International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Lightweight Secure Group Communication Methodology for RPL Based IoT Networks

Bandarupalli Rakesh[1], Dr. H. Parveen Sultana*[2]

Submitted: 10/12/2023 **Revised:** 21/01/2024 **Accepted:** 31/01/2024

Abstract: The rapid development of wireless sensor networks (WSNs) has made them becoming increasingly popular as a possible significant application. Every single time, this acknowledgment takes place. Internet of Things (IoT) networks place importance on security as data travels over an inflow of vulnerable devices. Information integrity preservation, consumption of energy optimization, and data security protection are therefore the most critical requirements for Internet of Things networks. A Group Key (GK) is a shared symmetric key that all members of a group use to encrypt their communications and keep them secret. Whenever a group member enters or leaves, the GK must be redistributed to ensure forward and backward confidentiality. Key management processes, such as generation and distribution, add load to the system when limited network resources are used. The present research uses the Advanced Encryption Standard (AES) to help with the development of a lightweight and secure Group Communication (GC) method. Protecting the data's integrity is the ultimate responsibility of the Public Key Infrastructure (PKI) integration. Protecting the network's GCs from attacks, reducing bandwidth consumption, and minimizing network overhead due to key redistribution and management procedures are all objectives of the AES-PKI algorithm. Additionally, by using both backward and forward secrecy during key distribution, the current AES-PKI technique provides data confidentiality. There is proof that this approach works well against network mobility in scalable IoT networks. Using dynamic simulation data that incorporates the ratio of received packets to overall energy usage, AES-PKI's performance is validated.

Keywords: *Security, Internet of Things (IoT), Group Communication, Energy Efficiency, Public Key Infrastructure (PKI), Integrity of Data.*

1. Introduction

Connecting physical objects through the web is the foundation of the Internet of Things (IoT) concept, which has the ability to build a global network. Internet of Things (IoT) technologies allow for the intelligent connection of sensor smart things [1]. These technologies include embedded systems, item identification, improvements in Wireless Sensor Networks (WSN), and nanotechnology. WSN is considered a vital communication component in real-time applications like smart cities, healthcare, transportation, and military contexts [2], and these WSN elements are integral to IoT networks. Devices connected to wireless sensor networks (WSNs) in the Low Power and Lossy Network (LLN) area of the network have limited access to storage, energy, and bandwidth, among other resources [3,4].

In order to connect devices in environments with environmental constraints, researchers have created the Routing Protocol for Local Light Networks (RPL). A low-power personal area network (PAN) based on the Internet Protocol (IP) can be established through the use of 6LoWPAN technology. It is possible to intercept, physically destroy, capture, or simulate sensor equipment via wireless comm unication [5-7]. The presence of uncontrolled operative situations and connectivity failures in wireless communication makes it susceptible to this vulnerability. Data integrity, confidentiality, and node authentication are among the security issues that researchers are focusing on, according to [8,9]. When it comes to wireless sensor networks (WSN), security is paramount in a lot of critical applications.

It is the responsibility of the network administrator to keep all communications private, ensure that all devices can be authenticated, and enable secure Group Communication (GC) amongst all members of the group [10]. If the Internet of Things (IoT) is to accomplish its goals, this must be in place. The confidentiality of group data depends on preventing unauthorized decryption of messages. The term "integrity" refers to the steps taken to guarantee that data is correct, reliable, and accurate before, during, and after transmission [11,12]. Because only approved devices can join a certain group or network, device authentication is crucial. Researchers have developed a wide variety of techniques for controlling encryption keys to ensure the security of WSNs. These approaches aim to tackle the limited processing power and energy resources of sensor nodes. Static key encryption is the foundation of these technologies. However, there are several issues with these methods, including the extensive memory space needed to store shared pairwise keys and the high communication cost [13–17].

[1] *Research Scholar, SCOPE, VIT, Vellore – 632014, INDIA*
[2] *Professor, SCOPE, VIT, Vellore – 632014, INDIA*
ORCID ID : 0000-0001-9108-5043
** Corresponding Author Email: hparveensultana@vit.ac.in*

The present research's objective is to solve these problems by creating an Internet of Things (IoT) network routing system that is both lightweight and secure. To ensure the data maintains secure over the entire process, the Advanced Encryption Standard (AES) is used for both data encryption and decryption. Data authentication in the Internet of Things (IoT) is made possible, however, by using Public Key Infrastructure (PKI) to ensure data integrity. This solution is prepared, flexible, and scalable. Cooja, the Contiki network simulator environment, is being used to test out an implementation of the proof of concept. Based on the outcomes of the performance validation trials, the proposed method uses randomly generated identities for every device, providing a better and more robust authentication process. Using a standard public key infrastructure (PKI) with a single entity standing in for identification, the method employs standard cryptography and authentication techniques.

The following is the summary of the paper: The second part provides an overview of current IoT methods for WSN, while the third part looks into the proposed method to address the drawbacks of these methods. The experiments that were carried out on the Cooja Simulator to validate the method are detailed in this section. The study is concluded and a summary of future prospects is provided in Section 5.

2. Literature Survey

A secure Multi-agent Reinforcement Learning (MRL) system was created by A. P. Renold and A. B. Ganesh [18] using convex nodes as mobile nodes. All the while, they were thinking about how to send legitimate messages. The mobile sink's data transmission was determined by means of convex nodes that were formed by means of the energy-aware convex hull method.In order to authenticate messages and guarantee the safety of data transmission, the ElGamal system used elliptic curve cryptography. By utilising a Support Vector Machine, or SVM, the presence of the malicious node might be detected within the network. Contiki conducted an experimental validation using the Cooja simulator to test the system's performance with different amounts of nodes interacting with static and wireless sources. It’s extremely high energy consumption is a result of the node's ability to change states quickly enough because it missed sufficient neighbors.

To develop a reliable and comprehensive solution for a society with high social intelligence, M. Babar and colleagues [19] collaborated. This solution included the development of a reliable and safe engine for DSM with multiple layers. Utilizing the IIoT, this engine is designed to satisfy severe business specifications. A payload-based authentication mechanism was employed to enhance the security of this DSM engine, as described in the lightweight handshake approach. I used the MapReduce parallel processing platform to enable DSM to process the data streams. Experimental results showed that DSM reduced connection overhead, reaction time, and memory usage. The DSM posed a significant admissions hurdle when compared to earlier damaging assaults. While the Kalman Filter was able to remove noisy data and improve the findings, it did not account for partial data that had missing values.

Datagram Transport Layer Security (DTLS) was designed and put into use by P. M. Kumar and U. D. Gandhi [20] to solve the reliability problems. In contrast, the DTLS protocol ran into difficult issues, such as the possibility of a DOS attack on the server.As a result, the authorization and authentication mechanism based on smart gateways protected the health data from these kinds of risks. Through the use of IoT wearable devices, this technology made it possible to gather health data. The purpose of this study was to develop an improved DTLS that uses smart gateways using a Contiki-based network simulator. To ensure that the improved DTLS protocol worked properly, we measured the time it took to complete the handshake and data transmission. Finding the optimal mobile nodes is a challenge for the Contiki simulator, which depends on a security algorithm.

According to A. Anand and colleagues, the GC was protected when the Topology Adaptive Re-kEying (TARE) method, which is flexible was put into implementation [21]. Network power consumption was reduced with the use of TARE's routing topology, distribution, and key derivation algorithms. Completely without compromising data or key confidentiality, this was achieved. We found that the TARE significantly reduced overhead for networks compared to network mobility. Results demonstrated that TARE outperformed challenge systems in terms of energy consumption, bandwidth utilization, and the number of encrypted message transfers that occurred during re-keying procedures. Several layers of energy usage and mapping tree repair are impacted as a result of the routing protocol (i.e., DODAG) not considering node revocations.

Researchers A. Bahramlou and R. Javidan found that using the A-RPL temporal routing protocol for LLN in conjunction with the M-RPL distributed computing technique optimised both control plane traffic and congestion [22]. The reduced communication overhead and enhanced network performance were both brought about by A-RPL. When compared to RPL and M-RPL, A-RPL resulted in a decrease in the total number of Directed Information Object (DIO) messages transmitted. We reduced energy consumption, packet loss percentage, and routing overhead overall by testing these techniques on the Contiki OS and the Cooja emulator. The decrease in data packets and the transmission rate adjustment by DIO in response to topology changes were a result of this.

3. Proposed Methodology

When faced with an adversarial wireless network, ensuring the safe transfer of data between nodes is essential for maintaining security. However, the inherent vulnerabilities of networks, particularly in multi-hop Wireless Sensor Networks (WSN), make them susceptible to various attacks. Routing misbehavior stands out as a particularly destructive threat, involving the arbitrary dropping of packets, which undermines effective communication within the network [23]. Through the use of collaborative simulation, nodes are able to reduce the frequency of routing misbehaviour and ensure trusted communications.

The RPL protocol, which stands for Routing Protocol for Low Power and Lossy Networks, includes multiple cryptographic techniques [24]. To stop malicious nodes from getting private information, these methods were developed. However, managing cryptographic keys becomes a challenge due to the high resource consumption in Wireless Sensor Networks (WSNs). This includes computational power, memory, poor bandwidth, and battery capacity. Protecting both data and cryptographic keys stored in unattended sensors becomes a challenging task due to these resource constraints [25].

This research presents a lightweight routing solution for secure Group Communications (GCs) in IoT networks. The questions that have been presented so far will be solved using this method. A summary of the system model and the related operational processes for the proposed method are given in the next section.

3.1. System Model

With their incorporated actuators, sensors, and slowly resourceful Group Communication (GC) architecture, IoT networks are perfect for the proposed method. The General Counsel is responsible for providing the necessary resources for internal communication. The created approach is compatible with a range of routing protocols since it is adaptable and can accommodate hierarchical topologies. Here and in the following research, RPL in Mode of Operation 3 (MOP3) acts as the fundamental routing protocol.

The sole protocol that has been standardized for usage in IoT networks is RPL, thanks to its ability to handle massive volumes of data. An IoT network's three critical layers are the application, transport, and network sublayers. Conversely, there are no network-layer security mechanisms in the current RPL. The RPL protocol has recently included the Advanced Encryption Standard (AES) algorithm to encrypt data objects and provide an extra degree of security, in order to solve this problem. To further ensure the creation of secure RPL messages, the cryptographic algorithm known as Public Key Infrastructure (PKI) is employed. PKI provides authentication and confidentiality for routing communications. The incorporation of safe routing at the network layer is shown in Figure 1.

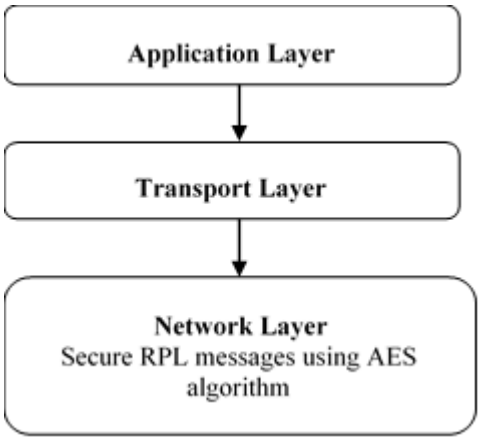

Fig. 1. Various Internet of Things (IoT) Layers for Routing Protocol Security

Reliable Power Line (RPL) is an essential part of low-power wireless networks that allows topologies similar to ad hoc network trees to be formed. In order for a configurable Objective Function (OF) to decide which parent node to select, every node maintains a parent that is only one hop away. Multiple traffic models, including unicast, multicast, and multipoint-to-point communication, are fully supported by RPL. As a receiver in this form of traffic, the root receives data in an upward direction from all the other nodes. Thanks to RPL, data may be transferred faster up the tree since nodes are simply a component of the topology in relation to the information supplied by the nearest neighbors.

A few limitations are there in the RPL graph, however. For messages to be processed, nodes closer to the root must be located in the tree's upper reaches. The result is a heightened rate of energy loss in nodes around the root. The following are examples of key RPL concepts

3.1.1 Direction-Oriented Directed Acyclic Graph (DODAG)

Furthermore, a tree-like graph with a single root node often acts as a border router and has outward-moving cycles or edges.

3.1.2 DODAG Information Solicitation (DIS)

To get RPL DAG information, nodes use ICMPv6 messages to communicate with their nearest neighbors. The data is subsequently encrypted using industry-standard static key algorithms including Blowfish, DES, and AES. Once the data has been encrypted, these methods are applied. The creation of Message Authentication Codes (MACs) follows.

3.1.3 DODAG Information Object (DIO)

The MAC address verification process involves receiving communications from DIS via ICMPv6.

3.1.4 Destination Advertisement Object (DAO)

Depending on the RPL mode, new nodes will connect with either their parents or the root to reach a destination inside the DAG. They will be able to reach their objective with this.

3.1.5 Storing Mode of RPL

Certain nodes maintain a record of routing information that is specific to each node's Sub-DODAG.

3.1.6 Non-Storing (NS) Mode of RPL

Because nodes can only learn about their parents, the complete DODAG must be kept in a routing table that is maintained by the roots. The proposed solution uses the Advanced Encryption Standard (AES) technology to secure the data, as will be shown in the following section.

3.2. Proposed Methodology Algorithm

The main objective of the research is to ensure the secure exchange of information inside a group setting. This becomes possible when messages are encrypted and decrypted using the Advanced Encryption Standard (AES) algorithm. To further guarantee the data's authenticity and integrity, Public Key Infrastructure (PKI) is employed. The following document explains the operational strategy of the proposed method.

Figure 2 illustrates the process of the AES-PKI Algorithm, which comprises three key phases: Authentication Phase, Group Key Sharing Phase, and Transmission Phase. Before the authentication phase, the nodes are registered at the root node. During the registration phase, the nodes that want to join the group will send their details using a DIS message to the root node. The child node will send the MAC address of the node to the root node. The root node will generate the S. No. using the MAC address and registration time and share it with the child node. Equation (1) illustrates the S. No. generation process.

$$S.\,No. \rightarrow MAC \oplus R.\,T. \qquad (1)$$

3.2.1 Authentication Phase

Authentication of the child nodes takes place during this phase of DODAG building. The child nodes are authenticated by the root node. Authentication of the node is done by using the MAC address of the child node and S. No. This step ensures the integrity and security of the Directed Acyclic Graph (DODAG) structure.

3.2.2 Group Key Sharing Phase:

The sink node initiates the sharing of the group key with the child nodes. This phase involves distributing the group key to ensure secure communication within the group.

3.2.2 Transmission Phase

During the transmission phase, data is encrypted using the AES algorithm. The encrypted data is then sent to the sink node, ensuring confidentiality and security during the data transmission process.

The AES-PKI Algorithm uses these three steps to provide a thorough method for ensuring group members' secure communication in IoT networks.

3.2.3 WSN Group Communication

Taking into account both the group's composition and the DODAG network topology established by RPL, the Group Communication (GC) entity is a memory-rich, robust entity. The central node of a directed acyclic graph (DAG) stores data about every other node in the network. We call this particular node the root node. With the exception of the root, every node in this architecture models a parent-child relationship. The outcome of RPL is a directed acyclic graph (DODAG) empty of cycles, with each node taking the shortest and most efficient path possible.

This approach was devised under the assumption that all nodes will receive messages without any delays or problems. Prior to getting into the AES-PKI system paradigm, this section will provide an overview of the necessary security requirements.

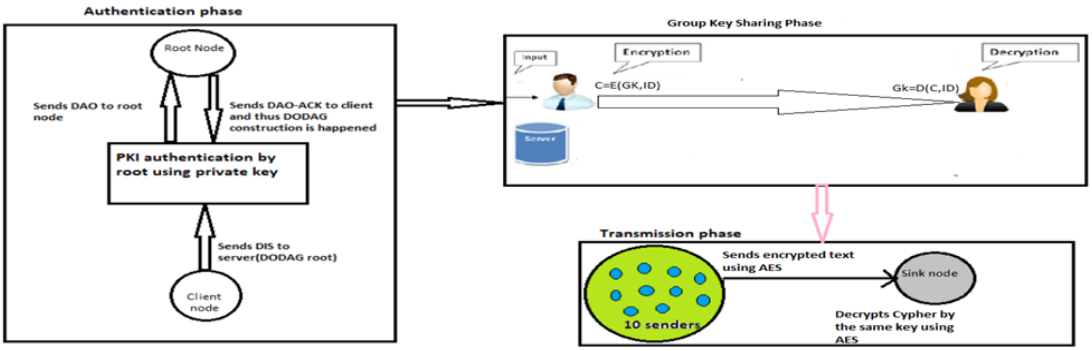

Fig. 2. Proposed Methodology Architecture for AES-PKI

3.2.4 Creating Multicast Groups

The formation of multicast groups and the transmission of messages are both supported by the whole group. It is GC's responsibility to perform these processes, which involve key generation and re-keying. Members of this group have some connection to GC.

3.2.5 Cryptographic Scheme Support

In a network where all nodes can execute typical cryptographic suites, message encryption and decryption are both feasible. Since the proposed method takes use of Group Keys (GK) to encrypt data at the group level, it can ensure that data included in a multicast group remains confidential. While every authorised member of the multicast group can decode messages no outside parties or malicious actors can do so.

3.2.5.1 AES Algorithm

AES, a significant algorithm in the cryptographic field, encrypts and decrypts sensitive information for data security. AES allows three operational key lengths, and in this study, a key length of 128 bits is used, represented by a 4x4 matrix. While the standard AES algorithm involves ten rounds for a 128-bit key length, this study increases the number of rounds to 12. A cypher state performs four basic operations—bytes subtraction, column mixing, round key addition, and row shifting—in every round.

The AES algorithm repeats these operations for ten rounds (except for the last round, where Mix Columns is not executed) based on the key length. The algorithm is outlined as follows:

The provided code snippet outlines the structure of the AES algorithm. Here's a breakdown of the key steps involved in each round:

Initialization:

- S = M: Initialize the state array S with the input data M.
- AddRoundKey(S, &w[0]): Perform the initial round key addition.

Rounds Loop:

- for i = 1 to n: Iterate through the rounds (where n is the number of rounds).
  - SubBytes(S): Substitute each byte in the state using a substitution table.
  - ShiftRows(S): Shift the rows of the state array.
  - MixColumns(S): Mix the columns of the state array.
  - AddRoundKey(S, &w[i*4]): Add the round key for the current round.

Final Round:

- After completing the specified number of rounds, perform the following operations for the final round:
  - SubBytes(S): Substitute bytes.
  - ShiftRows(S): Shift rows.
  - AddRoundKey(S, &w[40]): Add the final round key.

Given its massive data processing power, the Advanced Encryption Standard (AES) algorithm stands out among encryption algorithms. Consequently, AES with 128-bit keys is used for data encryption and decryption in this work.

3.2.6 Authentication and Data Integrity

Making sure that messages in a multicast group are authentic and free of manipulation attempts is of the utmost significance when they are being exchanged. Researchers in this study solved the problem of data integrity by applying PKI techniques. The client must always use the server's public key to encrypt their name, MAC address, serial number, and a randomly generated key (K'). The server can generate a random identifier using the media access control (MAC) device's address in addition to the serial number. Sending the encrypted data back to the client after encrypting it with key (K) shows the validity of the client's identification thanks to the timestamp.

Each client's name, MAC address, serial number, identity, and timestamp value are saved on the server along with all the other relevant data. A new identity will be assigned to the client by the server once the timestamp has expired. If the two clients want to communicate, the first client will ask the server for the second client's identity. Not only is the identity of the client being requested given in this request, but the name of the client making the request is also included. By utilising the identity of the initial client, this request is encrypted. To prevent unauthorised access, the server encrypts the client's identity together with the requested date when the client uses their identity to request the service.

Following this initial conversation, the first client will use the second client's identity to transmit their message, encrypted timestamp, and identification. By continuing to use each other's identities in future communications, we can keep track of which clients have sent which messages and when in a database table. The client node does the actual identification, and the server encrypts it with a randomly generated key ('K') before sending it back. Upon first connection, an end device uses the server's public key to encrypt its connection to the server.

Among the many factors considered essential when developing the server-side detection method are the serial number and the MAC address. When creating a new identity, the "K" represents the end device's supplied key for encrypting the produced identity. Clients and servers alike use the Advanced Encryption Standard (AES) technique for data encryption and decryption. Finally, a timestamp-shaped error is used to determine the integrity of the identity. When one identity expires, the server will create a new one and send it to the client that matches it. What follows is an explanation of the algorithms used to produce random IDs and numbers.

The provided algorithm outlines the authentication of a node and the sharing of a Group Key. Here's a breakdown of the

algorithm:

Algorithm for Group Key Sharing and Authentication of Node:

1. Start
2. C = E(MAC||S.No.||K, PK): Scan the serial number, and media access control (MAC) address, together with an arbitrary key K generated from the server's public key (PK) before transmitting it to the server for encryption.
3. MAC||S.No.||K = D(C, PRk): The MAC address, serial number, and key K can be obtained by decrypting the ciphertext (C) using the server's private key (PRk).
4. By combining the MAC address with the serial number, the server may establish an entirely random identity.
5. C1 = E(ID||T.S, K): Encrypt the concatenation of the generated identity (ID) and a timestamp (T.S.) for validity using key K.
6. Every client's identity, MAC address, and serial number are recorded by the server, together with the timestamp value that belongs to that client.
7. The client will be provided with a new identity by the server once the date has expired.
8. C2 = E(GK, ID): Encrypt the Group Key (GK) using the client's identity (ID).
9. GK = D(C2, ID): Decrypt the received ciphertext (C2) using the client's identity to obtain the Group Key (GK).
10. Stop

Explanation:

- In steps 2 and 3, the emphasis is on encrypting information for the purpose of client-server authentication and decrypting it, respectively. If the server wants to make sure the data is legitimate, it will decrypt the client's data.
- Step four involves the server creating a unique client ID from the client's MAC address and serial number.
- Steps 5 and 6 involve encrypting the generated identity along with a timestamp for validity, which is stored by the server.
- It is the server's responsibility to ensure that the client receives a fresh identity once the associated timestamp expires. The seventh stage is here.
- Steps 8 and 9 involve the encryption and decryption of the Group Key for secure communication within a multicast group.

Taking the serial number and media access control (MAC) address as inputs, the above technique explains how to generate an identity. Here is a rundown of the algorithm:

Algorithm for generate Identity:

1. Start
2. Set mLength = strlen(MAC.c_str())
3. Set sLength = strlen(serial.c_str())
4. If mLength == sLength then
    4.1. Set l, bl = mLength
5. Otherwise, if mLength<sLength then
    5.1. Set l = mLength
    5.2. Set bl = sLength
6. Set l = sLength
7. Copy MAC.c_str() to mBuf
8. Copy serial.c_str() to sBuf
9. Initialize x to 0
10. While x!= 16:
    10.1. Set k = generateRand(0,12)
    10.2. Set y = mBuf[k] + sBuf[k]
    10.3. While y > 127 or y < 32:
        10.3.1. Set y = rand() % 100
    10.4. End while
    10.5. iBuf[i++] = y
11. End while
12. Return iBuf
13. Stop

Explanation:

- The procedure discovers the base length (bl) and the minimum length (l) in addition to the length of the serial number and MAC address.
- This procedure then inserts the MAC address and serial number into the required buffers (mBuf and sBuf).
- A loop is initiated to generate a random identity (iBuf) of length 16.
- Within the loop, a random index k is generated, and the corresponding elements from mBuf and sBuf are summed up (y).
- If y falls outside the ASCII range [32, 127], it is recalculated until it meets the criteria.
- The calculated value y is stored in iBuf.
- The loop continues until the desired identity length of 16 is achieved
- The final iBuf is returned as the generated identity.

Algorithm for generateRand:

1. Start

2. Call srand(time(NULL)): Set the current time as the initial value for the random number generator.
3. Return min + rand() % (max - min): Produce a random integer between [min, max).
4. Stop

Explanation:

- srand(time(NULL)): Every time you run the random number generator, it will have a different seed because it is seeded with the current time.
- rand() % (max - min): Generates a random number within the range [0, max - min).
- min + rand() % (max - min): Shifts and scales the generated random number to fall within the desired range [min, max).

Only packets encrypted with the corresponding device's identity can be decrypted by that device during system operation. This method decreases the amount of memory needed for data storage, especially at the central server node, and it also preserves the device's legitimacy. The design of the system enhances its protection against replay and man-in-the-middle attacks. The effectiveness of the described method will be validated by discussing an experimental evaluation of the proposed method in the next section.

4. Experimental Results and Discussion

4.1. Experimental Setup

We will validate the proposed method compared to existing methodologies using the experimental setup described below:

4.1.1 Contiki Operating System:

It is possible to simulate the proposed method by implementing the use of the Contiki OS. When it comes to developing applications, Contiki has you covered for both WSNs and the IoT. It is a real-time OS that facilitates the development and testing of software.

4.1.2 Cooja Emulator:

Before releasing their code to the public on hardware, researchers can analyze it and test its functionality using Contiki's built-in emulator, Cooja.

4.1.3 Simulation Environment:

The experiments are simulated using Cooja Contiki 2.7 with the following hardware specifications:

- 16GB Memory RAM
- 1TB Hard disk
- i7 processor
- 8GB GPU

4.1.4 Node Configuration:

There are 25, 50, and 100 nodes in the simulation, and each of them works for half an hour.

4.1.5 Performance Metrics:

To find out how effective the created AES-PKI method is the following metrics are considered:

- Packets Delivery Ratio (PDR)
- Energy consumption of CPU

4.1.6 Validation Process:

In order to validate the system, we will perform a series of simulations with varying numbers of nodes and look at their PDR and energy consumption after 30 minutes.

The proposed AES-PKI method will be tested in this experimental setting to see how well it performs under varying network sizes with respect to packet reception and energy usage. You can learn a lot about how efficient and successful your strategy is going to be from the findings of this review.

Parameters	**Values**
OS	Contiki OS 2.7
Area	400m*400m
MAC Layer	IEEE 802.15.4
Duty Cycle	Null RDC
Network protocol	Contiki RPL
Objective Function	MRHOF
Application program	Examples/ipv6/rpl-collect
Mote Type	Sky Mote
Nodes Layout	Linear
Number of nodes	300
Radio Mediums Model	Unit Disk Graph Medium(UDGM):Distance Loss
Ranges of Nodes	Rx and Tx: 100m

Table 1: Simulation parameters

Table 1 illustrates the Contiki environment's simulation parameters. Several nodes' PDR performance have been examined in depth in the section that comes.

4.2. Comparison of AES-PKI's Performance with PDR

In order to evaluate the proposed AES-PKI method's performance, Packet Delivery Ratio (PDR) is used. All conditions, including and excluding those involving Public

Key Infrastructure (PKI), are considered in this analysis. A grand total of 300 packets were used for validation, and the results may be found in Table 2. Furthermore, Figure 3 provides visual representations.

Table 2: Ratio of Received Packets Values in Simulation

No. Of Nodes	Without PKI	With PKI
20	96.15	97.9
40	92.34	95.4
60	88.4	92.3
80	85.6	89.5
100	82.3	87.6
120	78.5	85.4
160	70.6	77.3
180	65.4	74.5
200	62.3	72.5
240	56.4	71.5
260	54.2	70.6
300	50.8	69.6

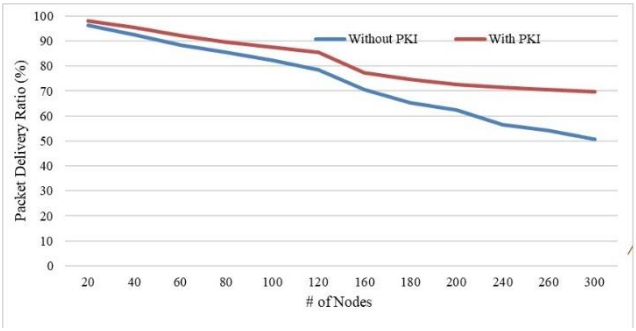

Fig. 3. Using PKI for Performance Analysis of the Proposed Method

The table and graphical representations offer insights into how the proposed AES-PKI method performs in terms of Packet Reception Ratio under different scenarios. The comparison between scenarios with and without PKI provides a comprehensive view of the impact of Public Key Infrastructure on the network's ability to receive packets effectively. Further analysis and discussion will be provided based on these results.

Results from the simulation study show that the PDR performed better when the proposed Public Key Infrastructure (PKI) was used. This is demonstrated by Table 2 and Figure 3. As compared to the approach without PKI, the proposed AES-PKI solution produced better PDR results. Enabled by less data loss due to better data authentication, this came to occur.

By comparing the proposed AES-PKI algorithm to existing methods, the effectiveness of the previous approach is evaluated in further detail [27]. Figure 4 depicts a comparison between the AES-PKI and the currently used techniques in terms of PDR. Compared to the SEC-MRL and SEC-TMP techniques, the proposed approach provided better performance. In a scenario with 25 nodes, AES-PKI algorithm achieved 96.5% PDR, where as SEC-MRL and SEC-TMP achieved 95 % and 95.8% . With 50 nodes, the proposed AES-PKI produced a PDR of 92.2%, where as SEC-TMP and SEC-MRL achieved 80.28% and 85.5%. While with 100 nodes AES-PKI obtained a PDR of 87.6% where as Sec-TMP and SEC-MRL achieved 75.6% and 76.89%.

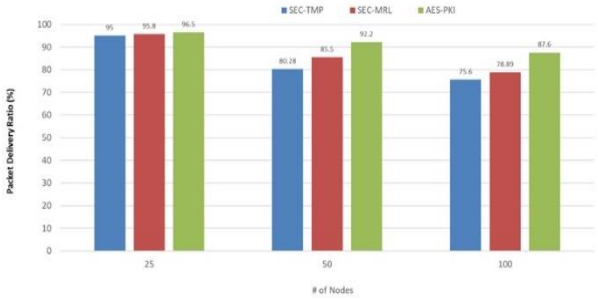

Fig. 4. Efficiency Evaluation of Packet Delivery Ratios

The next section will cover the validation of the node's sending and receiving features, as well as the total energy consumption of the CPU and LPM.

4.3. Energy Consumption-Based Performance Evaluation

Find out how much power each node uses by adding the numbers into the equation (2) shown below.

$$E = CPU_{energy} + LPM_{energy} + TX_{energy} + RX_{energy} \quad (2)$$

The CPU energy consumption is represented by the symbol CPU_{energy} , whereas "transmitted messages by node proportion" (TX) denotes the energy consumption of sent messages TX_{energy}, "low power mode" (LPM) denotes the energy consumption overall LPM_{energy}. At last, the term "received messages" (RX) describes the amount of power required to process received messages RX_{energy} . To get the CPU cycles and the radio cycles, you can use the power trace module of Cooja. In this study, kilowatts (or kW) are the units of measurement for energy. This simulation is run for several cycles using Equation (2), and the values that are simulated are described in detail in the sections that follow.

4.3.1. Distribution of Nodes Average Energy Consumption

An estimate of the average quantity of energy consumed can

be obtained by applying the following formula:

$$Ea = \frac{\sum_{i=0}^{n} E}{n} \qquad (3)$$

E is the amount of energy consumed by one node, and n is the total number of nodes used, in the equation that was previously demonstrated. Ea is the mean daily energy consumption. Figure 5 displays the average energy consumption of different node types while using the currently used methodologies. At a node count of 25, the proposed AES-PKI techniques used half a milliwatt of electricity. Power consumption was 0.6 mW when 50 nodes were present, and 1.3 mW when 100 nodes were used. We used the symmetric encryption algorithm AES to generate group keys, which means that the proposed methodology used less energy.

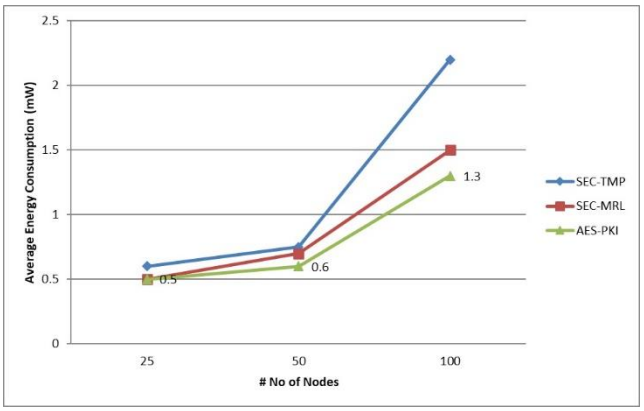

Fig. 5. Average energy use compared

To determine how much power the nodes in the GC communication network are using, we add up the individual total energies as we send each message. Based on these simulation results, the proposed AES-PKI enables key management to provide GC in LLN (the Internet of Things). Explanation being, LLNs are resource constrained since nodes in them are always moving, resulting in a dynamic topology. In order to address the problems that came up while using the RPL routing protocol to solve constrained networks, this study published an AES-PKI approach. It's an efficient and effective methodology. For this reason, the vast majority of real-world Internet of Things application situations employ the RPL as the routing protocol for AES-PKI.

5. Conclusion

Protecting connections and reducing power consumption are two of the primary problems with the Internet of Things (IoT), particularly when it comes to detecting incorrect measurements from sensors. As a result, resolving these problems must prioritise energy efficiency and data integrity. Observed data can be protected with cryptography, which appears to be an effective method. Low-Power and Lost Networks (LLNs) like the Internet of Things (IoT) present problems for important management processes because their topologies and resource requirements are always unpredictable. Given this, it is logical that any useful management strategy should aim at reducing the load on the power grid resulting from operational signals. As part of this research, we look at how to configure AES-PKI to provide secure Group Communication (GC) inside network architecture. In order to decrease the memory consumption, AES-128 bit symmetric key encryption and decryption makes use of a 16-byte identity as the key instead of a public key that is 272 bytes in size. Those end devices in the IoT ecosystem with limited memory and the central server node will benefit the most from this reduction. Improved defence against replay and man-in-the-middle attacks is possible because to AES-PKI's architecture. Unexpectedly it manages to do this while maintaining the privacy of the data or the keys. With its focus on improving the existing routing architecture for key distribution and derivation methods, the AES-PKI design also helps reduce network energy usage. Focusing towards future research, this study proposes the research and implementation of efficient methods designed for mobile node applications. Further research along this line would probably render the proposed methodology more robust and applicable to new Internet of Things (IoT) applications.

References

[1] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, (2010). “Vision and challenges for realising the Internet of Things”. Cluster of European Research Projects on the Internet of Things, European Commision, 3(3), 34-36.

[2] Sran, Sukhwinder Singh, Jagpreet Singh, and Lakhwinder Kaur. "Structure Free Aggregation in Duty Cycle Sensor Networks for Delay Sensitive Applications." IEEE Transactions on Green Communications and Networking 2.4 (2018): 1140-1149.

[3] Hassan, A., Alshomrani, S., Altalhi, A., & Ahsan, S. (2016). “Improved routing metrics for energy constrained interconnected devices in low-power and lossy networks”. Journal of communications and networks, 18(3), 327-332.

[4] Ranjan, Rajeev, and Shirshu Varma. "Challenges and implementation on cross layer design for wireless sensor networks." Wireless personal communications 86.2 (2016): 1037-1060.

[5] Rassam, Murad A., M. A. Maarof, and Anazida Zainal. "A survey of intrusion detection schemes in wireless sensor networks." American Journal of Applied Sciences 9.10 (2012): 1636.

[6] Zamanifar, Azadeh, Eslam Nazemi, and Mojtaba

Vahidi-Asl. "A mobility solution for hazardous areas based on 6LoWPAN." Mobile Networks and Applications 23.6 (2018): 1539-1554.

[7] Sen, Arpan, Tanusree Chatterjee, and Sipra DasBit. "LoWaNA: low overhead watermark based node authentication in WSN." Wireless networks 22.7 (2016): 2453-2467.

[8] Seo, S. H., Won, J., Sultana, S., & Bertino, E. (2014). “Effective key management in dynamic wireless sensor networks”. IEEE Transactions on Information Forensics and Security, 10(2), 371-383.

[9] Gopikrishnan, S., P. Priakanth, and Rolly Maulana Awangga. "HSIR: hybrid architecture for sensor identification and registration for IoT applications." The Journal of Supercomputing (2019): 1-19.

[10] Alassaf, N., Gutub, A., Parah, S. A., & Al Ghamdi, M. (2018). “Enhancing speed of SIMON: a light-weight-cryptographic algorithm for IoT applications”. Multimedia Tools and Applications, 1-25.

[11] Hasan, Ragib, Mahmud Hossain, and Rasib Khan. "Aura: An incentive-driven ad-hoc IoT cloud framework for proximal mobile computation offloading." Future Generation Computer Systems 86 (2018): 821-835.

[12] de Farias, C. M., Brito, I. C., Pirmez, L., Delicato, F. C., Pires, P. F., Rodrigues, T. C., ... & Batista, T. (2017). “COMFIT: A development environment for the Internet of Things”. Future Generation Computer Systems, 75, 128-144.

[13] Du, W., Deng, J., Han, Y. S., & Varshney, P. K. (2006). “A key predistribution scheme for sensor networks using deployment knowledge”. IEEE Transactions on dependable and secure computing, 3(1), 62-77.

[14] Du, W., Deng, J., Han, Y. S., Varshney, P. K., Katz, J., & Khalili, A. (2005). “A pairwise key predistribution scheme for wireless sensor networks”. ACM Transactions on Information and System Security (TISSEC), 8(2), 228-258.

[15] Rahman, Sk Md Mizanur, and Khalil El-Khatib. "Private key agreement and secure communication for heterogeneous sensor networks." Journal of Parallel and Distributed Computing 70.8 (2010): 858-870.

[16] Huang, Q., Cukier, J., Kobayashi, H., Liu, B., & Zhang, J. (2003, September). Fast authenticated key establishment protocols for self-organizing sensor networks. In Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications (pp. 141-150). ACM.

[17] Mavani, Monali, and Krishna Asawa. "Privacy enabled disjoint and dynamic address auto-configuration protocol for 6Lowpan." Ad Hoc Networks 79 (2018): 72-86.

[18] A. P. Renold, and A. B. Ganesh. "Energy efficient secure data collection with path-constrained mobile sink in duty-cycled unattended wireless sensor network." Pervasive and Mobile Computing (2019).

[19] M. Babar, F. Khan, Iqbal, W., Yahya, A., Arif, F., Tan, Z., & Chuma, J. M. (2018). A Secured Data Management Scheme for Smart Societies in Industrial Internet of Things Environment. IEEE Access, 6, 43088-43099.

[20] P. M. Kumar, and U. D. Gandhi. "Enhanced DTLS with CoAP-based authentication scheme for the internet of things in healthcare application." The Journal of Supercomputing (2017): 1-21.

[21] A. Anand, M. Conti, P. Kaliyar, and C. Lal, “TARE: Topology Adaptive Re-keying scheme for secure group communication in IoT networks”. Wireless Networks, 1-15, 2019.

[22] A. Bahramlou, and R. Javidan, "Adaptive timing model for improving routing and data aggregation in Internet of things networks using RPL." IET Networks 7.5 (2018): 306-312.

[23] Samian, N., Zukarnain, Z. A., Seah, W. K., Abdullah, A., & Hanapi, Z. M. (2015). “Cooperation stimulation mechanisms for wireless multihop networks: A survey”. Journal of Network and Computer Applications, 54, 88-106.

[24] Sakthivel, T., and R. M. Chandrasekaran. "A Dummy Packet-Based Hybrid Security Framework for Mitigating Routing Misbehavior in Multi-Hop Wireless Networks." Wireless Personal Communications 101.3 (2018): 1581-1618.

[25] Rouissi, Nejla, and Hamza Gharsellaoui. "Improved hybrid LEACH based approach for preserving secured integrity in wireless sensor networks." Procedia computer science 112 (2017): 1429-1438.

[26] Daemen, Joan, and Vincent Rijmen. "AES proposal: Rijndael." (1999).

[27] Renold, A.P., &Athi, B.G. (2019). Energy efficient secure data collection with path-constrained mobile sink in duty-cycled unattended wireless sensor network. Pervasive Mob. Comput., 55, 1-12.

[28] N. Kawasaki, “Parametric study of thermal and chemical nonequilibrium nozzle flow,” M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.