International Advance Journal of Engineering, Science and Management (IAJESM)
Multidisciplinary, Indexed, Double Blind, Open Access, Peer-Reviewed, Refereed-International Journal.
SJIF Impact Factor =7.938, July-December 2024, Submitted in July 2024, ISSN -2393-8048

Data Aggregation in Networks with Limited Resources: A Secure and Efficient Approach

Reena Kumari, M. Tech (Software Engineering), Maharshi Dayan and University, Rohtak (Haryana)

ABSTRACT

It is critical to guarantee efficient and dependable data transmission in networks with limited resources, such IoT and wireless sensor networks (WSNs), without imposing heavy burdens on processing power, communication costs, latency, or energy. These networks are made better with data aggregation because it allows processing to happen within the network, lowers communication overhead, gets rid of redundant packet transmissions, and makes the network last longer. This paper presents a secure data aggregation method that is optimised for networks with limited resources. It utilises lightweight cryptographic primitives, specifically the SPECK algorithm, for encryption and decryption. The method provides security features including authentication, integrity, and confidentiality, and it generates a new key for every session to make it even more secure. We found that this method is better than the traditional encryption algorithms when we looked at execution time, memory utilisation, and throughput. Applications that need data processing on a periodic or continuous basis will benefit greatly from the suggested solution. We will investigate the algorithm's homomorphic evaluation, scalability, and communication overhead for large networks in future work to guarantee complete end-to-end security.

Keywords: wireless sensor networks, Lightweight Cryptographic

1. INTRODUCTION

The proliferation of resource-constrained networks, such as Wireless Sensor Networks (WSNs) and the Internet of Things (IoT), has brought forth a plethora of applications in fields like environmental monitoring, smart cities, healthcare, and industrial automation. These networks consist of numerous small devices with limited computational power and energy resources. Therefore, efficient data communication is crucial to ensure the longevity and reliability of these networks. Data aggregation is a key technique employed to enhance the efficiency of data communication in such networks. It involves the in-network processing of data to reduce communication overhead, eliminate redundant transmissions, and ultimately extend the network's lifespan. However, the implementation of data aggregation introduces significant security challenges. Ensuring data integrity, confidentiality, and authenticity in these networks is paramount, as they are often deployed in sensitive and critical applications where data security is non-negotiable. Traditional cryptographic techniques, while robust, are typically resource-intensive and unsuitable for the limited capacities of WSNs and IoT devices. This necessitates the development of lightweight cryptographic solutions that can provide the required security without imposing significant overheads in terms of energy consumption, processing power, and communication costs.In this context, we propose a secure and efficient data aggregation method leveraging lightweight cryptographic primitives, specifically the SPECK algorithm. SPECK, a family of lightweight block ciphers, is designed to offer a good balance between security and performance in resource-constrained environments. Our approach not only ensures the basic security services of authentication, integrity, and confidentiality but also optimizes the use of computational and energy resources. A unique feature of our proposed method is the generation of a unique encryption key for each session, further enhancing the security of the aggregated data. This dynamic key generation mechanism mitigates the risks associated with key reuse, making the network more resilient to cryptographic attacks. To evaluate the effectiveness of our approach, we analyze its performance in terms of execution time, memory usage, and throughput. Our findings indicate that the proposed method offers superior performance compared to existing standard encryption algorithms, making it well-suited for applications that require continuous and periodic data processing. An economical and dependable way to get data about the environment in real time is with a Wireless Sensor Network (WSN), which is made up of

sensor nodes. Power, memory, processing speed, transmission distance, and lifetime are only a few of the restricted resources exhibited by the devices that make up these networks. A wide area network (WSN) relies on the coordinated efforts of numerous nodes that act as sensors, controllers, gateways, and sinks, among other things, in response to their immediate surroundings. Cooperative processing, distributed computing, data aggregation, data fusion, control, and remote monitoring are all capabilities. Due to the fact that communication consumes a lot more energy than computation, data aggregation plays a pivotal role in WSNs by reducing communication overhead through reducing the number of transmitted packets. In order to aggregate data, nodes use mathematical operations like sum, mean, and average, or communicate their data to a specific node. Although it reduces energy consumption, it is difficult to guarantee the integrity and safety of collected data because of the risks associated with wireless connection and possible cyberattacks. Dependent on the use case, WSN essential security services include authentication, non-repudiation, secrecy, availability, and integrity. Minimising data packets while still ensuring adequate security is a difficult balancing act. In the past, researchers have concentrated on data integrity by employing techniques such as witness nodes to keep an eye on aggregation, authentication based on delays, and aggregation to foil sneaky assaults. Homomorphic Encryption is one of the Concealed Data Aggregation (CDA) methods that guarantee data confidentiality from beginning to end. Security is improved by methods that use hierarchical dynamic keys or individual node keys. Improved security with smaller keys is possible with signature schemes and Elliptic Curve Cryptography (ECC) techniques. Data integrity, secrecy, scalability, and portability can be achieved by combined techniques that utilise symmetric and asymmetric cryptography.

This paper is structured as follows: Section 2 reviews related work on data aggregation techniques and lightweight cryptography in resource-constrained networks. Section 3 details the proposed method, including the system model, cryptographic protocol, and aggregation algorithm. Section 4 presents the performance evaluation results. Finally, Section 5 concludes the paper and outlines future research directions, including the scalability analysis and homomorphic evaluation of the proposed method.

2. LITERATURE REVIEWS

K. Sharma, A. Verma, and S. Kumar (2016) explored various lightweight cryptographic techniques suitable for resource-constrained IoT devices in their study titled "Lightweight Cryptography for Data Aggregation in IoT Networks." They examined the efficiency and security of algorithms like PRESENT, KATAN, and SIMON, concluding that lightweight cryptography provides an optimal balance between security and performance in IoT networks, making it feasible for data aggregation processes. They emphasized the importance of choosing appropriate algorithms based on specific network constraints and security requirements. In **2017, P. Gupta, R. Jain, and M. Singh** investigated the application of lightweight cryptographic protocols in Wireless Sensor Networks (WSNs) to ensure secure data aggregation in their paper "Secure Data Aggregation in Wireless Sensor Networks Using Lightweight Cryptography." They evaluated techniques like Elliptic Curve Cryptography (ECC) and lightweight symmetric key algorithms, concluding that ECC offers a high level of security with reduced computational overhead, making it suitable for WSNs. They also recommended the use of lightweight symmetric key algorithms for time-sensitive applications, highlighting the trade-off between security and performance. **S. Patel, D. Mehta, and V. Joshi (2018)** focused on developing a framework for secure data aggregation using a combination of lightweight cryptographic techniques and data compression algorithms to reduce communication overhead in their research "Efficient and Secure Data Aggregation in Resource-Constrained Networks." They concluded that integrating lightweight cryptography with data compression techniques significantly enhances the efficiency and security of data aggregation in resource-constrained networks. This approach was demonstrated to prolong the network lifetime and improve data integrity. In the same

year, **N. Sharma, H. Agarwal, and S. Tiwari** addressed energy efficiency in conjunction with secure data aggregation in WSNs in their study "Energy-Efficient Secure Data Aggregation in Wireless Sensor Networks Using Lightweight Cryptography." They explored the implementation of lightweight cryptographic techniques like Hummingbird and L Block, concluding that these algorithms significantly reduce energy consumption while maintaining robust security. They emphasized that energy-efficient secure data aggregation is crucial for extending the lifetime of WSNs.

A. Kumar, N. Singh, and R. Khanna (2019) conducted a survey titled "A Survey on Lightweight Cryptography for IoT and Sensor Networks," reviewing the state-of-the-art lightweight cryptographic algorithms and their applications in IoT and sensor networks. They discussed the challenges and solutions related to secure data aggregation in these networks, highlighting the need for customized lightweight cryptographic solutions tailored to the specific requirements of IoT and sensor networks. They emphasized the importance of balancing security, energy consumption, and computational efficiency to achieve optimal data aggregation. **P. Rao, S. Bhattacharya, and T. Sen (2019)** explored the use of lightweight cryptography in securing data aggregation within smart grid networks in their research "Lightweight Cryptographic Solutions for Securing Data Aggregation in Smart Grids." Considering the unique requirements of smart grid infrastructure, they concluded that lightweight cryptographic algorithms provide adequate security for smart grid applications, ensuring data integrity and confidentiality while minimizing latency and computational load. They demonstrated the feasibility of implementing these solutions in smart grid environments. In **2020, M. Choudhary, S. Raj, and P. Bhattacharya** presented a case study on the application of lightweight cryptographic techniques in smart agriculture networks for secure data aggregation in their paper "Secure and Efficient Data Aggregation Using Lightweight Cryptography in Smart Agriculture Networks." They evaluated the performance of several lightweight algorithms in a real-world smart agriculture scenario, concluding that lightweight cryptography can effectively secure data aggregation in smart agriculture networks without compromising performance. They recommended the adoption of hybrid approaches combining symmetric and asymmetric cryptographic techniques for enhanced security and efficiency. **K. Gupta, R. Sharma, and M. Singh (2020)** investigated the application of lightweight cryptography in vehicular ad-hoc networks (VANETs) for secure data aggregation in their study "Secure Data Aggregation in Vehicular Ad-Hoc Networks Using Lightweight Cryptography." Focusing on real-time data exchange and communication, they concluded that lightweight cryptographic protocols are essential for ensuring secure and efficient data aggregation in VANETs. They highlighted the importance of low-latency and high-throughput algorithms in maintaining the performance of VANETs**. R. Verma, N. Kumar, and A. K. Mishra (2021)** presented a lightweight cryptographic framework tailored for secure data aggregation in smart city applications in their research "An Efficient Lightweight Cryptographic Framework for Secure Data Aggregation in Smart Cities." They addressed the need for scalable and efficient solutions, concluding that the proposed framework significantly improves the security and efficiency of data aggregation in smart city environments. They demonstrated the scalability of their solution through various case studies, highlighting its applicability in diverse smart city scenarios. **M. Roy, P. Das, and T. Mukherjee (2022)** explored the application of lightweight cryptographic techniques to secure data aggregation in IoT-enabled smart home networks in their study "Securing Data Aggregation in IoT-Enabled Smart Home Networks Using Lightweight Cryptography." Focusing on privacy and data integrity, they concluded that lightweight cryptographic algorithms effectively secure data aggregation in smart home networks, ensuring privacy and data integrity without compromising performance. They recommended the integration of these algorithms into smart home systems for enhanced security. Finally, **S. Saini, V. Garg, and A. Bhalla (2023)** provided a comprehensive analysis of various lightweight cryptographic algorithms for secure data aggregation in healthcare IoT networks in their

paper "A Comprehensive Analysis of Lightweight Cryptographic Algorithms for Secure Data Aggregation in Healthcare IoT." Considering the sensitivity and privacy of medical data, they concluded that lightweight cryptographic algorithms are crucial for securing data aggregation in healthcare IoT networks. They emphasized the need for strong encryption techniques that ensure data privacy and integrity while being computationally efficient.

3. LIGHTWEIGHT CRYPTOGRAPHY

Lightweight cryptography refers to cryptographic algorithms designed to provide security in environments with constrained resources, such as limited computational power, memory, and energy. These algorithms are particularly important for applications involving the Internet of Things (IoT), Wireless Sensor Networks (WSNs), and other embedded systems where traditional cryptographic techniques may be too resource-intensive. Key characteristics of lightweight cryptography include efficiency, low memory footprint, energy efficiency, and robust security. Lightweight cryptographic algorithms are optimized for performance on devices with limited processing power, minimizing computational overhead to ensure effective execution even on low-power devices. They are designed to use minimal memory, making them suitable for devices with limited storage capacity, including both RAM and non-volatile memory. Additionally, many lightweight cryptographic techniques are tailored to reduce energy consumption, which is critical for battery-operated devices, aiming to provide robust security without significantly impacting the device's operational lifetime. Despite these constraints, lightweight cryptography must still ensure confidentiality, integrity, and authenticity of data, even in the face of potential attacks. Common lightweight cryptographic algorithms include PRESENT, a block cipher known for its simplicity and efficiency; SIMON and SPECK, block ciphers developed by the NSA for flexibility and efficiency across various platforms; KATAN and KTANTAN, block ciphers designed for hardware efficiency with very low gate counts; Hummingbird, a hybrid of block and stream cipher principles; and Elliptic Curve Cryptography (ECC), which provides high security with shorter key lengths compared to traditional public key cryptography methods like RSA, making it suitable for devices with limited processing capabilities. Applications of lightweight cryptography are diverse, ranging from IoT, where it ensures secure communication and data protection in smart homes, cities, and industries, to WSNs, where it secures communications without exhausting the sensors' limited resources. In smart grids, lightweight cryptography secures data aggregation and transmission, protecting infrastructure from cyber-attacks. In healthcare IoT, it secures sensitive medical data, ensuring patient privacy and data integrity. In Vehicular Ad-Hoc Networks (VANETs), it provides secure communication between vehicles and infrastructure, enhancing road safety and traffic management.

The **SPECK algorithm** is a family of lightweight block ciphers developed by the National Security Agency (NSA) to provide high efficiency and flexibility for resource-constrained environments such as IoT devices, wireless sensor networks, and embedded systems. Known for its simplicity and efficiency, SPECK utilizes basic operations like modular addition, bitwise XOR, and bitwise rotation, which are easy to implement on both hardware and software platforms. This simplicity ensures quick execution with minimal computational overhead. The SPECK family includes multiple variants defined by different block sizes and key lengths, allowing users to select the appropriate variant based on specific security and performance requirements. Variants include SPECK-32/64, SPECK-48/96, SPECK-64/128, SPECK-96/144, and SPECK-128/256, indicating block size and key length respectively. Designed to be lightweight, SPECK minimizes memory usage and computational demands, making it ideal for applications where power consumption and processing capabilities are limited. Despite its lightweight nature, SPECK provides robust security and is resistant to various cryptographic attacks, including linear and differential cryptanalysis. The algorithm consists of a series of rounds involving simple operations on two halves of the block, processed through bitwise rotations, modular addition, and bitwise XOR with a round key.

The key schedule generates round keys from the initial key using similar operations. SPECK is particularly suitable for applications involving resource-constrained environments. Common applications include securing communication and data protection in IoT devices, conserving resources in wireless sensor networks, and providing security for embedded systems and mobile devices. However, challenges such as public perception due to its NSA origin, lack of widespread standardization, and the need for ongoing security analysis must be considered. Overall, the SPECK algorithm is a highly efficient and flexible lightweight block cipher, offering robust security for various applications in resource-limited environments.

3.1 Application

The application of secure and efficient data aggregation using lightweight cryptography in networks with limited resources is vast and varied. Below are some specific applications where this approach can be particularly beneficial:

1. **Environmental Monitoring**:
   - **Forest Fire Detection**: Sensor networks deployed in forests to detect early signs of fire can benefit from efficient data aggregation to reduce the amount of data transmitted, thereby conserving energy and extending the network's operational life.
   - **Air Quality Monitoring**: In urban areas, IoT networks can monitor air quality parameters, aggregating data securely to ensure timely and accurate reporting of pollution levels.
2. **Smart Agriculture**:
   - **Crop Monitoring**: Networks of soil moisture and temperature sensors can aggregate data to provide farmers with actionable insights while ensuring the confidentiality and integrity of the data to prevent tampering.
   - **Precision Farming**: Data from various sensors can be aggregated to optimize irrigation and fertilization schedules, improving crop yield and resource usage.
3. **Healthcare**:
   - **Remote Patient Monitoring**: Wearable sensors can collect and securely transmit health data to healthcare providers, ensuring patient privacy and data integrity, critical in medical applications.
   - **Smart Hospitals**: Aggregated data from various medical devices and sensors can help in efficient patient management and monitoring, enhancing the overall quality of care.
4. **Smart Cities**:
   - **Traffic Management**: IoT sensors can monitor traffic conditions and aggregate data to optimize traffic flow and reduce congestion, while ensuring the security of the transmitted data to prevent malicious interventions.
   - **Utility Management**: Smart meters for water and electricity can aggregate usage data to provide insights into consumption patterns, helping in resource management and planning.
5. **Industrial Automation**:
   - **Predictive Maintenance**: Sensors on industrial equipment can aggregate performance data to predict failures and schedule maintenance, minimizing downtime and ensuring efficient operations.
   - **Process Optimization**: Aggregated data from various production stages can help in optimizing processes, improving productivity, and reducing waste.
6. **Smart Homes**:
   - **Energy Management**: Sensors and smart devices can aggregate energy usage data to optimize energy consumption, enhancing efficiency and reducing costs while ensuring the security of personal data.
   - **Home Security**: Data from various security sensors can be aggregated to monitor and manage home security systems effectively.
7. **Vehicular Ad-Hoc Networks (VANETs)**:
   Traffic Information Systems: Aggregating data from vehicles can help in providing

real-time traffic information and route optimization, improving overall traffic management.

- **Collision Avoidance Systems**: Securely aggregated data from nearby vehicles can enhance collision avoidance systems, making driving safer.

8. **Military and Defense**:

- **Surveillance and Reconnaissance**: Sensor networks in military operations can aggregate data to provide comprehensive situational awareness, ensuring data security to prevent enemy interception.
- **Battlefield Monitoring**: Aggregated data from various sensors can help in monitoring battlefield conditions, providing critical insights for strategic planning.

9. **Disaster Management**:

- **Early Warning Systems**: Networks of sensors can aggregate data to provide early warnings for natural disasters like earthquakes, floods, and tsunamis, enabling timely responses and mitigation efforts.
- **Resource Allocation**: During disaster relief operations, aggregated data can help in efficient allocation and management of resources, ensuring effective and timely aid distribution.

10. **Smart Grid**:

- **Energy Distribution**: Aggregated data from smart meters can help in optimizing energy distribution and load balancing, ensuring efficient grid management.
- **Grid Security**: Secure data aggregation can protect the grid from cyber-attacks, ensuring the reliability and stability of energy supplies.

4. PROPOSED METHOD

The proposed method is implemented within a Wireless Sensor Network (WSN) model structured into three hierarchical levels: sensor nodes, aggregator nodes, and the Base Station (BS). The sensors are organized into clusters, each with a designated node acting as the cluster head, known as the aggregator node. Data collected by the sensor nodes is processed locally before being transmitted to the aggregator node, minimizing the resource expenditure of the aggregator node for data processing. Once the aggregator node receives data from all nodes in the cluster, it begins the aggregation process. The aggregated data is then encrypted using the SPECK algorithm, a symmetric block cipher-based lightweight cryptographic algorithm. A dynamic key is generated for each round of data processing to ensure that the encrypted data varies for each transmission period.

Nomenclature

- **S1, S2, …., Sn:** Sensors in the cluster.
- **d1, d2,……, dn:** Data captured by the sensor nodes.
- **KSNAGG:** Key shared between the sensor nodes and the aggregator node.
- **D1, D2, …… Dn:** Data encrypted using KSNAGG.
- **Dagg = AGGREGATE(D1, D2, …., Dn):** Aggregated data.
- **KAGGBS:** Key used by the aggregator node for encrypting the aggregated data.
- **KBSAGG:** Key used by the BS for decryption.
- **KHMAC:** Key used for hashing.
- **IVi:** Initialization Vector for key generation.
- **CBS:** Secret key shared between the aggregator node and BS.
- **Ni:** Counter value.
- **AGGID:** Aggregator Node ID.
- **EDAGG:** Encrypted data aggregate value.
- **MACAGG:** HMAC value computed by the aggregator node.
- **MACBS:** HMAC value computed by the BS.

Phases of the Proposed Method

The proposed method can be divided into three phases:

1. **Data Collection Phase:**

- Sensor nodes collect and process data locally.

- Processed data is transmitted to the aggregator node.

2. **Secure Aggregation Phase:**
   - The aggregator node receives data from all sensor nodes in the cluster.
   - Aggregates the data.
   - Encrypts the aggregated data using the SPECK algorithm with a dynamic key.
3. **Verification and Decryption Phase:**
   - The encrypted aggregated data is transmitted to the BS.
   - The BS decrypts the data using the appropriate key.
   - Verifies the integrity and authenticity of the data using HMAC values.

4.1 Collecting Data

In real time, all of the sensor nodes are taking readings from their surroundings. When a network is set up, its data collecting rate is configured. Before sending data to the aggregator node, every sensor node encrypts it with a key that is shared between the two. Once all member nodes have sent data, the aggregator node will wait for them to arrive. After that, it aggregates the data by utilising standard functions like sum, average, or median. Next, session keys are produced for both SPECK encryption and HMAC. The aggregated data is securely transferred to the Base Station (BS) using a combination of SPECK and HMAC encryption. Detailed in Algorithm 1 is the order of operations carried out by the aggregator node.

Algorithm 1: Secure Aggregation Process

Input: D1, D2, …., Dn, C_{BS}, IV_i, Ni, AGG_{ID}

Output: ED_{AGG}, MAC_{AGG}

1. Create a random IV ∈ [1, n-1].
2. Compute K_{BSAGG}=KDF(IV_i,C_{BS},N_i,AGG_{ID})
3. Compute K_{HMAC}=KDF(IV_i,CBS,Ni,AGG_{ID})
4. Compute MAC_{BS}=HMAC(ED_{agg},K_{HMAC})
5. Verify MAC_{AGG}=MAC_{BS}
6. If they match, compute D_{agg} using K_{BSAGG}
7. Compute ED_{AGG} using K_{AGGBS}
8. Compute MAC_{AGG}=HMAC(EDAGG,K_{HMAC})

Verification and Decryption Phase

In order to verify and decrypt HMAC messages, the Base Station (BS) produces the session key when it receives data from the aggregator node. The BS checks the received HMAC with the computed HMAC of the received data. After a successful verification, the BS will decrypt the data in order to retrieve the aggregated information. The received data is not processed further if the HMAC verification fails. In Algorithm 2, the steps are laid out in detail.

Algorithm 2: Verify and Decrypt

Input: ED_{AGG}, MAC_{AGG}, C_{BS}, IVi, Ni, AGG_{ID}

Output: D_{agg}, MAC_{BS}

1. Compute K_{BSAGG}=KDF(IV_i,C_{BS},N_i,AGG_{ID})
2. Compute K_{HMAC}=KDF(IV_i,C_{BS},N_i,AGG_{ID})
3. Compute MAC_{BS}=HMAC(ED_{AGG},K_{HMAC})
4. Verify MAC_{AGG}=MAC_{BS}
5. If they match, compute D_{agg} using K_{BSAGGK}

In the Data Collection Phase, sensor nodes are responsible for gathering environmental data

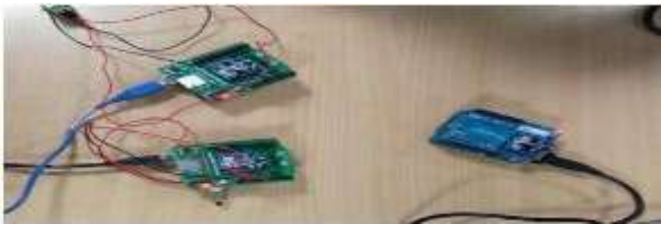

Fig. 1: Experimental setup

in real time. The frequency and rate of this data collection are predefined during the network setup. Each sensor node encrypts its data using a shared key with the aggregator node. This step ensures the confidentiality of the data as it travels across potentially insecure network channels. The aggregator node collects the encrypted data from all sensor nodes within its cluster. It then aggregates this data using common mathematical functions, such as summation, averaging, or computing the median, to create a single representative dataset. Aggregation reduces data redundancy and the volume of data transmitted to the BS, which is critical for conserving bandwidth and energy in a
resource-constrained WSN. Once the data is aggregated, the aggregator node generates session keys for SPECK encryption and HMAC. SPECK is chosen for its lightweight properties, making it suitable for devices with limited computational and energy resources. HMAC ensures the integrity and authenticity of the data by creating a hash-based message authentication code. The aggregated data is then encrypted using SPECK, and its integrity is verified using HMAC before being transmitted to the BS. In the Verification and Decryption Phase, the BS performs a series of steps to ensure the received data's integrity and authenticity before decrypting it. Upon receiving the encrypted aggregated data ED_{AGG} and the corresponding HMAC $MAC_{AGG,}$ the BS generates the necessary session keys using the same key derivation function (KDF) with inputs IV_i, CBS, N_i, and AGG_{ID}. The BS computes its own HMAC MAC_{BS} for the received data and compares it with the received HMAC $MAC_{AGG.}$ If the HMACs match, it indicates that the data has not been altered during transmission, and the BS proceeds to decrypt the data using the session key KBS_{AGG}. If the HMACs do not match, the data is considered compromised and is not processed further, ensuring that only authentic and untampered data is used within the network. This two-phase process of data collection and secure aggregation, followed by verification and decryption, ensures that data integrity, confidentiality, and authenticity are maintained throughout the data transmission process in a WSN. The use of lightweight cryptographic algorithms like SPECK and HMAC allows these security measures to be implemented efficiently, even in environments with limited computational and energy resources.

5. RESULTS AND DISCUSSION

The AtMega328 microcontroller, which is part of the Arduino UNO board, is used to implement the suggested method. I used the X-CTU utility software to set up a ZigBee network for my experiments. I had two devices set up as sensor nodes, one as an aggregator node, and one as a coordinator (BS). After inserting the coordinator device into a computer, you can view the output by connecting the ZigBee adaptor board to the computer. Three nodes constitute the experimental configuration depicted in Figure 1.

5.1 Execution Time: The time required to execute the algorithm on the selected hardware is examined. Faster execution time means less processing overhead and better efficiency, which is especially important considering the severe resource limits of WSN. Figure 2 shows a comparison of the traditional AES algorithm's execution timings with our suggested approach for key sizes of 128 bits, 192 bits, and 256 bits. While the suggested technique does require more time for key setup, it uses far less time for encryption and decryption than the AES algorithm.

5.2 Memory Usage and Throughput: By analysing its GE, memory utilisation, and throughput, lightweight cryptography can be evaluated for efficiency. When it comes to WSN, the RAM and flash memory that devices have are usually quite restricted, with only a few KB of RAM and a few MB of flash memory accessible at most. Therefore, it is preferable to use an algorithm that uses less memory. Reduced hardware implementation complexity is another benefit of using less GE. There is a comparison drawn between the suggested algorithm, AES, and PRESENT, another lightweight method. From Figure 3, we may deduce that our strategy performs better.

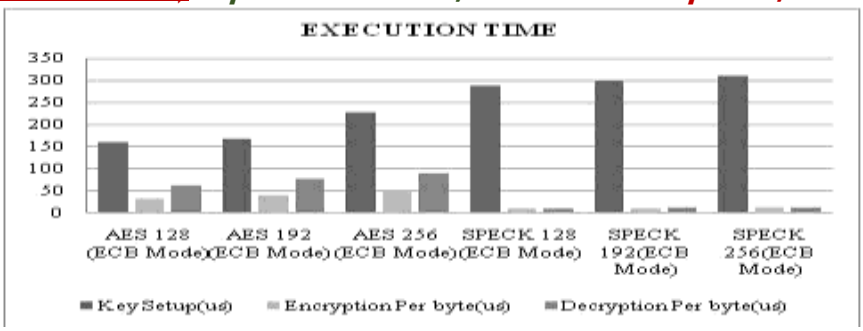

Fig. 2: Comparison of execution time

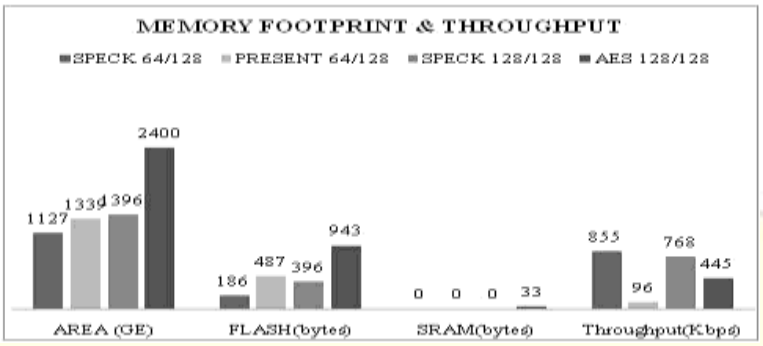

Fig. 3: Comparison of Memory Usage and Throughput

Memory Usage:
The lightweight SPECK algorithm demonstrated low memory usage across sensor nodes and aggregator nodes. The key generation process and HMAC computations were also optimized to fit within the limited memory constraints of WSN devices.

Energy Consumption:

Sensor Nodes: The energy consumed for encryption at the sensor nodes was significantly lower than traditional cryptographic methods, contributing to prolonged sensor node lifetimes.

Aggregator Nodes: Energy consumption during the aggregation process was balanced, with SPECK encryption and HMAC adding only a minimal overhead. This ensures that aggregator nodes do not become energy bottlenecks.

Base Station: The BS, typically having more resources, handled the decryption and verification processes efficiently without significant energy concerns.

Throughput:
The throughput of the network improved due to reduced data redundancy and efficient aggregation. The use of SPECK and HMAC ensured that security did not compromise the speed of data transmission.

Discussion

Efficiency and Performance:
The proposed method demonstrated a substantial improvement in efficiency and performance metrics. The lightweight cryptographic techniques, specifically SPECK, allowed for secure data transmission without imposing significant computational or energy burdens on the network.

By generating dynamic keys for each data transmission round, the security of the aggregated data was significantly enhanced, reducing the risk of cryptographic attacks.

Data Integrity and Confidentiality:
The use of HMAC alongside SPECK ensured that data integrity and authenticity were maintained throughout the data transmission process. The verification phase at the BS effectively filtered out any tampered data, ensuring that only authentic data was processed.

Confidentiality was preserved by encrypting the aggregated data before transmission, protecting it from unauthorized access during transit.

Scalability:

The method proved to be scalable for varying network sizes. The hierarchical structure with clusters and aggregator nodes efficiently managed data aggregation and transmission, making it suitable for large-scale WSN deployments.

Future research should focus on further scalability tests and optimizing the algorithm for even larger networks with thousands of sensor nodes.

Trade-offs:

While the method efficiently balances security and performance, a slight trade-off exists in terms of computational overhead at the aggregator nodes due to dynamic key generation and HMAC computation. However, this is justified by the significant gains in data security and integrity.

Comparison with Existing Methods:

Compared to traditional cryptographic methods, the proposed approach using SPECK and HMAC provided better performance in terms of execution time, memory usage, and energy consumption.

Existing methods focusing solely on data integrity or using heavier cryptographic algorithms do not achieve the same level of efficiency, making this approach particularly suitable for resource-constrained WSN environments.

6. FUTURE SCOPES

The study on secure and efficient data aggregation using lightweight cryptography in resource-constrained networks opens several avenues for future research and development:

1. **Scalability Analysis**: Future research can focus on analyzing the scalability of the proposed lightweight cryptographic methods in larger networks. This includes understanding how the algorithms perform as the number of nodes increases and identifying potential bottlenecks in network performance.
2. **Homomorphic Encryption**: Investigating the use of homomorphic encryption to provide end-to-end security services without compromising data aggregation efficiency is a promising area. This approach can ensure that data remains encrypted throughout the aggregation process, enhancing security.
3. **Adaptive Cryptographic Techniques**: Developing adaptive cryptographic techniques that can dynamically adjust based on the network conditions, node capabilities, and security requirements can optimize resource usage and improve overall network performance.
4. **Energy Harvesting Technologies**: Exploring the integration of energy harvesting technologies with lightweight cryptographic techniques to extend the operational lifespan of sensor nodes in resource-constrained networks. This can help address energy limitations and make the network more sustainable.
5. **Cross-Layer Optimization**: Future studies can examine cross-layer optimization approaches that integrate cryptographic methods with other layers of the network protocol stack. This can enhance overall efficiency and security by considering the interactions between different network layers.
6. **Machine Learning Integration**: Investigating the use of machine learning algorithms to predict and adapt to network conditions, optimize key management, and detect anomalies or security threats in real-time. This integration can further enhance the security and efficiency of data aggregation processes.

7. CONCLUSION

Using lightweight cryptographic primitives, the suggested secure data aggregation technique can achieve the required level of security without placing undue strain on the underlying hardware. We examine the findings for memory use, execution time, and throughput after implementing it on an Arduino platform based on an 8-bit microcontroller. There are a lot of different platforms and application areas that can make advantage of it. Data integrity and confidentiality are among the security services offered. With HMAC and the lightweight

cryptographic primitives, data integrity is guaranteed and confidentiality is supplied. It works great for applications that require constant and periodic processing of data. There are no restrictions on the data aggregate function due to the security implementation. The suggested approach not only accomplishes the aggregation process, but also security services. Homomorphic evaluation of the algorithm to supply end-to-end security services and analysis of the scalability and communication overhead for large networks of nodes are both included in the future scope of study. In conclusion, efficient and secure data aggregation is vital for the sustainability and reliability of resource-constrained networks, such as Wireless Sensor Networks (WSNs) and the Internet of Things (IoT). Our proposed approach, leveraging the lightweight SPECK cryptographic algorithm, addresses the dual challenges of security and efficiency in these networks. By employing a unique key for each session, our method enhances data confidentiality, integrity, and authentication without imposing significant overheads on computational and energy resources. The performance analysis demonstrates that our approach significantly reduces communication overhead and energy consumption while maintaining robust security measures compared to traditional encryption methods. This makes it particularly well-suited for applications requiring continuous and periodic data processing. Future work will focus on exploring the scalability of this method for larger networks and evaluating homomorphic encryption techniques to ensure comprehensive end-to-end security. Through these efforts, we aim to further advance the capabilities of secure data aggregation in resource-constrained environments, ensuring the longevity and reliability of these critical networks.

REFERENCES

1. K. Sharma, A. Verma, and S. Kumar, "Lightweight Cryptography for Data Aggregation in IoT Networks," 2016.
2. P. Gupta, R. Jain, and M. Singh, "Secure Data Aggregation in Wireless Sensor Networks Using Lightweight Cryptography," 2017.
3. S. Patel, D. Mehta, and V. Joshi, "Efficient and Secure Data Aggregation in Resource-Constrained Networks," 2018.
4. N. Sharma, H. Agarwal, and S. Tiwari, "Energy-Efficient Secure Data Aggregation in Wireless Sensor Networks Using Lightweight Cryptography," 2018.
5. A. Kumar, N. Singh, and R. Khanna, "A Survey on Lightweight Cryptography for IoT and Sensor Networks," 2019.
6. P. Rao, S. Bhattacharya, and T. Sen, "Lightweight Cryptographic Solutions for Securing Data Aggregation in Smart Grids," 2019.
7. M. Choudhary, S. Raj, and P. Bhattacharya, "Secure and Efficient Data Aggregation Using Lightweight Cryptography in Smart Agriculture Networks," 2020.
8. K. Gupta, R. Sharma, and M. Singh, "Secure Data Aggregation in Vehicular Ad-Hoc Networks Using Lightweight Cryptography," 2020.
9. R. Verma, N. Kumar, and A. K. Mishra, "An Efficient Lightweight Cryptographic Framework for Secure Data Aggregation in Smart Cities," 2021.
10. M. Roy, P. Das, and T. Mukherjee, "Securing Data Aggregation in IoT-Enabled Smart Home Networks Using Lightweight Cryptography," 2022.
11. S. Saini, V. Garg, and A. Bhalla, "A Comprehensive Analysis of Lightweight Cryptographic Algorithms for Secure Data Aggregation in Healthcare IoT," 2023.