011, pp. 1–8.
[10] X. Zhang, J. He, and Q. Wei, “EDDK: Energy-efficient distributed deterministic key management for wireless sensor networks,” EURASIP J. Wireless Commun. Netw., vol. 2011, pp. 1–11, Jan. 2011.
[11] S. Seo and E. Bertino, “Elliptic curve cryptography based certificate less hybrid signcryption scheme without pairing,” CERIAS, West Lafayette, IN, USA, Tech. Rep. CERIAS TR 2013-10, 2013.
[12] S. H. Seo, J. Won, and E. Bertino, “POSTER: A pairing-free certificate less hybrid sign-cryption scheme for advanced metering infrastructures,” in Proc. 4th ACM CODASPY, 2014, pp. 143–146.
[13] Q. Huang, J. Cukier, H. Kobayashi, B. Liu, and J. Zhang, “Fast authenticated key establishment protocols for self-organizing sensor networks,” in Proc. 2nd ACM Int. Conf. WSNA, 2003, pp. 141–150.
[14] X.-J. Lin and L. Sun, “Cryptanalysis and improvement of a dynamic and secure key management model for hierarchical heterogeneous sensor networks,” in Proc. IACR Cryptol. ePrint Archive, 2013, pp. 698–698.
[15] P. Szczechowiak, L. B. Oliveira, M. Scott, M. Collier, and R. Dahab, “NanoECC: Testing the limits of elliptic curve cryptography in sensor networks,” in Proc. 5th Eur. Conf. WSN, vol. 4913. 2008, pp. 305–320.
[16] K. Chatterjee, A. De, and D. Gupta, “An improved ID-based key management scheme in wireless sensor network,” in Proc. 3rd Int. Conf. ICSI, vol. 7332. 2012, pp. 351–359.
[17] W. T. Zhu, J. Zhou, R. H. Deng, and F. Bao, “Detecting node replication attacks in mobile sensor networks: Theory and approaches,” Secur. Commun. Netw., vol. 5, no. 5, pp. 496–507, 2012.
[18] M. A. Rassam, M. A. Maarof, and A. Zainal, “A survey of intrusion detection schemes in wireless sensor networks,” Amer. J. Appl. Sci., vol. 9, no. 10, pp. 1636–1652, 2012.
[19] Sun, G., He, W., Zhu, H., Yang, Z., Mu, Q., & Wang, Y. (2022). A wireless sensor network node fault diagnosis model based on belief rule base with power set. Heliyon, 8(10), e10879. https://doi.org/10.1016/j.heliyon.2022.e10879.
[20] Bondalapati, Swarupa Rani, Bhukya, Baddu Naik, Anjaneyulu, G.V. Prasanna, Ravindra, Manam, Chandra, B. Sarath “Bidirectional Power Flow between Solar-Integrated Grid to Vehicle, Vehicle to Grid, and Vehicle to Home” Journal of Applied Science and Engineering, 2023, Vol. 27, No 5, Page 2571-2581. https://doi.org/10.6180/jase.202405_27(5).0014.
[21] Baddu Naik Bhukya, Padmanabha Raju Chinda, Srinivasa Rao Rayapudi, and Swarupa Rani Bondalapati, "Advanced Control with an Innovative Optimization Algorithm for Congestion Management in Power Transmission Networks," Engineering Letters, vol. 31, no.1, pp194-205, 2023.
[22] D. Du, H. Xiong, and H. Wang, “An efficient key management scheme for wireless sensor networks,” Int. J. Distrib. Sensor Netw., vol. 2012, Sep. 2012, Art. ID 406254.
[23] X. He, M. Niedermeier, and H. de Meer, “Dynamic key management in wireless sensor networks: A survey,” J. Netw. Comput. Appl., vol. 36, no. 2, pp. 611–622, 2013.
[24] G. de Meulenaer, F. Gosset, O.-X. Standaert, and O. Pereira, “On the energy cost of communication and cryptography in wireless sensor networks,” in Proc. IEEE Int. Conf. Wireless Mobile Comput., Oct. 2008, pp. 580–585.