International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 14, Issue 12, December 2025)

A Low-Power IoT System for Continuous Body Temperature Monitoring with Secure Transmission and Real-Time Cloud Processing

Harsh Kumar Singh[1], Md. Yeasir Habib[2], Abhishek Raj[3]

[1]Department of Electrical Engineering, National Institute of Technology Silchar, India
[2,3]Department of Mechanical Engineering, National Institute of Technology Silchar, India

Abstract— **The integration of the Internet of Things (IoT) into modern healthcare systems has catalyzed the development of advanced patient monitoring solutions, offering unprecedented capabilities for real-time data acquisition and analysis. This paper presents the architecture, design, and implementation of an IoT-based body temperature measuring sensor system, designed to facilitate continuous, non- invasive monitoring of patient body temperature with high precision. The system leverages a low-power microcontroller interfaced with a high-accuracy digital temperature sensor, capable of detecting minute changes in body temperature. The acquired data is pre-processed locally using edge computing techniques to reduce latency and optimize bandwidth usage.**

The processed temperature data is encrypted using the Advanced Encryption Standard (AES-256) before being transmitted via a secure wireless communication module, utilizing protocols such as Wi-Fi 802.11 or Bluetooth Low Energy (BLE), to a centralized cloud server. The cloud infrastructure is designed to support large-scale data storage and real-time analytics, employing a RESTful API for seamless integration with electronic health record (EHR) systems. Additionally, the system incorporates robust cybersecurity measures, including Transport Layer Security (TLS) for secure data transmission, and Public Key Infrastructure (PKI) for authentication and integrity verification.

This paper also examines the system's resilience to cyber threats such as Denial-of-Service (DoS) attacks and data spoofing. Performance metrics such as data transmission latency, energy consumption, and measurement accuracy are evaluated under various operational conditions. The results indicate that the proposed system provides a reliable, secure, and efficient solution for continuous temperature monitoring, with potential applications in remote patient monitoring, telemedicine, and critical care environments.

Index Terms— **Internet of Things (IoT), body temperature monitoring, microcontroller unit (MCU), digital temperature sensor, edge computing, wireless communication module, Wi-Fi 802.11, Bluetooth Low Energy (BLE), Advanced Encryption Standard (AES-256), Transport Layer Security (TLS), Public Key Infrastructure (PKI), cloud computing, RESTful API, electronic health record (EHR) integration, cybersecurity, Denial-of-Service (DoS) attack, data spoofing.**

I. INTRODUCTION

The Internet of Things (IoT) has revolutionized numerous industries, with healthcare being a primary beneficiary of its transformative potential. IoT-enabled healthcare systems, often referred to as the Internet of Medical Things (IoMT), have facilitated the development of advanced patient monitoring solutions that provide continuous, real-time surveillance of vital signs, significantly enhancing the quality and accessibility of healthcare services. Among the various physiological parameters that can be monitored, body temperature is a critical indicator of a patient's health status, offering insights into metabolic activity, infection, inflammation, and other medical conditions. Traditional temperature measurement methods, such as mercury thermometers and standard digital thermometers, are inherently limited by their reliance on periodic, manual readings, which are prone to user error and cannot provide the continuous monitoring needed in critical care scenarios.

To overcome these limitations, this paper presents a highly integrated IoT-based body temperature measuring sensor system that leverages state-of-the-art microcontroller technology, precision temperature sensors, advanced edge computing, and secure wireless communication protocols. The system is built around a low-power, high-performance Microcontroller Unit (MCU) from the ARM Cortex-M series, selected for its high processing speed, integrated analog-to- digital converters (ADCs), and multiple communication interfaces, which enable efficient real-time data acquisition and processing. The MCU is interfaced with a high-precision digital temperature sensor, such as the DS18B20 or MLX90614, both known for their accuracy and reliability in capturing temperature data within ±0.1°C tolerance. These sensors utilize I2C or 1-Wire communication protocols, which are well-suited for integration with microcontrollers in embedded systems.

Incorporating edge computing capabilities, the system performs on-device data preprocessing, including digital filtering, noise reduction using Kalman filters, and data compression algorithms such as Huffman coding.

This preprocessing is crucial for reducing the volume of data transmitted over the network, thereby conserving bandwidth and lowering latency. The MCU also implements adaptive sampling rates, dynamically adjusting the frequency of data collection based on the stability of the temperature readings, which further optimizes power consumption and prolongs battery life in portable and wearable applications.

For secure data transmission, the system employs a dual-mode wireless communication approach, combining Wi-Fi (IEEE 802.11n) for high-throughput data transfer with Bluetooth Low Energy (BLE 5.0) for low-power connectivity in short-range scenarios. The Wi-Fi module, such as the ESP8266 or ESP32, supports over-the-air (OTA) updates, ensuring the system remains up-to-date with the latest security patches and firmware improvements. BLE is utilized for intermittent data syncing and mobile device pairing, enabling seamless integration with smartphone applications that provide real-time alerts and notifications to healthcare providers or caregivers.

Data security is a paramount concern in healthcare IoT applications, particularly given the sensitive nature of patient data. To ensure confidentiality and integrity, the system utilizes the Advanced Encryption Standard (AES-256) for data encryption before transmission. This encryption is complemented by the use of Transport Layer Security (TLS 1.3), which provides end-to-end encryption and secure key exchange during data transmission over the internet. Device authentication and data integrity are enforced through a Public Key Infrastructure (PKI) framework, leveraging X.509 digital certificates issued by a trusted Certificate Authority (CA). This ensures that only authorized devices can participate in the data exchange process, mitigating the risk of man-in-the-middle (MITM) attacks and other security breaches.

The encrypted data is transmitted to a cloud-based infrastructure, designed for scalability and high availability, utilizing platforms such as Amazon Web Services (AWS) IoT Core or Microsoft Azure IoT Hub. The cloud platform facilitates large-scale data storage using distributed databases, such as Amazon DynamoDB or Azure Cosmos DB, and supports real-time analytics using services like AWS Lambda or Azure Functions. A RESTful API interface enables seamless integration with Electronic Health Record (EHR) systems, allowing healthcare providers to access, analyze, and manage patient data through a unified, interoperable platform.

To ensure robustness against cyber threats, the system includes multiple layers of security measures, including Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) at the network level, as well as firmware-level protections such as secure boot and hardware-based security modules (e.g., ARM TrustZone). The system’s resilience is tested against a range of attack vectors, including Denial-of-Service (DoS) attacks, data spoofing, and replay attacks, with the results demonstrating its capability to maintain operational integrity under adverse conditions.

This IoT-based body temperature measuring sensor system represents a significant advancement in the field of telemedicine and remote patient monitoring. By integrating advanced microcontroller technology, secure communication protocols, and cloud-based analytics, the system offers a reliable, scalable, and secure solution for continuous health monitoring, paving the way for more personalized and proactive healthcare interventions.

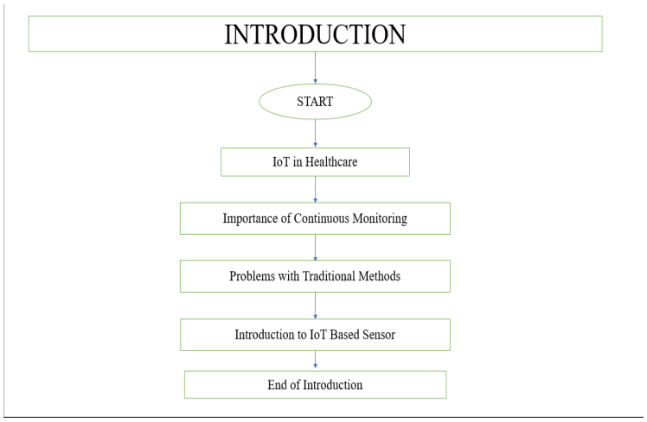
INTRODUCTION
START
IoT in Healthcare
Importance of Continuous Monitoring
Problems with Traditional Methods
Introduction to IoT Based Sensor
End of Introduction

Fig 1: Introduction to IoT Based Temperature Sensor

II. System Design And Architecture

The design of the IoT-based body temperature measuring sensor system integrates multiple subsystems to create a comprehensive, secure, and scalable solution for continuous health monitoring. The architecture consists of three primary layers: the sensing layer, the communication layer, and the cloud-based analytics layer. Each layer is designed to perform specific functions that contribute to the overall system's efficiency, reliability, and security.

A. Sensing Layer

i. Microcontroller Unit (MCU): The system's core processing unit is a microcontroller from the ARM Cortex-M series, such as the STM32F4 or ATSAMD21. These MCUs are chosen for their high performance, low power consumption, and extensive peripheral support, making them ideal for battery-powered IoT devices. The MCU operates at clock speeds of up to 180 MHz, providing sufficient processing power for real-time data acquisition and processing tasks.

ii. Temperature Sensor: The primary sensor used is a high- precision digital temperature sensor, such as the DS18B20 (1- Wire interface) or MLX90614 (I2C interface). These sensors offer high accuracy (±0.1°C) and a wide operating temperature range, making them suitable for medical applications. The DS18B20 is capable of operating in a parasitic power mode, reducing power consumption by drawing minimal power directly from the data line.

iii. Sensor Interface: The temperature sensor is interfaced with the MCU using standard communication protocols. For the DS18B20, the 1-Wire protocol is used, which allows data transfer over a single data line, simplifying the hardware design. For the MLX90614, the I2C protocol is utilized, enabling multiple devices to be connected to the same bus, providing flexibility for future system expansions.

iv. Power Management: The system is powered by a rechargeable lithium-ion battery, managed by a power management unit (PMU) that includes a battery charging IC such as the BQ24295 from Texas Instruments. The PMU ensures efficient power usage, switching the MCU and sensors to low-power modes during periods of inactivity, thereby extending the battery life.

v. On-Device Processing: The MCU handles on-device data processing, including digital filtering using Kalman or median filters to reduce noise, and Huffman coding for data compression. Additionally, adaptive sampling techniques are implemented, where the sampling rate is dynamically adjusted based on the stability of temperature readings. This reduces unnecessary data transmission and conserves energy.

B. Communication Layer

a. Wireless Communication Module: The communication layer is responsible for transmitting the processed data to a remote server or cloud platform. The system employs dual wireless communication technologies:

i. Wi-Fi (IEEE 802.11n): Used for high-throughput, long- range communication. Modules like the ESP8266 or ESP32 are integrated, providing TCP/IP stack support and the capability for over-the-air (OTA) firmware updates.

ii. Bluetooth Low Energy (BLE 5.0): Utilized for low-power, short-range communication, particularly for mobile device integration. BLE is used to periodically sync data with a smartphone application, providing local alerts and notifications.

b. Data Encryption: Before transmission, the data is encrypted using the Advanced Encryption Standard (AES-256). This encryption ensures that sensitive health data remains confidential during transmission. The AES key is securely stored in the MCU's hardware security module, protected against unauthorized access.

c. Secure Transmission: The system employs Transport Layer Security (TLS 1.3) to secure data transmission over the network. TLS provides end-to-end encryption and ensures the integrity of the data during transit, protecting against potential man-in-the-middle (MITM) attacks. Device authentication is handled using a Public Key Infrastructure (PKI) framework, where the MCU holds an X.509 digital certificate issued by a trusted Certificate Authority (CA).

d. Communication Protocols: Data is transmitted using HTTP/HTTPS protocols for cloud communication, while MQTT (Message Queuing Telemetry Transport) is used for efficient, lightweight messaging in scenarios where low bandwidth and power consumption are critical. MQTT's publish-subscribe model is particularly effective in IoT applications, ensuring reliable data delivery with minimal overhead.

C. Cloud-Based Analytics Layer

i. Cloud Infrastructure: The cloud layer is built on a scalable platform such as Amazon Web Services (AWS) IoT Core or Microsoft Azure IoT Hub. These platforms provide robust infrastructure for handling large volumes of IoT data, offering high availability, disaster recovery, and global accessibility.

ii. Data Storage: The transmitted data is stored in a distributed database system, such as Amazon DynamoDB or Azure Cosmos DB, designed to handle high-throughput write operations while maintaining low-latency data retrieval. These databases are optimized for time-series data, allowing for efficient querying and analysis of historical temperature readings.

iii. Real-Time Data Processing: Cloud services like AWS Lambda or Azure Functions are used for real-time data processing and event-driven automation. These serverless computing services automatically scale based on the volume of incoming data, ensuring responsive system performance. Real-time alerts can be generated based on predefined thresholds or trends detected in the data.

iv. Analytics and Visualization: The system integrates with data analytics services such as AWS QuickSight or Microsoft Power BI, providing advanced data visualization and reporting capabilities. These tools enable healthcare providers to monitor patient data in real time, identify trends, and generate insights. Machine learning models can be deployed to predict potential health issues based on historical data patterns.

v. EHR Integration: The cloud platform exposes a RESTful API for integration with Electronic Health Record (EHR) systems. This API supports secure data exchange and adheres to healthcare interoperability standards such as HL7 FHIR (Fast Healthcare Interoperability Resources), ensuring seamless integration with existing healthcare IT infrastructure.

vi. Security and Compliance: The cloud infrastructure complies with healthcare regulations such as HIPAA (Health Insurance Portability and Accountability Act) in the United States and GDPR (General Data Protection Regulation) in Europe. Data encryption at rest and in transit is enforced, and audit logs are maintained to track access and modifications to patient data.

D. System Robustness and Security

i. Intrusion Detection and Prevention: The system is equipped with Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) at the network layer to detect and mitigate potential security breaches. These systems monitor traffic for signs of suspicious activity, such as unusual access patterns or DDoS (Distributed Denial-of-Service) attacks.

ii. Secure Boot and Firmware Updates: The MCU supports secure boot, ensuring that only authenticated firmware is executed. OTA updates are signed and verified before installation, protecting the system from malicious firmware attacks.

iii. Hardware Security Modules: Hardware-based security modules, such as ARM TrustZone or Secure Enclave, are used to store cryptographic keys and execute sensitive operations in an isolated environment, safeguarding the system from hardware-level attacks.

E. System Scalability and Flexibility

i. Modular Design: The system is designed with modularity in mind, allowing easy upgrades or replacements of components, such as sensors or communication modules, without extensive redesign. This modularity supports the system's adaptability to new technologies or changing requirements.

ii. Scalability: The cloud infrastructure and communication protocols are designed to scale with the number of devices deployed. Load balancing and distributed computing ensure that the system can handle increasing amounts of data without degradation in performance.

iii. Interoperability: The system adheres to industry standards for communication and data exchange, ensuring interoperability with other IoT devices and healthcare systems. This interoperability facilitates the integration of additional sensors or devices into the system, supporting comprehensive health monitoring.

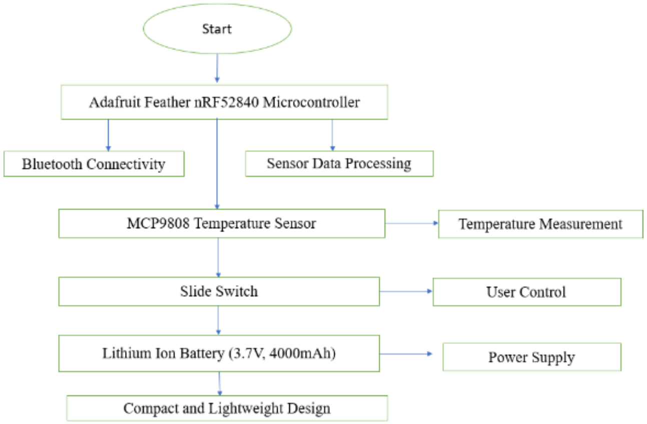
Start
Adafruit Feather nRF52840 Microcontroller
Bluetooth Connectivity
Sensor Data Processing
MCP9808 Temperature Sensor
Temperature Measurement
Slide Switch
User Control
Lithium Ion Battery (3.7V, 4000mAh)
Power Supply
Compact and Lightweight Design

Fig 2 System Architecture of the Temperature Sensor

III. Security Considerations Of Temperature Sensor

Security is paramount in the design and deployment of IoT- based healthcare systems, where sensitive patient data is continuously collected, transmitted, and stored. The IoT-based body temperature measuring sensor system incorporates multiple layers of security measures to protect data confidentiality, integrity, and availability (CIA). This section outlines the key security considerations, encompassing device- level security, communication security, data storage and processing security, and compliance with regulatory standards.

A. Device-Level Security

i. Secure Boot: The microcontroller unit (MCU) in the system supports a secure boot process, which ensures that only authenticated and trusted firmware is executed. Secure boot is implemented using a hardware root of trust, typically embedded in the MCU, which verifies the digital signature of the firmware before allowing it to run. This mechanism prevents the execution of unauthorized or malicious code, protecting the device from firmware tampering or attacks that seek to alter the system's operation.

ii. Firmware Security and Over-the-Air Updates: Firmware security is enhanced through digitally signed OTA updates. The system employs asymmetric cryptography (e.g., RSA or ECC) to sign the firmware updates, with the corresponding public key stored securely in the device's hardware security module (HSM). Before an update is applied, the device verifies the signature to ensure the integrity and authenticity of the firmware. This process prevents unauthorized firmware from being installed, which could otherwise compromise the system.

iii. Hardware Security Modules (HSMs): The MCU is equipped with an HSM, such as ARM TrustZone or a Secure Enclave, which isolates sensitive cryptographic operations from the rest of the system. The HSM securely stores cryptographic keys and performs encryption, decryption, and authentication operations in a protected environment, making it resistant to physical attacks, such as side-channel attacks or attempts to extract keys through invasive methods.

iv. Physical Security: Physical security measures are also considered, particularly for scenarios where the device might be deployed in public or unmonitored environments. The device casing is tamper-resistant, with sensors that detect and respond to physical tampering attempts. These responses might include erasing sensitive data or locking down the device to prevent unauthorized access.

B. Communication Security

i. Data Encryption: All data transmitted between the IoT device and the cloud or mobile application is encrypted using the Advanced Encryption Standard (AES) with a 256-bit key length (AES-256). This level of encryption is considered highly secure and is widely adopted in security-sensitive applications. The encryption is performed in the device's HSM, ensuring that the encryption keys are never exposed to the application layer, thus protecting against key leakage.

ii. Transport Layer Security (TLS): The system employs Transport Layer Security (TLS) version 1.3 for secure communication over the network. TLS provides end-to-end encryption and integrity verification, protecting the data from interception or tampering during transmission. TLS 1.3 is the latest version of the protocol, offering improved security features such as forward secrecy, which ensures that even if a session key is compromised, past communications remain secure. The use of Perfect Forward Secrecy (PFS) in TLS 1.3 ensures that session keys are not derivable even if the server's private key is compromised.

iii. Mutual Authentication: Mutual authentication is implemented using X.509 digital certificates. Both the IoT device and the cloud server are required to present and verify certificates before establishing a secure connection. This mutual authentication process ensures that only trusted devices can communicate with the cloud platform, preventing man-in- the-middle (MITM) attacks and ensuring that data is transmitted only to legitimate endpoints.

iv. Secure Communication Protocols: The system uses MQTT (Message Queuing Telemetry Transport) over TLS for lightweight and secure messaging. MQTT's publish-subscribe model, combined with TLS, ensures that data is transmitted securely with minimal overhead, making it suitable for IoT devices with limited computational resources. HTTP/HTTPS protocols are used for RESTful API communications, with HTTPS providing secure data exchange between the cloud platform and external systems, such as Electronic Health Record (EHR) systems.

v. Secure Key Management: Key management is a critical aspect of communication security. The system employs a Public Key Infrastructure (PKI) for managing digital certificates and encryption keys. The private keys are securely stored in the HSM, and all key exchanges during the TLS handshake are encrypted using elliptic curve cryptography (ECC) or RSA, depending on the configuration. The PKI ensures that encryption keys are regularly rotated, reducing the risk of long- term key compromise.

C. Data Storage and Processing Security

i. Encrypted Data Storage: Data at rest, whether stored on the device or in the cloud, is encrypted using AES-256. On-device storage, such as flash memory, is partitioned, with sensitive data stored in encrypted volumes protected by the HSM. In the cloud, data is encrypted using the cloud provider's encryption services, such as AWS KMS (Key Management Service) or Azure Key Vault, ensuring that patient data is safeguarded even in the event of a data breach.

ii. Secure Cloud Infrastructure: The cloud infrastructure hosting the IoT data is designed to meet high-security standards, including compliance with healthcare regulations like HIPAA and GDPR. The cloud environment is protected by network firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS), which monitor and block unauthorized access attempts. Data is processed in secure, isolated environments, using serverless computing services like AWS Lambda or Azure Functions, which further reduce the attack surface.

iii. Access Control and Authentication: Access to the cloud- based data and services is controlled using role-based access control (RBAC) and multi-factor authentication (MFA). RBAC ensures that users only have the permissions necessary to perform their roles, minimizing the risk of insider threats. MFA adds an additional layer of security, requiring users to provide two or more verification factors to gain access, reducing the likelihood of unauthorized access due to credential compromise.

iv. Data Integrity Verification: Data integrity is maintained through the use of cryptographic hash functions, such as SHA- 256, which generate a unique hash value for each data block. This hash is stored alongside the data, and any alteration to the data would result in a mismatch between the stored hash and the computed hash, indicating potential tampering. Integrity checks are performed at both the device and cloud levels, ensuring that the data remains unaltered throughout its lifecycle.

D. Threat Mitigation and Resilience

i. Intrusion Detection and Prevention Systems (IDPS): The system is protected by Intrusion Detection and Prevention Systems (IDPS) at both the network and device levels. These systems monitor for signs of malicious activity, such as unusual access patterns, attempted exploits, or traffic anomalies. Upon detecting a potential threat, the IDPS can take preemptive actions, such as blocking traffic from suspicious IP addresses, isolating compromised devices, or alerting administrators.

ii. Protection Against Denial-of-Service (DoS) Attacks: Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks pose a significant threat to the availability of IoT systems. To mitigate these attacks, the system employs rate limiting, traffic filtering, and load balancing techniques at the network edge. Cloud providers also offer DDoS protection services, such as AWS Shield or Azure DDoS Protection, which detect and mitigate large-scale DDoS attacks by absorbing and filtering malicious traffic before it reaches the core infrastructure.

iii. Secure Update Mechanism: Regular firmware updates are essential for maintaining the security of IoT devices, as they often address vulnerabilities discovered post-deployment. The OTA update mechanism is secured with cryptographic signatures, ensuring that only authentic updates from the trusted source are applied. This mechanism also supports rollback in case of update failures, ensuring that the device can revert to a previous, known-good state if necessary.

iv. Anomaly Detection and Response: The system incorporates anomaly detection algorithms, which analyse patterns in device behaviour and network traffic to identify potential security incidents. For instance, sudden spikes in data transmission or unexpected changes in sensor readings could indicate a compromised device or network. Upon detecting an anomaly, the system can trigger automated responses, such as alerting administrators, initiating device quarantine, or escalating the issue for further investigation.

E. Compliance and Regulatory Considerations

i. HIPAA Compliance: In the United States, the Health Insurance Portability and Accountability Act (HIPAA) establishes stringent requirements for the protection of healthcare data. The system is designed to comply with HIPAA regulations, ensuring that all Protected Health Information (PHI) is encrypted both in transit and at rest, access to data is strictly controlled, and audit logs are maintained to track access and modifications to patient records.

ii. GDPR Compliance: For deployments in the European Union, the system adheres to the General Data Protection Regulation (GDPR), which mandates the protection of personal data and privacy. GDPR compliance includes ensuring data subject rights, such as the right to access, correct, and delete personal data. Data processing activities are documented, and Data Protection Impact Assessments (DPIA) are conducted to evaluate and mitigate privacy risks.

iii. ISO/IEC 27001 Certification: The cloud infrastructure and data management practices align with ISO/IEC 27001, an international standard for information security management. This certification demonstrates a commitment to maintaining a robust security posture, with a focus on risk management, continual improvement, and compliance with legal and regulatory requirements.

iv. Data Retention and Deletion Policies: The system implements clear data retention policies, specifying the duration for which patient data is stored, after which it is securely deleted.

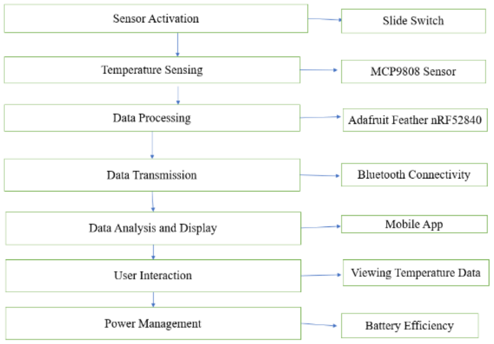

Fig 3: Security Consideration of Temperature Sensor

IV. CONCLUSION

The research presents a comprehensive IoT-based system for continuous body temperature monitoring, addressing the limitations of traditional temperature measurement methods. By integrating advanced microcontroller technology, precision temperature sensors, edge computing, and secure wireless communication protocols, the system achieves real-time, non- invasive monitoring with high accuracy.

Key technical advancements include the use of a low-power ARM Cortex-M microcontroller, which efficiently handles real-time data acquisition and processing, and the implementation of adaptive sampling techniques that optimize power consumption. The system's dual-mode communication approach, combining Wi- Fi for high-throughput data transfer and Bluetooth Low Energy (BLE) for low-power connectivity, ensures flexibility and efficiency in different operational scenarios. Security is a critical focus, with the system employing AES-256 encryption, TLS 1.3 for secure data transmission, and robust hardware security measures like secure boot and hardware security modules (HSMs).

These features ensure the confidentiality and integrity of sensitive patient data throughout its lifecycle. The cloud-based architecture supports scalable and secure data storage and analytics, with integration capabilities for electronic health records (EHR) through a RESTful API. This enables seamless integration into existing healthcare systems, paving the way for enhanced remote patient monitoring and telemedicine applications.

In conclusion, the proposed IoT-based body temperature monitoring system offers a reliable, secure, and scalable solution, significantly improving the efficiency and accuracy of patient monitoring in various healthcare settings. The system's design principles and security considerations make it a valuable tool for modern healthcare, with the potential for broad application in critical care, remote monitoring, and telemedicine environments.

REFERENCES

[1] D. M. Han and J. H. Lim, "Design and Implementation of Smart Home Energy Management Systems Based on ZigBee," IEEE Transactions on Consumer Electronics, vol. 56, no. 3, pp. 1417-1425, Aug. 2010.

[2] L. Da Xu, W. He and S. Li, "Internet of Things in Industries: A Survey," IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233-2243, Nov. 2014.

[3] R. Want, B. N. Schilit and S. Jenson, "Enabling the Internet of Things," Computer, vol. 48, no. 1, pp. 28-35, Jan. 2015.

[4] M. R. Palattella et al., "Internet of Things in the 5G Era: Enablers, Architecture, and Business Models," IEEE Journal on Selected Areas in Communications, vol. 34, no. 3, pp. 510-527, March 2016.

[5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, Fourthquarter 2017.

[6] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645-1660, Sept. 2013.

[7] D. E. Kouicem, A. Bouabdallah and H. Lakhlef, "Internet of Things Security: A Top-Down Survey," Computer Networks, vol. 141, pp. 199- 221, Aug. 2018.

[8] S. R. Moosavi et al., "End-to-End Security Scheme for Mobility Enabled Healthcare Internet of Things," Future Generation Computer Systems, vol. 64, pp. 108-124, Nov. 2018.

[9] H. Alemdar and C. Ersoy, "Wireless Sensor Networks for Healthcare: A Survey," Computer Networks, vol. 54, no. 15, pp. 2688-2710, Oct. 2010.

[10] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1125-1142, Oct. 2018.

[11] Y. Yang, L. Wu, G. Yin, L. Li and H. Zhao, "A Survey on Security and Privacy Issues in Internet-of-Things," IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1250-1258, Oct. 2018.