Enhancing Security in IoT Networks: Advancements in Key Exchange, User Authentication, and Data Integrity Mechanisms

Alumuru Mahesh Reddy[1], Dr. M. Kameswara Rao[2]
Research Scholar, Department of ECM, Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur, India[1]
Professor, Department of ECM, Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur, India[2]

***Abstract*—Future Internet (FI) will be shaped by the Internet of Things (IoT), however because of their limited resources and varied communication capabilities, IoT devices present substantial challenges when it comes to securing connectivity. The adoption of robust security measures is hindered by limited compute power, memory, and energy resources, hence diminishing the promise for improved IoT capabilities. Confidentiality, integrity, and authenticity are ensured via authentication mechanisms are influenced by privacy needs, which are driven by sorts of customers that IoT networks service. Authentication is crucial in vital industries like linked cars and smart cities where hackers might use holes to access sensor data. Verification of the Gate Way Node (GWN), which is responsible for mutual authentication, user and sensor registration, and session key creation, is essential. The efficiency of key creation has been enhanced to tackle temporal intricacies linked to different key lengths. With notable advantages, the novel method shortens the time required to generate cryptographic keys: only 60 milliseconds for 100-bit keys and 120 milliseconds for 256-bit keys. This improvement fortifies resistance against new cyber threats by strengthening security basis of IoT networks and enhancing responsiveness and dependability. Through open transmission channels, users send login requests, and after successfully authenticating, they create session keys to establish secure connections with cloud servers. Python simulation results show how resilient the system is to security threats while preserving affordable interaction, processing, and storage. This development not only strengthens IoT networks but also guarantees their sustainability in the face of changing security threats.**

Keywords*—*IoT; Public key; key authentication; gate way node; data integrity mechanisms

I. Introduction

Improvements in wireless communication, embedded systems, and energy-efficient radio technologies over the past decade were crucial in enabling tiny devices to react and monitor their surroundings and shape a new networking paradigm able to act upon physical objects, ushering in the era of the Internet of Things (IoT) [1]. Connecting "Anything" to "Anyplace" and "Anytime" enables the third dimension of the Internet of Things vision, which will lead to the development of new applications and services that will affect the ecological, medical, financial, and social well-being. The potential for IoT to revolutionize human interaction with the physical environment is enormous. Smart cities, health monitoring, home automation, smart transportation, smart agriculture, and smart grids are just a few of the many possible uses for the Internet of Things. By 2020, there will be almost 50 billion "things" connected to the internet, or IoT devices, according to a recent study by CISCO [2], [3]. IoT's enormous potential has led some to call it the "next wave" of the Internet. The widespread use of IoT technology and applications relies heavily on their security. Without guarantees in terms of system-level confidentiality, authenticity, and privacy, IoT solutions are unlikely to be adopted on a large scale. It is difficult to establish end-to-end secured communications between IoT entities because of the heterogeneity of IoT and because the majority of IoT devices are resource constrained. Organizational and academic researchers continue to focus on the problem of IoT security. Fig. 1 shows the Future Vision of IoT.

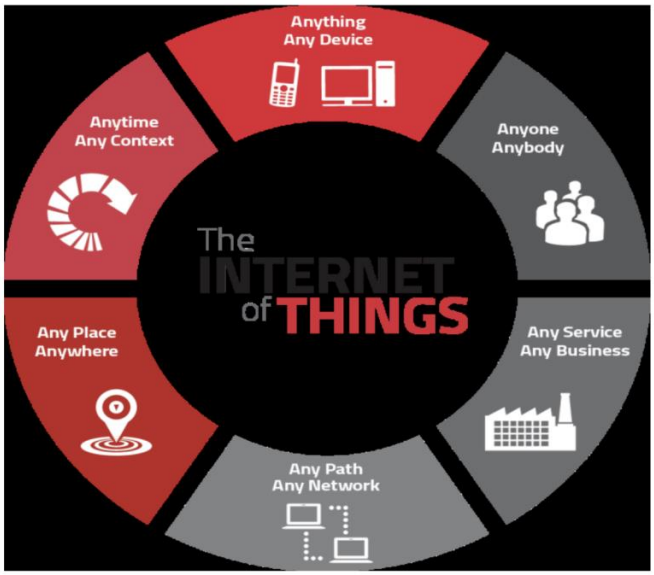


Fig. 1. Future vision of IoT.

The number of devices connected to the Internet of Things (IoT) is rising. The Internet of Things (IoT) encompasses not just traditional computing and communication equipment but also a wide variety of other devices utilized in various spheres of everyday lives. As a result, hundreds of billions of gadgets could end up linked together [4]. In the IoT ecosystem, inanimate things automatically share data and communicate

with one another over the Internet. With IoT, information may be shared between living and nonliving things to accomplish tasks. The data they collect, evaluate, and act upon can all be done automatically by those devices [5]. Authentication can be thought of as the first line of defense because it guarantees that security procedures will be followed. A handshake is an optional authentication procedure that must be completed before permission may be given. Authenticating a device entails checking its claimed characteristics [6]. Sensitive information can be stolen and malicious acts carried out if authentication is weak. One of the first stages in ensuring the security of the entire system is to deploy strong authentication procedures [7]. Before any information can be transferred between IoT devices, authentication must take place. Traditional authentication methods often need one of the well-known authentication elements, such as a secret the user knows or a token the user possesses, to verify the identity of a user.

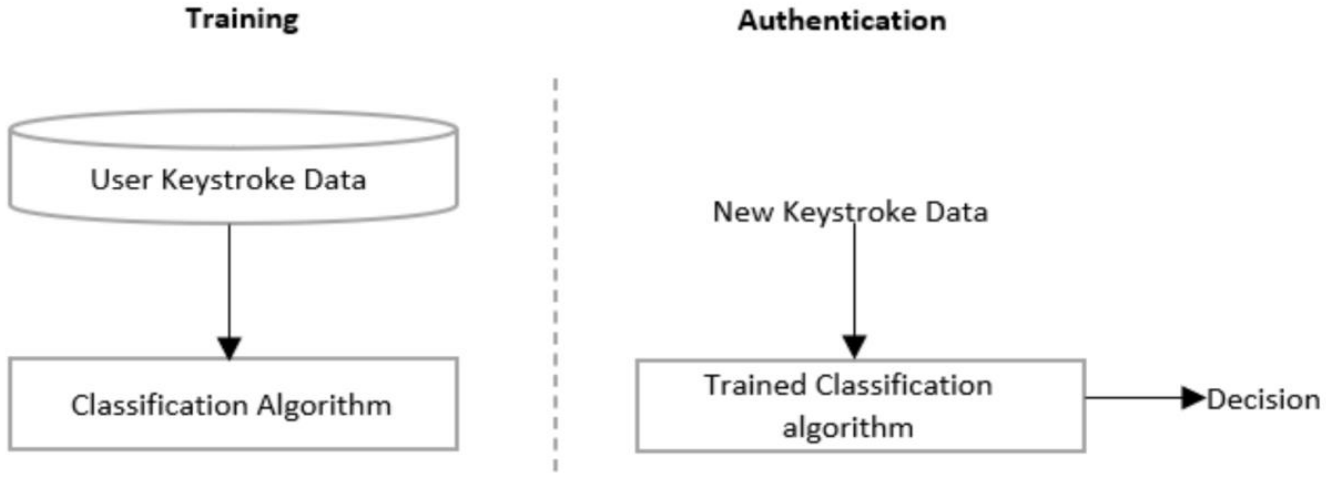


Fig. 2. Process of continuous authentication.

Two-factor authentication and multi-factor authentication are two examples of current authentication methods that use multiple authentication factors to verify the identity of a person or device [8]. To make it extremely difficult, if not impossible, for an unauthorized party to obtain access to a protected resource, security experts have developed two- or multi-factor authentication systems [9]. Two-factor authentication methods have gained popularity in recent years as a means to guarantee safe system logins by adding an extra layer of security and inspiring user confidence. There are two main types of authentication procedures: machine-to-machine and human-to-machine [10]. Classification is the challenge of determining to which group an observation belongs in a statistical classification scheme. Assigning a spam or non-spam classification to an email or a patient's diagnosis based on observable characteristics (Sex, blood pressure, the presence or absence of particular symptoms, etc.) are two examples. Continuous authentication makes use of keyboard data for user classification using classification algorithms. The continuous authentication process is depicted in Fig. 2. The user's new keystroke data is fed into the trained algorithm, which was itself trained using keystroke data [11]. Authentication is considered to have been successful if the categorized user is the same as the input user. Users are denied access if the algorithm's classification of them does not match the supplied user. The system architecture of the proposed work incorporates components from the Cloud, Industry, Sensing Devices, Gateway Node (GWN), Trusted Third Party, and Sensing Devices to provide a safe framework for industrial monitoring. Users with smart cards may monitor factories remotely over the internet, protecting comprehensively address the multifaceted aspects of advancing security protocols in IoT networks. Section I introduces the overarching challenges in IoT security and highlights the privacy of any data that is sent. To effectively transmit sensor data, steps for key agreement and authentication are started by the consumer, the GWN, and the sensors. The GWN creates and distributes session keys, registers sensors, and preloads them with credentials to enable secure connection. In industrial contexts, this method improves user oversight and data transfer security. Energy consumption must be taken into account while implementing efficient key generation techniques in the Internet of Things. In addition to increasing computing performance, optimizing key generation lowers energy consumption, which is essential for IoT devices with constrained power sources.

The following is the proposed work's primary contribution,

- Enhanced security without compromising performance is made possible by optimized algorithms, which are essential for Internet of Things devices with limited resources.
- Secure access to sensing devices is ensured via smart card-enabled user authentication through GWN, enhancing system integrity overall.
- The timely monitoring of production processes made possible by the prompt transfer of sensor information to users improves productivity and decision-making skills.
- Enabling safe connections between users and sensing devices, the authenticated key agreement mechanism assures the production of shared session keys.

- The suggested system design provides a secure and reliable data transmission framework, enabling users to efficiently monitor industrial operations.

The study is divided into five parts that concentrate on improvements in data integrity, user authentication, and key exchange. Section I provides the Authentication Methods in IoT Security. Group key management mechanisms in IoT networks are examined in Section II. The limitations of traditional approaches are covered in Section III. The process for enhancing key generation algorithms is described in Section IV. Section V presents the results and discussion, while Section VI provides a final summary.

II. Classification of Group Key Management Protocols based on IoT Network Nodes

With the development of so many joint programs, the importance of group key management has grown substantially. There are two main types of group communication: static and dynamic. Members of a static group do not rotate or shuffle about. Keys do not need to be updated after they have been disseminated. Members of a dynamic group, on the other hand, come and go regularly. The Group Controller (GC) is responsible for handling the frequent key updates necessary to keep forward and backward secrecy[12]. Fig. 3 displays the variety of GKM methods now in use. Centralized key management refers to the method by which all key management functions are performed by a single organization. In the distributed GKM method, a member of the group is chosen on the fly and given responsibility for running the GKM. It is possible to categorize IoT network group key management protocols according to the types of network nodes used in the protocol. Each participant in the contributory key management method does its part in generating the group key. Both tree-based and non-tree-based models can be used to accomplish these important management strategies [13]. The literature on GKM is rife with examples of tree-based models. In tree-based models, each leaf stands for a user, and each node in the tree's path to the root represents that person's auxiliary keys. The inherent hierarchy of a tree-based organization is implicitly encouraged. The rekeying process is simplified by the group controller's logical organization of the keys within the structure. Some popular types are as follows:

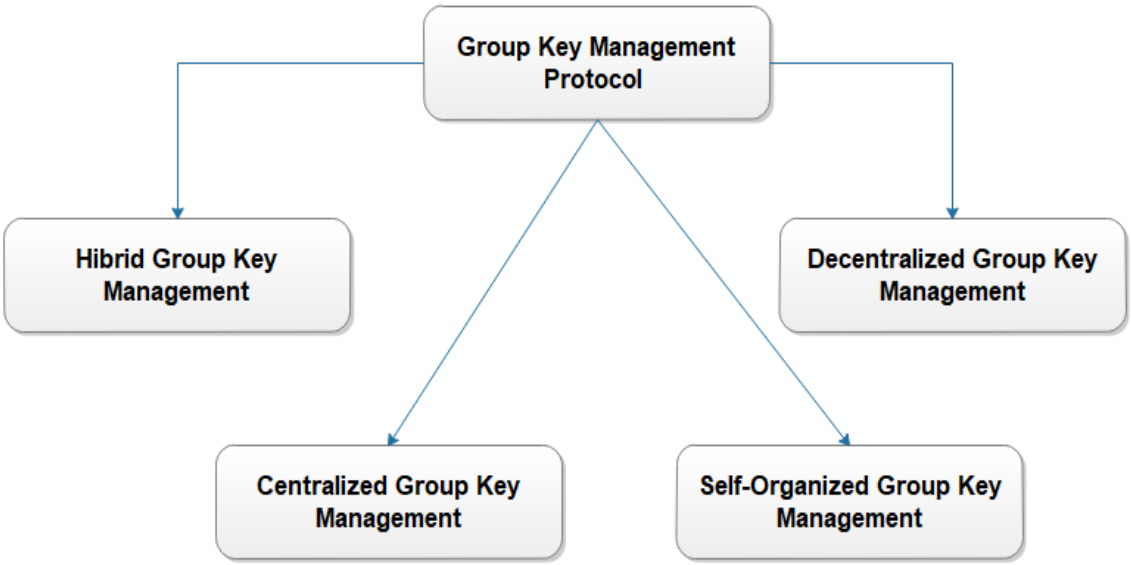


Fig. 3. Group key management protocol.

A. Centralized Group Key Management

The group keys in an IoT network are managed by a centralized authority in this sort of protocol. The group keys are generated, disseminated, and kept up-to-date by a centralized authority. Logical Key Hierarchy (LKH) and Group Key Management Protocol (GKMP) are two exemplars of centralized group key management protocols[14]. One way to handle group keys in a network is using centralized group key management, in which a single server or other coordinating body is in charge of key generation, distribution, and upkeep for all users. Secure communication between a set of network nodes is the primary function of group keys. The central authority in a centralized group key management system is responsible for managing the production and distribution of keys. The group keys are generated by a trusted source and safely disseminated to all participating nodes via cryptographic protocols. Network nodes make contact with a centralized authority to request and receive the group keys required for secure group communication. The coordinating body carries out a number of crucial managerial tasks, including:

1) Key generation: The central authority uses cryptographic procedures to produce random, robust group keys.

2) Key distribution: Once the group keys have been generated, they will be sent out to all of the connected devices. To protect the privacy and safety of the keys, they might be dispersed through encrypted channels or protocols.

3) Key updates: The group keys may be updated at regular intervals by the central authority to account for network changes and new security standards. This keeps the group's communication secure by ensuring that compromised or obsolete keys are changed.

4) Key revocation: The central authority can revoke the associated key in the event that a node is compromised or departs the group. In this way, unrecognized nodes are denied access to the group's communication.

Smaller networks with a controllable number of nodes and generally stable network topologies are often good candidates for centralized Group Key Management. Scalability, security, and privacy issues in bigger or more dynamic IoT systems should be carefully considered, despite the fact that it provides a clear and regulated solution to group key management.

B. Decentralized Group Key Management

Multiple entities or nodes in the IoT network share the duty of key management thanks to decentralized protocols[15]. These methods do not require a governing body to control keys. Using methods like key trees or shared key derivation, nodes coordinate to set up and maintain group keys. Both the Tree-based Key Management Protocol (TKMP) and the Distributed Group Key Management Protocol (DGKMP) are examples of protocols for decentralized group key management. When it comes to handling group keys in a network, decentralized group key management is the way to go because it allows for key production, distribution, and maintenance to be handled by a number of different entities or nodes. Decentralized group key management is based on cooperation amongst nodes, as opposed to centralized group key management, which is controlled by a single body[16]. Decentralized Group Key Management has the following salient features and qualities:

5) Key generation: In a decentralized system, many different nodes work together to generate keys. The group keys are generated by these nodes working together using cryptographic procedures. Methods like key derivation and tree-based key establishment can be used in the key generation process.

6) Key distribution: Decentralized protocols use procedures for key distribution among the participating nodes rather than depending on a centralized entity. In order to disperse the group keys, the nodes in the network either disclose their keying material or engage in key exchange protocols. To protect the privacy and authenticity of the key distribution procedure, secure channels or protocols can be used.

7) Key updates: Without requiring a central authority, decentralized Group Key Management techniques allow for instantaneous key upgrades. When changes are required to the group keys, such as when new nodes join or depart the group, the nodes work together to make such changes. Because of this adaptability, the network can better accommodate shifts in its constituent nodes.

8) Key revocation: Decentralized protocols deal with the issue of key revocation in the event that a node is compromised or unauthorized in the same way that centralized methods do. In order to keep group communication safe in the face of compromised keys, nodes coordinate a process of revocation.

For larger IoT networks, especially ones with fluid memberships or dispersed designs, decentralized Group Key Management methods are a good option. Decentralized techniques offer scalability, resilience, and adaptability but require special consideration of complexity, communication overhead, and trust issues.

C. Hybrid Group Key Management

When it comes to managing group keys, hybrid protocols incorporate the best features of both centralized and decentralized systems. They use the strengths of the two models to create a workable, scalable plan[17]. The initial group keys, for instance, may be generated and distributed by a centralized body, while subsequent key modifications may be handled independently by each participant. The goal of hybrid protocols is to strike a compromise between centralized management and decentralized robustness. HKMP and CDKMP are two examples of hybrid group key management protocols. A hybrid way to managing group keys in a network, Hybrid Group Key Management takes the best features of both centralized and decentralized systems. Its goal is to provide an effective and scalable solution for group key management in IoT networks by combining the best features of the two models[18]. The key management process in Hybrid Group Key Management is split between centralized and decentralized elements. Some critical administration tasks are under the purview of the coordinating body, while others are delegated to the various nodes. Depending on the protocol and the demands of the network, the precise allocation of tasks may change. The essential features and qualities of Hybrid Group Key Management are as follows:

1) Centralized functions: The coordinating body carries out essential management tasks, such as

a) Key generation: The first group keys are generated using cryptographic procedures by a centralized body.

b) Key distribution: The initial group keys are dispersed to the nodes by the coordinating body.

c) Policy enforcement: The governing body regulates who is allowed to access the group keys and what they may do with them.

2) Decentralized functions: The participating nodes work together and supply input for a variety of critical management tasks, including:

a) Key updates: When necessary, nodes work together to update keys by creating new ones or modifying existing ones.

b) Key revocation: When a node is compromised or leaves the group, the other nodes work together to revoke its access to the shared keys.

c) Key distribution and storage: Group keys can be distributed to newly joined nodes by existing nodes, or keys can be securely stored and shared across nodes.

3) Coordination and communication: Hybrid Group Key Management techniques call for a centralized organization and all participating nodes to coordinate and communicate with one another[19]. Distributing initial group keys, enforcing policies, and receiving updates all need communication between the central entity and the nodes. The updating, revoking, and distribution of group keys are all processes that require cooperation between nodes.

Hybrid Group Key Management techniques combine the benefits of centralized and decentralized approaches to group key management. They may function for IoT networks with various needs for scalability, compositional flexibility, and security policy variety. However, care must be taken during design and implementation to handle the complexity and trust issues raised by the hybrid nature of these protocols.

D. Self-Organizing Group Key Management

Nodes can create groups and generate group keys independently of any central authority or predetermined network architecture thanks to self-organizing protocols. The establishment, distribution, and maintenance of keys are all key management processes that need cooperation across nodes[20]. These protocols work well in ever-changing, low-resource Internet of Things settings. Protocols like Peer Group Key Management (PGKM) and SOGM are examples of self-organizing group key management. In self-organizing group key management, nodes in a network work together to set up and manage group keys without any central authority. In this method, keys are not managed by a centralized authority or within a strict framework. Instead, the participating nodes work together to carry out essential management functions. Self-Organizing Group Key Management has the following salient features and qualities:

1) Autonomous group formation: The network's nodes will naturally cluster together based on shared characteristics or geographic closeness, for example. These clusters could evolve over time as nodes enter and exit the network.

2) Key establishment: Each group's keys are determined by a consensus of the members. To generate a group key securely, they may use a key establishment technique like Diffie-Hellman key exchange or elliptic curve cryptography. In most cases, a combination of safe pairwise communication and cryptographic activities makes up the key establishment procedure.

3) Key distribution: After a group key has been formed, it will be shared across the participating nodes. The key can be efficiently disseminated to all group members by either direct communication between nodes or through the use of multicast communication mechanisms. The delivery of the key is encrypted or conducted through a secure channel to prevent unauthorized parties from gaining access to it.

4) Key updates: Nodes in a self-organizing system coordinate the distribution of critical updates in response to shifts in the group's make-up or the level of protection it needs. The group key is updated by consensus whenever there is a change in membership due to either new or departing nodes. This replaces revoked or compromised keys to keep group communication safe.

Protocols for self-organizing key groups provide a decentralized and autonomous solution for IoT network key management. They work well in situations where a centralized authority would be impossible due to a lack of stability or sufficient resources[21]. However, self-organizing systems necessitate careful study and robust processes to assure the entire system's integrity and resilience, particularly in the areas of security, scalability, and management. It should be noted that the aforementioned categorization is not comprehensive, and that different group key management protocols used in IoT networks may utilize a variety of modifications or combinations of these classifications.

III. PROBLEM STATEMENT

The constant change of their surroundings combined with the complexity and diversity of IoT networks may lead to limitations. Furthermore, the classification offered could not cover all conceivable iterations or pairings of group management of key protocols utilized in Internet of Things networks, which could restrict its usefulness in specific situations[18].The proposed work entails creating a thorough framework that takes into consideration the complicated and changing characteristics of settings for group management of keys in Internet of Things networks. By providing flexible and adaptive protocols, this framework will solve the shortcomings of existing classifications and efficiently handle a variety of network conditions.

IV. METHODOLOGY

The system architecture utilized in this study is elucidated in Fig. 4, comprising six integral components: Trusted Third Party, Gateway Node (GWN), Sensing Devices, User, Cloud, and Industry. Positioned as a top-level industry official, the user possesses the capability to remotely monitor individual factories via the web at regular intervals. Crucially, maintaining the privacy of data transmitted among the user, GWN, and sensors is imperative [19]. Smart card-equipped users leverage the GWN to solicit access to the sensors. The information collected by sensors is promptly relayed to the user in near-real time. An authenticated key agreement process is initiated with a login request transmitted from the user to the GWN. Subsequently, the GWN verifies the user's identity and forwards the request to the sensors. In response to GWN's request, sensing devices provide their secret shares, allowing GWN to reconstruct the secret value. Utilizing this reconstructed secret value, sensing devices generate a shared session key and transmit messages to the user securely. Ultimately, the user gains access to sensor-collected data, empowering them to efficiently oversee and manage the manufacturing process. This intricate system architecture establishes a secure and efficient framework for data transmission and user interaction within the context of industrial monitoring.

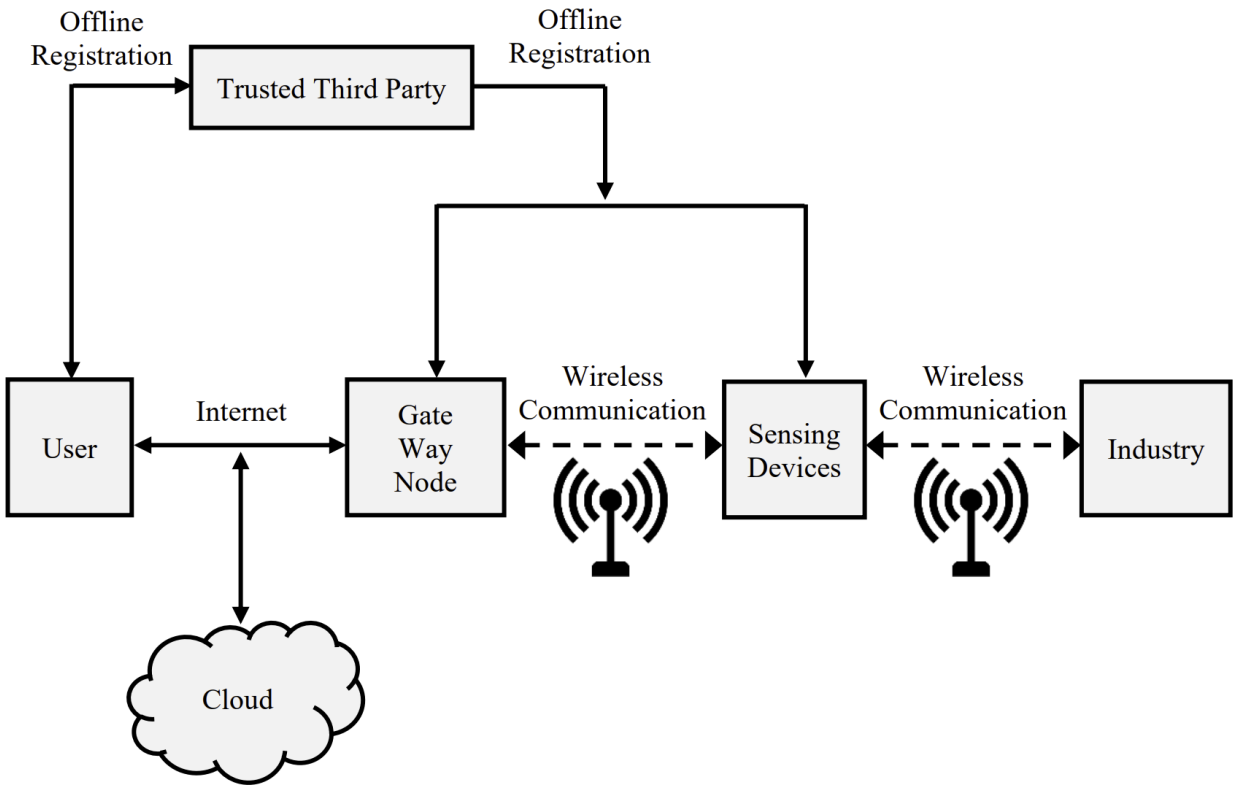
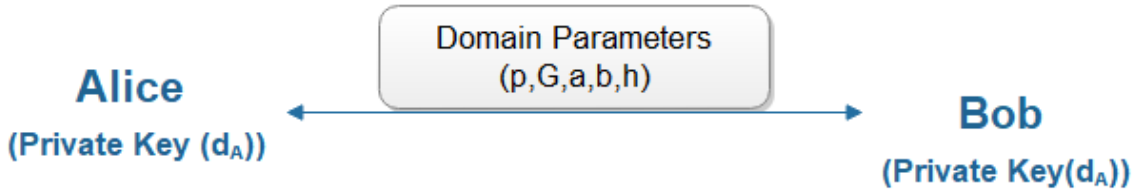


Fig. 4. Proposed system architecture.



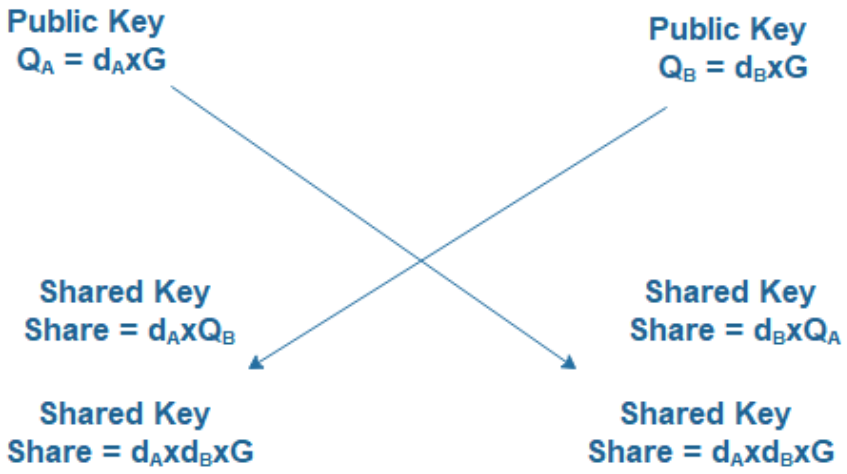


Fig. 5. Generalized key management scheme.

The suggested approach relies on a trustworthy third party, the Gate Way Nodes (GWN), which not only registers the sensors but also pre-loads them with credentials. At first, the user and GWN utilize the TTP private key to generate a shared secret key. The idea of existing cryptography algorithms is used to calculate this shared secret key[22]. The user then sends the GWN node the key agreement protocol using the shared secret key. The user transmits a set of confidential parameters to the GWN, which in turn causes the GWN to generate a group key. This group key is passed around in a safe manner. At this point, the sensors use the group key to generate a new session key for use in GWN communications between the user and sensors. The user will also be given the group key from GWN to use in generating the session key. The session key computed by the user and the sensors must be same for the protocol to function properly. The user and the sensors will communicate over GWN by exchanging the industrial data using this session key.

In Fig. 5, is an example of a centralized GKM scheme, in which all group key management functions are handled by a single location. It's crucial that the key is kept secret and only the authorized individuals have access to it. New group members join and old group members leave frequently in most group focused applications.[4] It is crucial to keep both forward secrecy (wherein new members are not revealed with the old key) and backward secrecy (wherein former members are not revealed with the new key) in place. The ability to easily share data in the cloud is essential for businesses and organizations that have made the decision to move their operations to the cloud. The businesses have benefited from working with their contemporaries because it has increased output. As a result, healthcare expenditures decrease and doctors have a more complete picture of their patients' health. Students have little trouble cooperating on group assignments.

Sharing data in the cloud always involves more than one person having access to that data, making data privacy and security paramount. Protecting data privacy while enabling data sharing is of paramount importance. Generation, distribution, and updating of group keys for use in encryption and decryption are all critical functions of group key management in cloud data sharing.

The best encryption is useless without secure key management. Therefore, a new Compressed Trie based Group Key Distribution (CTGKD) method is proposed in this chapter to ensure the construction of trustworthy groups and the safe distribution of keys. The primary focus of this effort is on decreasing rekeying-related communication and computation costs. In this case, a secure group is formed using a compressed trie structure, and keys are dispersed among the members. The steps of the proposed Compressed Trie based Group Key Distribution (CTGKD) protocol are shown in Fig. 6.

1) Key generation: The key freshness attribute, in which the session key is never reused, is an important part of the security architecture of group key establishment.

The gateway and the group members then safely exchanged a session key after this. The gateway then encrypts the session key with the shared long-term secret S. Given that the proxy has a share (ci,di), he needs to acquire (m-1) shares from the other members of the group in order to rebuild the secret S and extract the session key. Fig. 7 and its explanation follow to provide more detail about this stage.

2) Key distribution and verification: Message authentication and message integrity of key shares between the gateway and each member of the group are required because the communication medium between the gateway and the group members is an unprotected public wireless channel. The gateway uses a lightweight and safe approach based on cryptographic hash functions and the xor operation to disperse the shares. To ensure the safe transfer of the secret shares (ci, di), mutual authentication between the GWN, Nj, and P is kept active at this stage. If an attacker tried to reconstruct the secret S, at least (m=5) out of n shares are necessary to recover the secret S, and even if the attacker acquired (m-1shares), he still cannot recover S. At the completion of this phase, each node in the group will have a share. The values (ci, di) are used by the proxy node to reconstruct the secret S, which is then used to encrypt the session key SK, which is used for encryption/decryption of communication between the GWN and the multicast group members n, after an authenticated shares distribution has taken place. Fig. 8 shows the Development of Key Exchange, User Authentication and Data Integrity mechanisms for IoT based network.

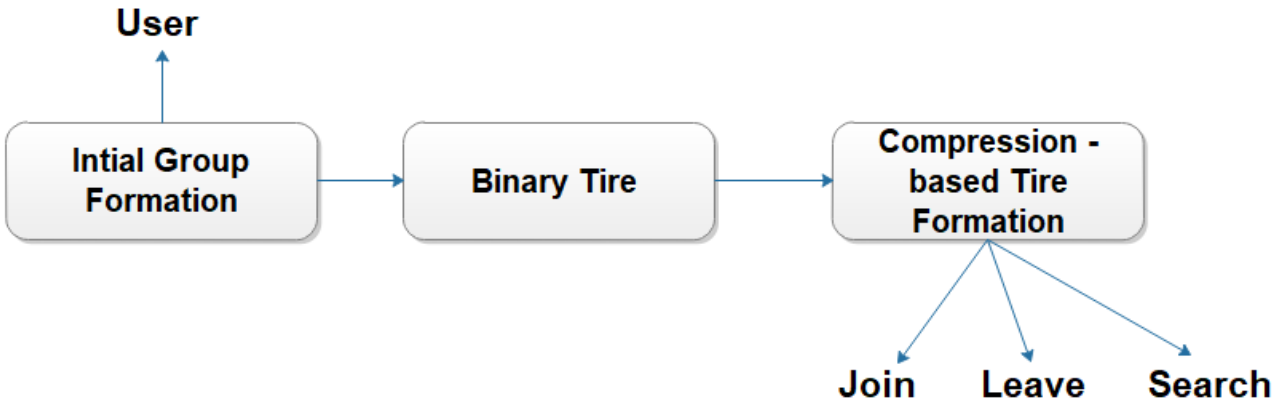


Fig. 6. Block diagram for the proposed protocol.

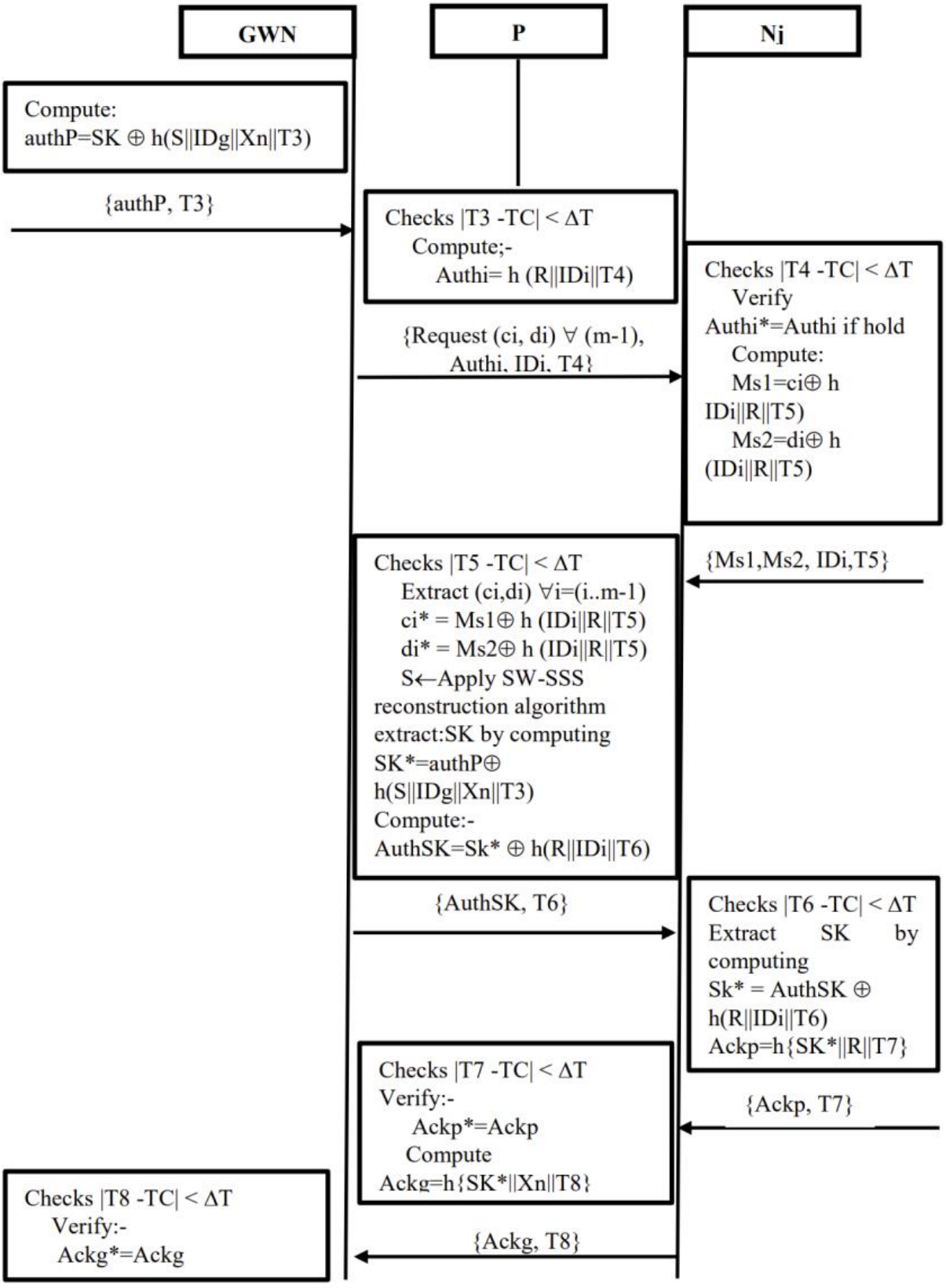


Fig. 7. Key generation and verification.

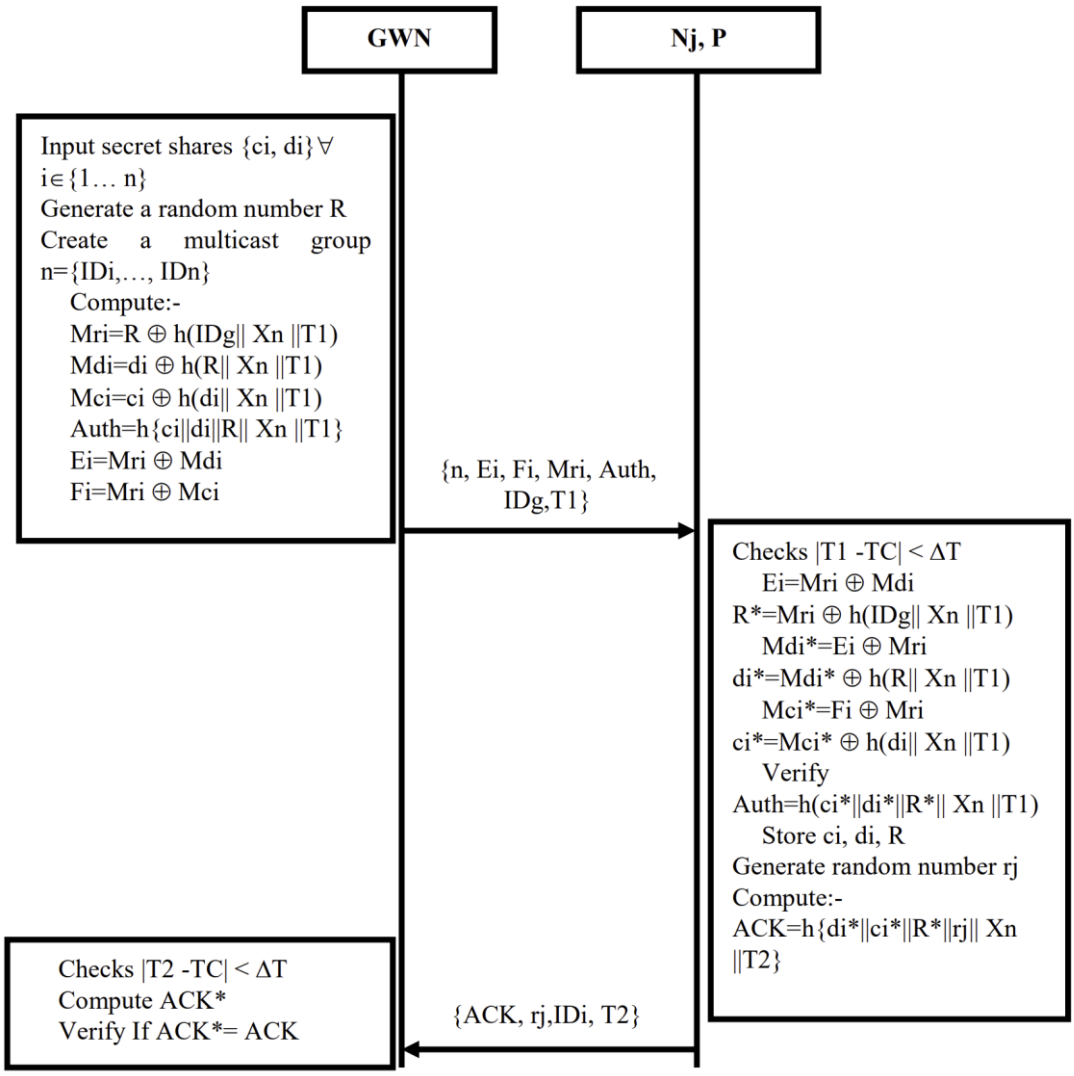


Fig. 8. Development of key exchange, user authentication and data integrity mechanisms for IoT based network.

V. Discussion and Result

Memory use, response time, MAC generation time, and security overhead are all evaluated to gauge how well the proposed framework performs. The suggested framework has a high security cost. Consumption of memory is the amount of computer memory actually being put to use storing information. Memory usage for the proposed system is depicted in Fig. 9 in relation to the encrypted data partitions. Table I numbers show that the amount of data storage needed for each component is roughly the same size.

TABLE I. Memory Consumed by Various Partitions of the Split Encrypted Data

Partition	Memory Used (in Bytes)
Partition 4	442
Partition 3	453
Partition 2	438
Partition 1	442

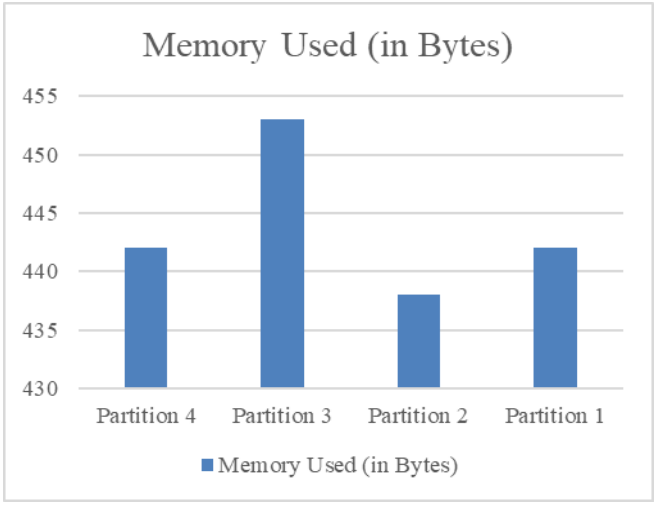


Fig. 9. Plot for memory consumed by various partitions of the split encrypted data.

The time it takes to generate a key at different key lengths is shown in Fig. 10 and Table II. The time required to generate a key grows proportionally with its length. Key generation for 100-bit keys takes 60 milliseconds, for 256-bit keys it takes 120 milliseconds, and so on.

TABLE II. THE TIME REQUIRED TO GENERATE A KEY GROWS PROPORTIONALLY WITH ITS LENGTH

Key Length (bits)	Key generation (ms)
1024	800
512	320
256	120
128	80
100	60

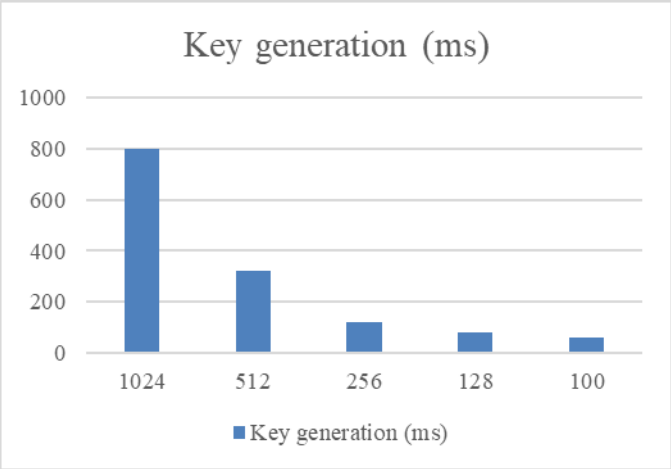


Fig. 10. Plot for the time required to generate a key grows proportionally with its length.

System delay on either the end-user or the service provider's end might contribute to the security burden. The security overhead causes a system delay because of the time it takes to verify and decrypt data on the user's side and to generate and encrypt data on the owner's side. By dividing the total duration of transmission by the total time of security operations (MAC Generation/Verification and Encryption/Decryption), get the security overhead percentage. The burden of data transfer must be minimized.

Fig. 11 depicts the Plot for Security overhead at Data Owner. PUF, a one-way hash function, a bitwise XOR operation, and symmetric encryption will be used in this stage to build two-factor mutual authentication. Throughout this stage, messages are encrypted using the AES method and a 128-bit key length to ensure their safety throughout transmission. Data integrity between N and the IG is also validated and guaranteed by employing a 256-bit cryptographic hash function that operates in one direction only. The parties exchange proposals for how to generate session keys during this stage. The parties might choose to use the Elliptic Curve Diffie Hellman Key Exchange Protocol (ECDH) or a one-way hash function to produce the session key. To generate a common secret key, you can utilize a key-agreement technique like ECDH Key Exchange. Key exchange is depicted in Fig. 12.

To guarantee the highest level of source location anonymity, combined two steps in the proposed method: random multipath and tunnels with spoofed communications. To determine how well the proposed method conceals the location of the source, we must calculate the safety period, which is defined as the estimated number of hops an adversary must take to retrace their steps from the sink to the source. With the suggested method, the hop count can range from 10 to 35, and each relaying node has a probability of P = 0.8 of generating a tunnel with a length L, i=0.5, and D=3. In Fig. 13, show how the suggested method greatly lengthens the safety time compared to RPL, and how it may be used to protect the anonymity of the source location. We found that the source location privacy is better protected and the safety duration is longer with a longer tunnel. However, the suggested method still outperforms RPL in safeguarding the confidentiality of the source location, despite the fact that the safety time noticeably increases when tunnels length L= 10 and reduces when tunnels L = 3.

TABLE III. SECURITY OVERHEAD AT DATA OWNER

File Size (MB)	Security Overhead at Data Owner	
	Symmetric Encryption with AES-256	*Proposed Method*
800	12.9	12.2
700	12.7	12.1
600	12.5	11.8
500	13.4	12.9
400	12.8	12.5
300	13.7	13.4
200	15.1	14.8
100	16.5	16

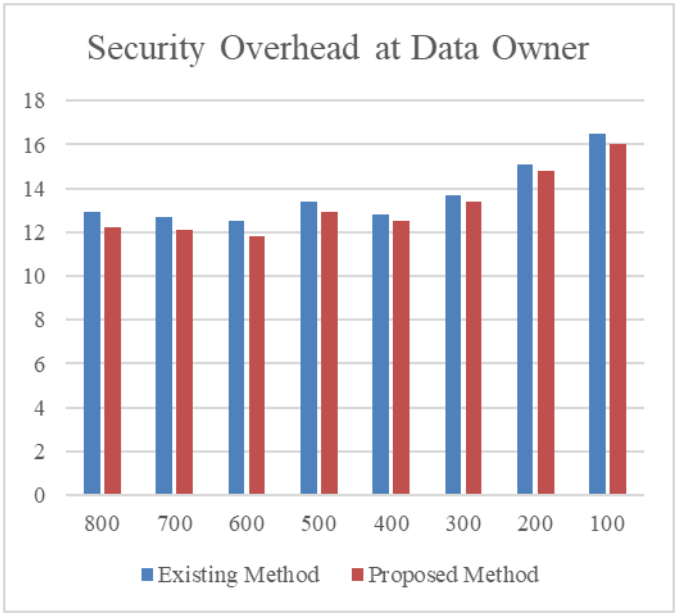


Fig. 11. Plot for security overhead at data owner.

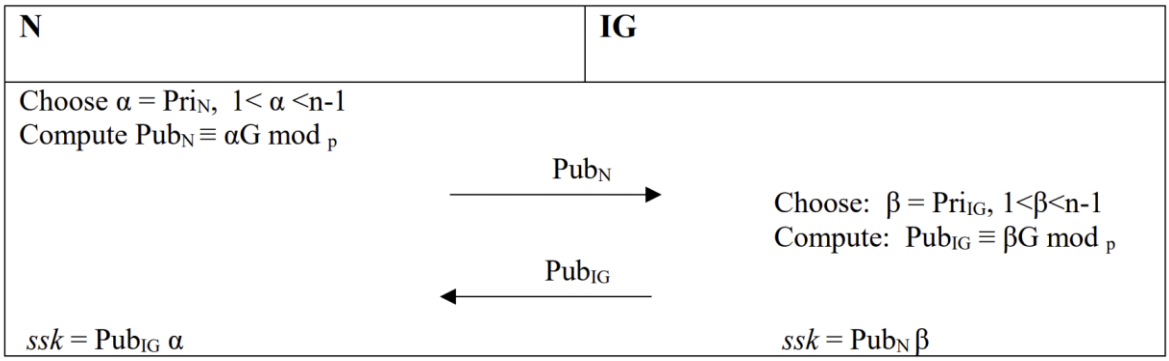


Fig. 12. Key exchange protocol.

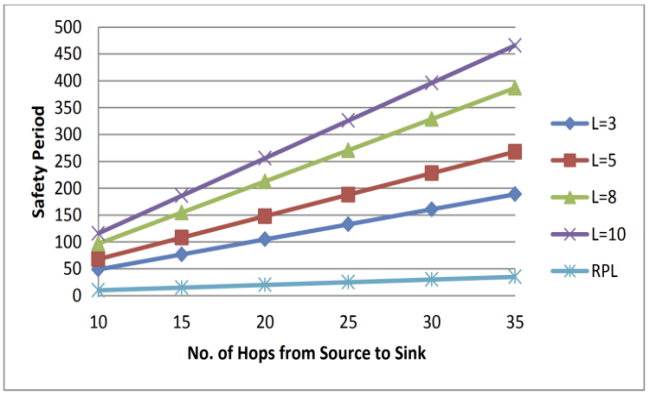


Fig. 13. Plot for the safety period vs No. of hops from the source link.

A. Discussion

Because of the limited resources and various communication capabilities of IoT devices, the discussion of the offered remark emphasises how crucial it is to solve security issues in IoT networks. Although key generation process optimisation presents promise increases in efficiency, more study is necessary to investigate complete security measures that can adapt to emerging cyber threats. Future research may concentrate on creating cryptographic algorithms that are lightweight and sensitive to the constraints of Internet of Things devices, improving their security without appreciably raising computing cost. Additionally, studies should look at fault tolerance and anomaly detection techniques as ways to lessen the impact that hacked sensors might have on network communication protocols. Notwithstanding the noteworthy progress, it is imperative to recognize its limits, such as the possible trade-offs among security and resource restrictions and the continuous upgrades and maintenance required to handle new vulnerabilities. Given the constantly changing cyber dangers, this emphasizes the need for ongoing multidisciplinary interaction among researchers, industry stakeholders, and policymakers to guarantee the sustainability and resilience of IoT ecosystems.

1) Key generation optimization: Central to contributions is the optimization of key generation algorithms, acknowledging the proportional relationship between key length and generation time. The substantial reduction in key generation times, exemplified by 60 milliseconds for 100-bit keys and 120 milliseconds for 256-bit keys, marks a significant stride in bolstering the efficiency of security processes. This enhancement not only addresses a pressing issue in existing protocols but also positively impacts the responsiveness of IoT systems, mitigating potential vulnerabilities during key establishment phases [16].

2) User authentication and data integrity: The optimized key generation process plays a pivotal role in strengthening user authentication procedures. The authenticated key agreement, initiated through a login request from the user to the Gateway Node (GWN), ensures secure access to sensors. The exchange of secret shares between sensing devices and the GWN, leading to the reconstruction of the secret value, forms a robust foundation for secure communication and authentication. Moreover, the heightened efficiency in key generation positively influences data integrity. The secure and prompt transmission of sensor-collected data to the user in near-real time is instrumental in ensuring the reliability of the information. This, in turn, empowers users to make informed decisions and manage manufacturing processes with confidence.

3) System architecture and user empowerment: The elucidation of the system architecture, encompassing components such as the Trusted Third Party, GWN, Sensing Devices, User, Cloud, and Industry, provides a comprehensive understanding of the ecosystem. The user, positioned as a top-level industry official, gains the ability to remotely monitor factories, emphasizing the practical implications of our advancements in real-world scenarios.

Research not only addresses existing vulnerabilities in IoT security but propels the field forward by optimizing key generation, enhancing user authentication, and ensuring data integrity. Establishing an equilibrium between customization and standardization, guaranteeing interoperability with various IoT network topologies, and successfully managing dynamic security risks might present difficulties of proposed work. Subsequent efforts will focus on improving the suggested framework via empirical assessment, assessing its resilience and scalability in actual Internet of Things implementations, and investigating innovative methods for improved security and effectiveness.

VI. CONCLUSION

While low-power, low-performance devices are the foundation of IoT networks, the field of IoT safety and privacy has generated a lot of interest from academics recently. This work developed CTGKD, a unique strategy that uses a compacted trie-based structure to address the scalability difficulties of cloud key distribution. This approach differs from the traditional tree-like architectures seen in previous literature. This work is unique because it closes a gap in existing solutions by using compressed attempts to the key management problem. The findings of the performance analysis show that CTGKD is more effective at communicating than standard LKH tree-based key management. In the future, the emphasis will be on creating new methods and altering current security guidelines to achieve a balance between the strict protocol requirements and the resource-constrained characteristics of Internet of Things devices. In these methods for lightweight authentication, key creation, and origin location privacy have been developed to tackle some of the security issues in Internet of Things networks. To improve these techniques and guarantee that they are appropriate for IoT situations with limited resources, more study is necessary. Furthermore, while identification, confidentiality, and key management are addressed in the suggested security architecture for cloud data storage, access control methods are absent, making it impossible to guarantee correct data access by authorized users. To ensure allowed access to cloud data, further work will require incorporating access control procedures and regulations into the security architecture. Additionally, simplifying access for clients to all cloud services through the integration of Single Sign-On (SSO) within the security architecture would improve efficiency and user experience. These improvements will ensure that authorized users may obtain and use data safely and effectively while also improving the total safety and accessibility of IoT networks.

REFERENCES

[1] J.-D. Wu, Y.-M. Tseng, and S.-S. Huang, "An Identity-Based Authenticated Key Exchange Protocol Resilient to Continuous Key Leakage," IEEE Systems Journal, vol. 13, no. 4, pp. 3968–3979, Dec. 2019, doi: 10.1109/JSYST.2019.2896132.

[2] L. Meng, H. Xu, H. Xiong, X. Zhang, X. Zhou, and Z. Han, "An Efficient Certificateless Authenticated Key Exchange Protocol Resistant to Ephemeral Key Leakage Attack for V2V Communication in IoV," IEEE Transactions on Vehicular Technology, vol. 70, no. 11, pp. 11736–11747, Nov. 2021, doi: 10.1109/TVT.2021.3113652.

[3] Q. Fan, J. Chen, M. Shojafar, S. Kumari, and D. He, "SAKE*: A Symmetric Authenticated Key Exchange Protocol With Perfect Forward Secrecy for Industrial Internet of Things," IEEE Transactions on Industrial Informatics, vol. 18, no. 9, pp. 6424–6434, Sep. 2022, doi: 10.1109/TII.2022.3145584.

[4] J. I. E. Pablos, M. E. Marriaga, and Á. L. P. del Pozo, "Design and Implementation of a Post-Quantum Group Authenticated Key Exchange Protocol With the LibOQS Library: A Comparative Performance Analysis From Classic McEliece, Kyber, NTRU, and Saber," IEEE Access, vol. 10, pp. 120951–120983, 2022, doi: 10.1109/ACCESS.2022.3222389.

[5] V. Thakur, G. Indra, N. Gupta, P. Chatterjee, O. Said, and A. Tolba, "Cryptographically secure privacy-preserving authenticated key agreement protocol for an IoT network: A step towards critical infrastructure protection," Peer-to-Peer Networking and Applications, pp. 1–15, 2022.

[6] T.-C. Hsieh, Y.-M. Tseng, and S.-S. Huang, "A leakage-resilient certificateless authenticated key exchange protocol withstanding side-channel attacks," IEEE Access, vol. 8, pp. 121795–121810, 2020.

[7] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for internet of things," Big Data Mining and Analytics, vol. 4, no. 1, pp. 1–9, 2021.

[8] T.-T. Tsai, S.-S. Huang, Y.-M. Tseng, Y.-H. Chuang, and Y.-H. Hung, "Leakage-resilient certificate-based authenticated key exchange protocol," IEEE Open Journal of the Computer Society, vol. 3, pp. 137–148, 2022.

[9] I.-C. Lin, C.-C. Chang, and Y.-S. Chang, "Data security and preservation mechanisms for industrial control network using IOTA," Symmetry, vol. 14, no. 2, p. 237, 2022.

[10] M. I. G. Vasco, A. L. P. del Pozo, and C. Soriente, "A key for john doe: Modeling and designing anonymous password-authenticated key exchange protocols," IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 3, pp. 1336–1353, 2019.

[11] J. Zhang, X. Huang, W. Wang, and Y. Yue, "Unbalancing Pairing-Free Identity-Based Authenticated Key Exchange Protocols for Disaster Scenarios," IEEE Internet of Things Journal, vol. 6, no. 1, pp. 878–890, Feb. 2019, doi: 10.1109/JIOT.2018.2864219.

[12] H.-Y. Lin, "Traceable Anonymous Authentication and Key Exchange Protocol for Privacy-Aware Cloud Environments," IEEE Systems Journal, vol. 13, no. 2, pp. 1608–1617, Jun. 2019, doi: 10.1109/JSYST.2018.2828022.

[13] A.-L. Peng, Y.-M. Tseng, and S.-S. Huang, "An Efficient Leakage-Resilient Authenticated Key Exchange Protocol Suitable for IoT Devices," IEEE Systems Journal, vol. 15, no. 4, pp. 5343–5354, Dec. 2021, doi: 10.1109/JSYST.2020.3038216.

[14] T.-T. Tsai, Y.-H. Chuang, Y.-M. Tseng, S.-S. Huang, and Y.-H. Hung, "A Leakage-Resilient ID-Based Authenticated Key Exchange Protocol With a Revocation Mechanism," IEEE Access, vol. 9, pp. 128633–128647, 2021, doi: 10.1109/ACCESS.2021.3112900.

[15] A. Musuroi, B. Groza, L. Popa, and P.-S. Murvay, "Fast and Efficient Group Key Exchange in Controller Area Networks (CAN)," IEEE Transactions on Vehicular Technology, vol. 70, no. 9, pp. 9385–9399, Sep. 2021, doi: 10.1109/TVT.2021.3098546.

[16] S. Li, T. Zhang, B. Yu, and K. He, "A Provably Secure and Practical PUF-Based End-to-End Mutual Authentication and Key Exchange Protocol for IoT," IEEE Sensors Journal, vol. 21, no. 4, pp. 5487–5501, Feb. 2021, doi: 10.1109/JSEN.2020.3028872.

[17] "Group Key Agreement Protocol Based on Privacy Protection and Attribute Authentication | IEEE Journals & Magazine | IEEE Xplore." Accessed: Jan. 04, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8753618.

[18] M. Malik, M. Dutta, and J. Granjal, "A Survey of Key Bootstrapping Protocols Based on Public Key Cryptography in the Internet of Things," IEEE Access, vol. 7, pp. 27443–27464, 2019, doi: 10.1109/ACCESS.2019.2900957.

[19] A. S. Sani, D. Yuan, W. Bao, and Z. Y. Dong, "A Universally Composable Key Exchange Protocol for Advanced Metering Infrastructure in the Energy Internet," IEEE Transactions on Industrial Informatics, vol. 17, no. 1, pp. 534–546, Jan. 2021, doi: 10.1109/TII.2020.2971707.

[20] "Full-Resilient Memory-Optimum Multi-Party Non-Interactive Key Exchange | IEEE Journals & Magazine | IEEE Xplore." Accessed: Jan. 04, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8950068.

[21] S. Dey and A. Hossain, "Session-Key Establishment and Authentication in a Smart Home Network Using Public Key Cryptography," IEEE Sensors Letters, vol. 3, no. 4, pp. 1–4, Apr. 2019, doi: 10.1109/LSENS.2019.2905020.

[22] X. Li, D. Yang, X. Zeng, B. Chen, and Y. Zhang, "Comments on 'Provably Secure Dynamic Id-Based Anonymous Two-Factor Authenticated Key Exchange Protocol With Extended Security Model,'" IEEE Transactions on Information Forensics and Security, vol. 14, no. 12, pp. 3344–3345, Dec. 2019, doi: 10.1109/TIFS.2018.2866304.