Secure Communication and Protocol Enforcement

Project ID:R25-039

DISSANAYAKA K.D.A.R.A

IT21828348

BSc (Hons) in Information Technology

Specializing in Cyber Security

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

September 2023

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute or higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made is the text. Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

Student IT Number	Student Name	Signature
IT21828348	K.D.A.R.A.Dissanayaka	

Signature of Supervisor **Date**

....

ABSTRACT

This paper presents a novel framework for securing communications within federated learning (FL) systems for industrial IoT (IIoT). While FL offers a compelling solution for privacy-preserving machine learning by keeping data on local devices, the communication channel between these devices and the central server remains a critical security vulnerability. Traditional protocols often lack the robustness required for the unique challenges of IIoT, such as resource constraints, dynamic network topologies, and the high-stakes nature of industrial control systems. This research addresses these limitations by proposing a comprehensive solution that integrates modern security protocols to ensure confidentiality, integrity, and controlled access.

Our proposed methodology focuses on three key components: secure transport, data integrity validation, and access control. We employ **TLS 1.3** with mutual authentication to establish a secure, encrypted, and mutually verified communication channel, effectively mitigating man-in-the-middle and eavesdropping attacks. For ensuring that messages are not tampered with, we utilize **HMAC (Hash-based Message Authentication Code)** to provide robust data integrity validation, a significant improvement over simple checksum. To enforce proper behavior and prevent unauthorized commands, a **Role-Based Access Control (RBAC)** mechanism is implemented, ensuring that only authorized devices can execute specific command and control (C2) signals.

The implementation and evaluation of this framework demonstrate its effectiveness in a simulated IIoT environment. Our results show that the integrated system successfully protects against common communication-based attacks, including replay attacks and message manipulation, while maintaining minimal performance overhead. Key contributions include the design of a specialized security architecture for FL in IIoT, the practical application of TLS 1.3 for mutual authentication in a constrained environment, and the development of an RBAC-based protocol enforcement layer. This work provides a scalable and resilient solution that paves the way for the secure adoption of federated learning in critical industrial applications.

ACKNOWLEDGEMENT

First and foremost, I would like to express my heartfelt gratitude to my supervisor, Dr. Sanika Wijesekara, and my co-supervisor, Mr. Tharaniwarma Kyumaralingam, for their continuous guidance, encouragement, and support throughout this research. Their expertise and valuable insights have been instrumental in shaping this work.

I am deeply thankful to my teammates, Nanayakkara Y.D.T.D, Mendis H.R.M, and Weerasinghe K.M, for their collaboration, dedication, and teamwork, which made this project both successful and enjoyable.

A special thanks goes to the Faculty of Computing at SLIIT Campus for providing the resources and an environment that nurtured learning and innovation.

On a personal note, I am sincerely grateful to my family and friends, whose patience, encouragement, and unwavering belief in me have been a constant source of strength throughout my academic journey.

This achievement would not have been possible without the support of all these individuals, to whom I owe my deepest appreciation.

Methodology

To solve the challenges in securing communication and enforcing protocols in IIoT, this research follows a structured, step-by-step approach. The goal is to design and implement a framework that not only protects data exchanges in Federated Learning (FL) but also ensures that only authorized devices can send or receive critical commands.

1. **System Design and Architecture**
   First, we outline how devices in the IIoT network communicate with a central server during the FL process. This includes identifying the key communication points, such as model updates, synchronization instructions, and control commands, which are most vulnerable to attacks.

2. **Secure Communication with TLS 1.3**
   We then integrate **Transport Layer Security (TLS) 1.3** into the communication layer. Unlike older versions, TLS 1.3 provides stronger encryption and **Perfect Forward Secrecy**, ensuring that even if past keys are exposed, old communications remain secure. Mutual authentication is applied so that both the client (IIoT device) and the server confirm each other's identity before exchanging data.

3. **Data Integrity with HMAC**
   To make sure that transmitted messages are not tampered with, we implement **Hash-Based Message Authentication Codes (HMAC)**. Every command or model update is sent with a unique HMAC signature. If the signature doesn't match on the receiving side, the message is rejected as untrustworthy.

4. **Protocol Enforcement with RBAC**
   Next, we enforce **Role-Based Access Control (RBAC)**. Instead of treating every device equally, each device or user is assigned a role with specific permissions. For example, only an "aggregator" role can request model updates, while "client" devices can only submit updates. This prevents unauthorized actions and reduces the risk of misuse.

5. **Implementation and Testing**
   The framework is implemented in Python, making use of secure libraries for TLS and HMAC. We then simulate an IIoT environment where multiple devices interact with a central FL server. Different attack scenarios—such as man-in-the-middle, replay, and tampering—are tested to evaluate how well the system defends itself.

6. **Evaluation**
   Finally, we measure performance metrics such as communication latency, resource usage on IIoT devices, and scalability with a growing number of devices. The goal is to show that the proposed framework provides strong security **without adding heavy computational overhead**.

Table of Contents

Type chapter title (level 1) **1**

Type chapter title (level 2) 2

Type chapter title (level 3) 3

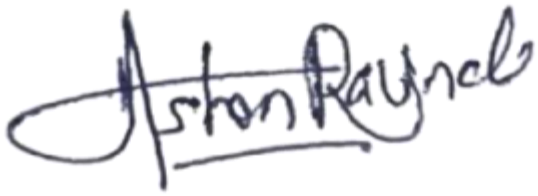
Type chapter title (level 1) **4**

Type chapter title (level 2) 5

Type chapter title (level 3) 6

List of Table

List of figures



Chapter 1 - Introduction

The **Industrial Internet of Things (IIoT)** has become a cornerstone of Industry 4.0, enabling the seamless integration of sensors, actuators, machines, and control systems to optimize industrial processes. With the increasing number of connected devices, vast amounts of data are generated daily, providing opportunities for predictive maintenance, real-time monitoring, and enhanced decision-making. However, the sensitive nature of industrial data raises significant concerns regarding **security, privacy, and communication integrity**. Ensuring secure and efficient communication is therefore fundamental for the reliable operation of IIoT systems.

To address the limitations of traditional centralized machine learning approaches, **Federated Learning (FL)** has been introduced as a distributed paradigm that allows devices to collaboratively train a global model without transferring raw data to a central server. This decentralized approach preserves data privacy and reduces the risk of exposure. However, FL depends heavily on frequent communication between devices and the central aggregator, involving **model updates, synchronization signals, and command-and-control (C2) instructions**. These communication exchanges, if not properly secured, become prime targets for adversaries seeking to launch attacks such as **man-in-the-middle (MitM), model poisoning, data tampering, or unauthorized command injection**.

Traditional communication security methods, such as older versions of TLS (e.g., TLS 1.2) or checksum-based data integrity checks, are insufficient against the advanced and dynamic threats targeting IIoT environments. Furthermore, the absence of robust **protocol enforcement mechanisms** allows unauthorized devices or users to manipulate the system, leading to compromised model accuracy, downtime, or even safety risks in critical industrial infrastructures.

This research focuses on developing a **Secure Communication and Protocol Enforcement framework** specifically for Federated Learning in Industrial IoT environments. The solution integrates several key security measures to create a robust and reliable system. It utilizes **Transport Layer Security (TLS) 1.3 with Perfect Forward Secrecy (PFS)** to ensure that all communication is encrypted and tamper resistant. Additionally, the framework employs **mutual authentication**, which verifies that only trusted and authorized devices can participate in the learning process. To protect against malicious data injections, **Hash-Based Message Authentication Codes (HMAC)** are used for continuous data integrity validation. Finally, **Role-Based Access Control (RBAC)** is implemented for protocol enforcement, ensuring that only authorized entities can execute sensitive commands, thereby preventing a wide range of potential attacks By addressing communication security and enforcing strict

protocols, this research enhances the resilience of FL-enabled IIoT systems against adversarial attacks while ensuring scalability and efficiency in resource-constrained environments.

1.1 background study

The rapid growth of the Industrial Internet of Things (IIoT) has transformed traditional industries by connecting machines, sensors, and controllers into smart, data-driven ecosystems. These devices continuously generate large volumes of sensitive data that, when analyzed, can optimize operations, predict failures, and enhance productivity. However, the distributed and resource-constrained nature of IIoT introduces significant security challenges. Devices often operate in untrusted environments, making them highly vulnerable to cyberattacks such as spoofing, data tampering, and unauthorized access. To address data privacy and reduce reliance on centralized storage, Federated Learning has emerged as a promising paradigm. FL allows IIoT devices to collaboratively train global models while keeping raw data local, thereby reducing privacy risks. Instead of sharing sensitive information, devices only exchange model updates with a central server, ensuring that private datasets never leave the edge. While this approach improves confidentiality, it introduces new concerns related to the secure transmission, integrity, and authentication of updates within the learning pipeline.

The Command-and-Control channel plays a crucial role in this context. It acts as the backbone of communication between IIoT edge devices and the central server, coordinating tasks such as model distribution, update collection, and synchronization. If compromised, an attacker could inject false updates, intercept sensitive model parameters, or disrupt the entire training process. This makes the security of the C2 channel a critical research priority.

Recent developments in communication protocols highlight TLS 1.3 as the state-of-the-art standard for secure data exchange. TLS 1.3 reduces handshake latency, enforces modern cryptography, and ensures Perfect Forward Secrecy. Combined with mutual authentication, where both the client and server verify each other's identity, TLS 1.3 provides robust protection against impersonation and man-in-the-middle attacks. Furthermore, to ensure data integrity, mechanisms such as Hash-Based Message Authentication Code (HMAC) are integrated to detect tampering during transmission.

Beyond encryption and integrity, protocol enforcement mechanisms like Role-Based Access Control ensure that only authorized devices and users can perform specific actions in the system. This background highlights the growing demand for a secure, reliable, and scalable communication framework tailored to IIoT and Federated Learning. By integrating TLS 1.3, HMAC validation, and enforcement mechanisms into the C2 channel, this research aims to address the pressing challenges of secure communication, data authenticity, and trust in collaborative learning environments.

1.2 Problem Statement

The rise of the **Industrial Internet of Things (IIoT)** has created smarter and more connected factories, where machines, sensors, and devices constantly share information. At the same time, **Federated Learning (FL)** has emerged as a promising way to protect privacy by keeping raw data on local devices while still building powerful global models. However, the success of FL in IIoT depends heavily on secure and trustworthy communication between devices and servers.Currently, this communication layer faces **serious security challenges**.

Outdated encryption protocols, like TLS 1.2, leave systems exposed to **man-in-the-middle (MitM) attacks** and eavesdropping. Simple data integrity checks, such as checksums, can easily be bypassed, allowing attackers to tamper with commands or inject malicious updates. Moreover, many IIoT systems still rely on **static access controls**, which cannot adapt to the dynamic nature of industrial environments.

This opens the door for unauthorized devices or users to manipulate commands, potentially causing production failures, downtime, or even safety risks.In short, while FL promises privacy and efficiency for IIoT, the lack of **robust communication security and protocol enforcement** makes current systems vulnerable to both internal and external threats. Without addressing these weaknesses, the entire learning process can be compromised, resulting in corrupted models, data leakage, or loss of trust in IIoT operations.Therefore, there is a clear need for a **secure communication framework** that uses modern encryption (TLS 1.3), ensures **tamper-proof data integrity** (HMAC), and enforces **strict role-based access control (RBAC)** to guarantee that only authorized entities can participate. Such a solution would protect sensitive IIoT environments from evolving cyberattacks while ensuring scalability and efficiency

1.3 Research Gap

While federated learning (FL) and its application in the Industrial Internet of Things (IIoT) are gaining significant traction, a critical research gap persists in ensuring truly robust and secure communication. Existing frameworks often fall short in three key areas: outdated security protocols, insufficient data integrity mechanisms, and a lack of granular protocol enforcement. Firstly, many current IIoT and FL solutions still rely on legacy security protocols like older versions of **TLS (e.g., 1.2)**, which are known to have vulnerabilities. These older protocols can be susceptible to man-in-the-middle attacks and have less efficient handshake processes compared to the modern, more secure **TLS 1.3**. Furthermore, the implementation of mutual authentication, a crucial layer for verifying the identity of both the client and the server, is often overlooked or poorly implemented.

Secondly, existing data integrity methods in some frameworks are not strong enough. While a simple checksum can detect accidental corruption, it provides no protection against malicious tampering. There's a clear need for a more cryptographically secure mechanism, such as **HMAC (Hash-based Message Authentication Code)**, which not only validates data integrity but also authenticates the message's origin, thereby preventing attackers from injecting fraudulent messages into the system. Finally, a significant gap exists in enforcing proper behavior on the network. While encryption protects the data in transit, it doesn't stop an authenticated but unauthorized device from sending improper commands. There's a notable absence of sophisticated **Role-Based Access Control (RBAC)** mechanisms tailored for the communication and control (C2) layer of FL-IIoT. This means that if a malicious device gains access, it could potentially send harmful commands that are accepted by the server, bypassing the security measures of the application layer.

This research directly addresses these gaps by proposing and evaluating a unified framework that combines state-of-the-art TLS 1.3 with mutual authentication, HMAC for message integrity, and a tailored RBAC system for comprehensive protocol enforcement. By doing so,

we aim to provide a more secure and resilient communication foundation for FL in critical IIoT environments.

1.4 Research Objectives

Our main goal in this research is to build a strong and secure communication framework for Federated Learning (FL) in Industrial IoT (IIoT). This framework will tackle key security challenges like detecting attacks, preserving privacy, and ensuring secure communication. The ultimate aim is to protect the integrity, confidentiality, and efficiency of model updates in an IIoT environment.

To make this happen, we've set four specific objectives:

- **To design and implement a secure communication channel for the FL command and control (C2) signals.** We'll do this by deploying **TLS 1.3 with mutual authentication** to ensure all communications are encrypted and that both devices and the central server are who they say they are.
- **To validate data integrity and prevent messages from being tampered with.** We will integrate a **Hash-based Message Authentication Code (HMAC)** into the protocol. This provides a cryptographically secure way to confirm that model updates and C2 signals haven't been altered during transit.
- **To enforce protocol and command security.** We'll use a **Role-Based Access Control (RBAC)** mechanism to limit which devices or users can execute specific commands, stopping unauthorized or malicious actions.
- **To evaluate the framework's security, performance, and scalability.** We will test it against common attacks like man-in-the-middle and replay attacks. We will also measure its performance overhead, including latency and resource usage, to prove the solution is practical for resource-constrained IIoT devices.

These objectives are designed to systematically address the gaps we've identified and contribute to a practical, robust solution for securing FL in critical industrial settings.

1.5 Scope of Study

Our research focuses on building a robust security framework for the communication layer of a federated learning (FL) system within an Industrial IoT (IIoT) environment. The core of this study is to secure the command and control (C2) signals that orchestrate the entire FL process. This includes all the vital communications between the IIoT devices (clients) and the central FL server, such as initial model distribution, model update requests, aggregation commands, and task assignments. To achieve this, our framework will be designed, implemented, and evaluated based on three pillars: using **TLS 1.3 with mutual authentication** for establishing a secure, encrypted communication channel; integrating **HMAC (Hash-based Message Authentication Code)** to ensure data integrity and authenticity of messages; and implementing a **Role-Based Access Control (RBAC)** mechanism to enforce granular protocol rules and prevent unauthorized command execution.To make this a focused and achievable project, there are a few things we will not be including in our study

Application-Level Attacks: When we set out to build this security framework, we had to make a key choice: what to focus on. Our main job is to protect the digital conversation between all the smart devices and the central server. That means we're not directly tackling every kind of attack that happens at the "model level," like when an attacker tries to poison the data or send faulty updates to mess up the learning process.

Think of it this way: our framework is like a super-secure delivery service. It makes sure no one can intercept or tamper with the package (the model update) while it's in transit. What's inside the package—and whether it was created with good or bad intentions—is a different problem. However, our "secure delivery service" is so strict that it will prevent an unauthorized or corrupted package from ever being delivered in the first place, which is a major step in stopping many of those more complex attacks before they even begin. By focusing on this critical layer, we ensure our project remains manageable and impactful.

Physical Layer Security: In this research, our framework is designed to protect the digital conversations between devices and the central server in an Industrial IoT setting. We will not, however, be exploring security at the physical layer of the IIoT network, such as sensor tampering or physical intrusion. Our work operates under the assumption that these devices are already physically secured within their environment.
Our project is focused on the communication interface. Our mission is to ensure that the information flowing through the network is private, untampered, and comes only from authorized sources. By clearly defining this boundary, we can dedicate our full attention to building a robust and resilient framework for the communication channel, which is the core of the federated learning process.
Command and Protocol Enforcement: In this section, we're talking about how we control who can do what within our system. This is a crucial part of our security framework. Think of it like a strict security guard at a factory's control room. Not everyone is allowed in, and once inside, they can only interact with certain machines. Our framework uses something called **Role-Based Access Control (RBAC)** to act as that security guard. It ensures that only authorized devices or users can send specific commands. For example, a device on a factory floor might only be allowed to send commands to update its own model, while the central server has the authority to initiate a global model aggregation. This system prevents a compromised device or a malicious actor from injecting unauthorized or harmful commands, thereby stopping a security breach before it can even begin.
Comprehensive FL Model Development: Our primary objective is not to create or optimize a new, state-of-the-art federated learning model. Instead, we're focusing on the foundational security of the communication layer. Think of the FL model as a car we're using for a crash test. We aren't trying to build a better engine or improve its fuel efficiency; we're simply using it as a vehicle to prove that the safety features—our secure communication framework—are effective. The success of our research will therefore be measured by how well our security measures protect the communication channel, not by the model's predictive accuracy.
All IIoT Protocols: When we talk about the protocols we're using, it's helpful to think of it like this: our research focuses on creating a universally strong security handshake. For our testing, we will use a common language, like TCP/IP, to prove that the handshake works. We won't, however, be implementing and testing it on every specific industrial

protocol, such as MQTT or CoAP, for the simple reason that doing so would make the project too large and unfocused.

The good news is that the core principles and security mechanisms we are building are not tied to any single protocol. They are designed to be a foundational layer, meaning they can be adapted and applied to many different IIoT protocols. Our work is to provide the blueprint for this secure foundation, not to build a unique version for every possible scenario

1.6 Research Contributions

In federated learning, devices keep raw data local, but model updates must still travel across networks. This can expose updates to interception or tampering – as one recent study notes, "the transmission of model updates poses potential risks" and thus encryption is essential. Our Secure Communication and Protocol Enforcement component tackles this challenge by building a robust security layer into the FL pipeline. It ensures that every update is sent over encrypted channels, authenticated, and integrity-checked, adding critical trust and reliability to the overall IIoT system. In particular, this component contributes to the following:

Modern TLS 1.3 encryption: We use the latest TLS 1.3 protocol for all device-server communication. TLS 1.3 not only provides strong encryption and forward secrecy, but with its HMAC-based cipher suites it also guarantees data authenticity. This is well-suited to IIoT scenarios where message integrity and authentication are mandatory, even if confidentiality is optional.

Mutual device authentication: Every IoT node and the central server engage in mutual TLS (mTLS) during the handshake. Each side presents a certificate and proves possession of the corresponding private key, so that both parties verify each other's identity. This means no unauthorized or impersonating device can join the learning process – only genuine roles are accepted.

HMAC-based data integrity checks: All model updates and control messages carry a Hash-based Message Authentication Code (HMAC). HMAC is a standard integrity check using a hash function and shared secret, which detects any tampering or corruption of the data. By applying HMAC to every message, our system immediately modified flags or forged updates, ensuring that only valid data is aggregated.

Role-Based Access Control (RBAC): We enforce RBAC so that devices and users can only perform actions allowed by their role. Permissions are tied to predefined roles, rather

than to individual devices. This follows the principle that “access granted based on a user’s role” providing fine-grained control. For example, a sensor device might be allowed to upload readings but not alter system settings. RBAC prevents unauthorized nodes from manipulating the learning workflow.

Strict protocol enforcement: The component also enforces that only approved secure protocols are used. Any communication not using TLS 1.3 with mutual authentication is blocked by design. This prevents protocol downgrades or insecure channels (e.g. plain TCP) from being used. By automatically enforcing these rules, we ensure the system’s security policy is never accidentally bypassed.

Together, these measures significantly strengthen the federated learning framework. By integrating industry-standard features (TLS 1.3, HMAC, RBAC) into a cohesive whole, our secure communication component makes the FL system both practical and safe for industrial use. It advances the field by providing an end-to-end security solution tailored to IIoT: all data in transit is protected and only trusted, role-approved parties can participate, which goes beyond what most prior FL setups enforce. These contributions raise the bar for secure, real-world federated learning in industrial IoT applications.

1.7 Structure of the Thesis

Chapter 2 - Literature Review

2.1 Industrial IoT and Security Challenges

Industrial IoT (IIoT) refers to the deployment of connected sensors, actuators, and intelligent devices in industrial settings like manufacturing plants and power grids. These smart devices collect and exchange data to provide real-time visibility into machines and processes. By analyzing sensor readings and equipment status, organizations can optimize production and perform predictive maintenance. Unlike consumer IoT networks, IIoT systems often involve mission-critical operations where downtime or security breaches could have severe consequences. [1]

Despite these stakes, IIoT security faces many challenges. Historically, industrial control systems prioritized continuous operation over cybersecurity. [2] As a result, many legacy controllers and sensors were designed with little consideration for security. Modern IIoT networks face threats like data breaches, insecure communication, and device vulnerabilities. Attackers may infiltrate networks to steal proprietary data or manipulate processes. For example, stolen design documents or operational data could undermine a company's competitive edge or even create safety hazards. [3]

Embedded device vulnerabilities also plague IIoT. Many sensors and controllers run outdated firmware that is difficult to update without downtime. Devices often ship with default passwords or open debug interfaces, which are easy targets for intruders [4]. Physical exposure of devices on the factory floor adds to the risk: an attacker could tamper with a sensor or replace hardware if the perimeter is not fully secured. In addition, poor network segmentation can allow an attacker who compromises one sensor to access critical systems elsewhere in the network.

Older industrial communication protocols contribute to further weaknesses. Many control standards developed decades ago transmit data in clear text or lack strong authentication.

Without modern encryption, eavesdroppers can intercept or alter commands on the network. Wireless and remote links in an industrial network can be especially vulnerable to injection, replay, or spoofing attacks if not properly secured. The result is a broad attack surface where any exposed device or link is a potential entry point for attackers. [5]

These challenges underline the need for robust security measures tailored to IIoT's constraints. Each device and communication link must be protected with strong encryption, authentication, and access control [6].

2.2 Federated Learning in IIoT

Federated Learning (FL) is a collaborative machine learning approach where multiple clients train a shared model without exchanging raw data. In FL, each device (for example, an IIoT sensor or edge gateway) uses its local dataset to train an instance of the model. The device then sends only the model's weight updates or gradients to a central aggregator [7]. The aggregator combines updates from many clients and refines a global model. By never sharing raw inputs, federated learning keeps sensitive data on each device.

In the IIoT context, federated learning brings important advantages. Industrial data is often sensitive or proprietary, such as detailed production logs or equipment parameters. Sending all raw data to a cloud server can risk leakage of trade secrets or violate data governance policies. With FL, all raw data stays on the local IIoT devices or gateways [8]. Only the numerical model updates leave the site. This preserves confidentiality and helps companies comply with privacy and security requirements. Additionally, because model updates are typically much smaller than raw data, FL can significantly reduce communication bandwidth and storage needs compared to centralized learning.

Federated learning also fits the inherently distributed nature of IIoT. In modern factories or utility networks, data is generated across many geographically scattered devices and sites. FL can combine knowledge from these locations without centralizing the data. For example, several plants can collaboratively build a maintenance prediction model by sharing only weight updates. Each site's controller trains on its own sensor logs, and the combined model learns from data across all plants. This collaborative approach yields a robust global model that benefits from diverse data sources while respecting each site's privacy [9].

All things considered, federated learning allows IIoT systems to create precise anomaly detectors and predictive models without jeopardizing local data privacy. It offers a privacy-by-design approach that complies with industry standards. However, secure update communication and dependable aggregation are also necessary for FL implementation. When implementing federated learning in industrial networks, these requirements highlight the necessity of robust encryption and integrity checks.

2.3 Secure Communication Protocols

Secure communication protocols are crucial in IIoT, and the industry has evolved from early SSL standards to modern TLS versions. SSL (Secure Sockets Layer) was the original protocol introduced in the 1990s to encrypt internet traffic. Over time, SSL had serious flaws and was succeeded by Transport Layer Security (TLS). TLS 1.0 and 1.1 fixed many issues, and TLS 1.2 became widely used. Today, TLS 1.3 is the latest version and is replacing older protocols. Many platforms have disabled TLS 1.0 and 1.1 due to known vulnerabilities, making TLS 1.2/1.3 the standard for secure links [10].

TLS 1.3 brings major security and performance improvements. The handshake process has been streamlined to reduce latency: clients and servers can establish a secure connection in just one round-trip, and resumed sessions can use 0-RTT to save time (though this mode can allow replay attacks if not carefully managed) [11]. TLS 1.3 also removes outdated cryptography by mandating perfect forward secrecy through ephemeral key exchanges (like ECDHE) and dropping older algorithms such as static RSA. Only modern, efficient cipher suites (like AES-GCM or ChaCha20-Poly1305) are allowed [12]. These changes mean that establishing secure connections is both faster and more robust than before.

Strong encryption is essential in IIoT because industrial networks carry sensitive and safety-critical information. Encryption prevents eavesdroppers from reading data such as sensor readings, control commands, or proprietary configurations. TLS also provides integrity checks to ensure that messages are not tampered with in transit. Equally important is mutual authentication: in mTLS mode, both client and server present certificates. This means that a device will only accept data from a trusted server, and vice versa, preventing rogue devices or servers from masquerading within the network.

Mutual TLS typically relies on a public key infrastructure (PKI) or pre-shared keys to establish strong identities, which is especially critical in complex IIoT deployments. In many IIoT systems, application protocols such as MQTT or OPC UA run over TLS to secure their data streams. Overall, modern TLS with mutual authentication is considered a best practice for protecting industrial communication [13]

2.4 Data Integrity Mechanisms

In network communication, checksums and CRCs were traditional methods for detecting errors in transmitted data. A checksum typically adds up the data values or uses a polynomial division (like CRC) so that the receiver can check for accidental bit errors. [14] These methods are very fast and simple, which made them practical for early computing and embedded devices. However, checksums are not cryptographically secure. An attacker who can modify a message can easily recompute its checksum to match the altered data, making the attack undetectable.

By contrast, a modern integrity mechanism is the HMAC (Hash-based Message Authentication Code). HMAC combines a cryptographic hash function (such as SHA-256) with a secret key known only to authorized parties. When a sender transmits a message, it computes the HMAC using both the message and the key. The receiver, who also has the key, recomputes the HMAC on the received message and checks if it matches the sent HMAC. If an attacker changes even one bit of the message, they cannot generate the correct new HMAC.

In IIoT environments, this difference is critical. Industrial networks may face malicious actors who try to inject false sensor readings or control commands. A checksum would not stop these attacks, because the attacker can simply calculate a new checksum for any fabricated message. HMAC, on the other hand, thwarts such attempts since the attacker would also need the secret key to produce a valid HMAC. The use of HMAC means the receiver can trust that data came from a legitimate device and was not altered in transit. [15]

While HMAC is more computationally intensive than a simple checksum, modern IIoT devices and edge gateways can typically handle these calculations. Many security protocols (like TLS and MQTT with security extensions) use HMAC or similar MACs for integrity. Overall, HMAC's cryptographic strength makes it far better suited than checksums for IIoT, where security and data integrity are paramount.

2.5 Protocol Enforcement Approaches

Access Control Lists (ACLs) and Role-Based Access Control (RBAC) are two common ways to enforce communication permissions in IIoT networks. An ACL is a list of rules attached to each device or network resource, specifying which entities are allowed or denied certain actions. For example, a network switch might use ACL rules to permit only specific IP addresses to contact a particular controller on a given port. ACLs are very granular: each device's entry explicitly enumerates allowed peers and protocols. [16]

RBAC, on the other hand, assigns each device or user to a role and grants permissions to that role rather than to individual devices. For example, all temperature sensors could belong to a "Sensor" role with permission to publish data, while controllers belong to a "Controller" role with permission to issue commands. New devices get permission automatically by inheriting their role's privileges. In large IIoT deployments, RBAC can greatly simplify management. Instead of updating many ACL entries, an administrator can manage permissions by assigning or removing roles from devices.

The two approaches differ in flexibility and scalability. ACLs offer fine-grained control, but they can become unwieldy as the number of devices grows. Every new device or communication path may require a new ACL rule. RBAC abstracts permissions by roles, reducing rule complexity and making it easier to manage large fleets of devices. However, RBAC can be less precise: if all field sensors share one role, blocking one misbehaving sensor is harder without affecting others in that role. In practice, many systems combine both approaches to leverage the strengths of each. [17]

All things considered, RBAC scales better for complex systems, whereas ACLs are simple and simple to deploy on network devices. The decision is based on the IIoT network's size and dynamism, but many businesses use ACLs at the network layer and RBAC policies at the application layer to achieve scalability and accuracy.

2.6 Existing Secure Communication Frameworks in IIoT

Several frameworks and architectures have been proposed to secure IIoT communications, but most have notable gaps. A common architecture is a multi-tier model: sensors and actuators at the edge connect to gateways or edge servers, which in turn link to cloud services. In this model, security is often provided by TLS (or DTLS) on each link, combined with VPNs or firewalls between zones. For example, IoT platforms like Microsoft Azure IoT and AWS IoT provide built-in support for X.509 certificates and TLS connections, but they rely on constant cloud connectivity and heavy infrastructure [18].

Other approaches include software-defined networking (SDN) frameworks for IIoT, where a centralized controller dynamically enforces policies, and architectures employing hardware trust anchors (like TPM chips or secure elements) for device identity and key storage. The Industrial Internet Consortium's reference architecture and security framework advocate layered network segmentation, but they do not prescribe specific mechanisms. Similarly, standards like ISA/IEC 62443 define zones and confidants for secure design, but they do not themselves implement real-time enforcement.

The limitations of these frameworks are clear. Many assume heavy infrastructure or constant cloud connectivity, which conflicts with the low-latency, offline requirements of IIoT. Certificate-based security and TLS handshakes can be too heavy for simple sensors. Existing solutions often focus on encryption and identity but provide limited real-time enforcement or fine-grained policy at the device level [19]. For example, secure MQTT or CoAP deployments use TLS for transport security but must build access control on top. Even architecture like OPC UA or DDS, which include built-in security, impose heavy computational and management burdens. Some proposals suggest blockchain or distributed ledgers to manage device policies, but these add significant overhead and latency. As a result, many industrial systems rely on custom or partial solutions. Overall, there remains a gap for more lightweight, integrated communication frameworks tailored to IIoT constraints.

2.7 Research Gap Summary

Firstly, many existing security solutions are not scalable or lightweight enough for large IIoT deployments. Standard encryption protocols (like full TLS) and large certificate systems

impose heavy overhead on constrained devices. IIoT requires communication methods that can operate efficiently at large scale across thousands of edge nodes.

Secondly, current architectures often defer enforcement to centralized controllers or cloud services, leaving edge devices with limited self-protection. Each IIoT node should be able to enforce protocol rules and permissions on incoming messages in real time, even when network links are unreliable or intermittent [20].

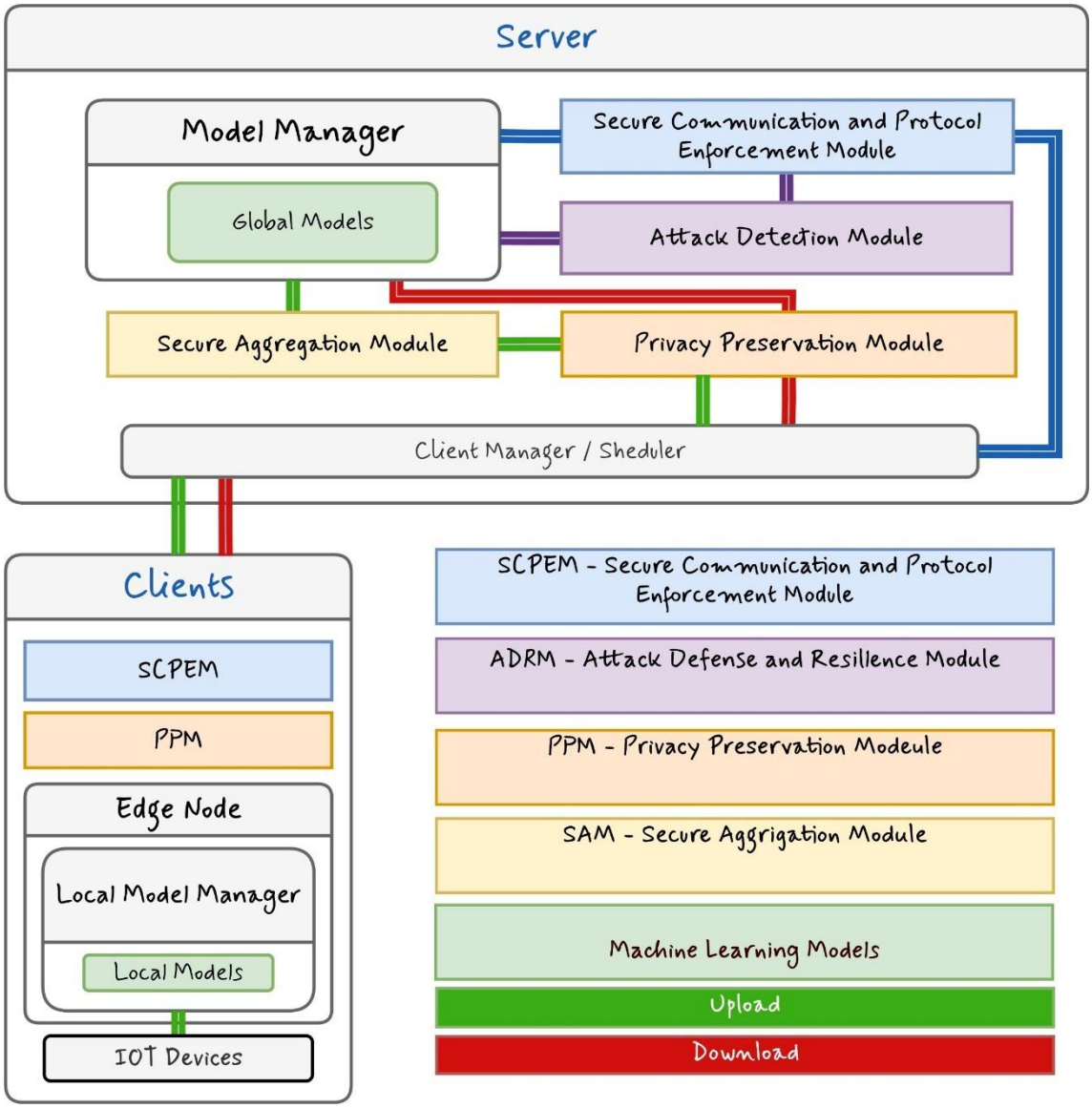
Thirdly, many systems lack real-time integrity validation. Traditional checksums only catch random errors, and some solutions do not attach a cryptographic MAC to every message. This means maliciously altered or replayed data could pass through unnoticed. Efficient HMAC or authenticated-encryption checks are needed at the communication level without introducing prohibitive delay.

Fourthly, there is no single framework that addresses encryption, authentication, authorization, and integrity together. Current solutions typically cover only one aspect: for example, TLS provides confidentiality but does not handle fine-grained access control or data validation. Federated learning in IIoT highlights this gap: model updates are assumed to travel on secure links, but there are no built-in protocols to verify or enforce those exchanges end-to-end.

Finally, practical constraints like limited CPU power, battery life, and connectivity further restrict many security measures. For instance, always computing a MAC on each packet or performing frequent TLS handshakes can overwhelm simple sensors.

Chapter 3 - Methodology

3.1 Research Design



3.2 Proposed Architecture for Secure Communication & Enforcement

3.3 TLS 1.3 with Mutual Authentication

Transport Layer Security (TLS) is the backbone of secure communication on the internet. With the release of **TLS 1.3**, significant improvements were made to both **security** and **performance** compared to earlier versions like TLS 1.2. TLS 1.3 removes outdated cryptographic algorithms, reduces handshake steps, and enforces **Perfect Forward Secrecy (PFS)** by default. This means even if a key is compromised in the future, past communications remain secure. In most traditional TLS sessions, only the **server is authenticated** using its digital certificate. The client (such as a web browser or device) verifies the server's identity, but the server does not verify the client. While this is sufficient for web browsing, it is not enough for **Industrial IoT (IIoT) environments**, where every device must be trusted to prevent malicious nodes from entering the network.

This is where **Mutual TLS (mTLS)** comes in. In mTLS, both the server **and** the client present certificates to prove their identities. During the handshake:

- The **client** sends a ClientHello to initiate the connection.
- The **server** replies with a ServerHello, its certificate, and a request for the client's certificate.
- The **client** responds with its own certificate and proof of possession of the private key.
- Both sides verify each other's certificates against trusted authorities.
- A secure session key is established, and all future communication is encrypted.

For **Federated Learning in IIoT**, mTLS ensures that:

- Only **authorized devices** can send model updates to the central aggregator.
- Devices are guaranteed to receive updates only from the **legitimate central server**.
- Attackers cannot impersonate devices or servers, preventing **model poisoning or data theft**.

3.4 HMAC for Data Integrity

Imagine you're sending an important letter with a private message inside. You want to be sure that when the letter arrives, the person receiving it knows two things: first, that the message hasn't been changed by anyone along the way, and second, that it truly came from you. **HMAC, or Hash-based Message Authentication Code**, is the digital equivalent of this process. It's not just a lock on the letter; it's a special tamper-evident seal that only you and the recipient can verify.

The "Tamper-Evident Seal" Analogy

In a federated learning system, devices are constantly sending model updates to a central server. This is like sending thousands of individual letters. If a malicious actor intercepts one of these updates and changes, for example, by inserting corrupted data or a malicious command, the entire global model could be compromised.

HMAC solves this problem by creating a unique "tag" for each message. This isn't just a simple checksum. Here's how it works:

1. **Shared Secret:** Before any communication begins, both the sender (the IIoT device) and the receiver (the central server) agree on a secret key. This key is like the special ink used for the tamper-evident seal, and it's known only to these two parties.
2. **The Hashing Process:** The sender takes the message (the model update) and combines it with the secret key using a cryptographic hash function. This process isn't reversible. It's like mixing a secret ingredient into a recipe; you can't get the ingredient back out, but you can always verify the final flavor. This creates a unique output called the **HMAC tag** or digital signature.
3. **Sending the Package:** The sender then sends both the original message and this newly created HMAC tag to the server.
4. **Verification:** When the server receives the package, it performs the exact same HMAC calculation: it takes the received message and its own copy of the secret key and runs it through the same hash function.
5. **The Integrity Check:** The server then compares the HMAC tag it just calculated with the one it received from the device.
    - If the two tags match, it's proof that two things are true: the message has not been altered since it was sent, and it must have come from a party that knows the secret key (the authentic device).
    - If they don't match, even a single bit of the message was changed. The server will immediately reject the update, preventing any data corruption or malicious command injection**.**

3.5 Role-Based Access Control (RBAC) for Protocol Enforcement

Think of a busy office building with different types of employees, like an accountant, an IT specialist, and a security guard. Not everyone in the building needs to access every room or every file cabinet. The accountant needs to see financial records but has no business with the server room. The IT specialist can access the server room but shouldn't be messing up with the financial books. The security guard can only access doors and cameras.

Role-Based Access Control (RBAC) is a security system that works just like this. Instead of giving every individual a long list of permissions, you assign them to a specific **role**. Each role has a pre-defined set of permissions. This makes managing who can do what much easier and safer.

In my federated learning project, we apply this concept to the communication between the IIoT devices and the server. This is what we call **protocol enforcement**.

Our framework implements **Role-Based Access Control (RBAC)** to strictly enforce the communication protocol, ensuring every message adheres to the rules. This system works by defining specific roles for devices—such as "Client Device," "Aggregator," or "Command Unit”and assigning precise permissions to each. For example, a "Client Device" is only permitted to send model updates and receive the global model, while an "Aggregator" is allowed to receive and combine updates from multiple clients.

This creates a digital security guard at the communication interface. When a message arrives, our framework checks the device's assigned role and verifies if it has permission to perform the requested action. If a "Client Device" attempts to send a command it's not authorized for, our system immediately denies the request. This powerful enforcement mechanism prevents a compromised device from carrying out unauthorized actions, keeping the entire system secure and running smoothly.

3.6 Tools and Technologies (Python, OpenSSL, Libraries)

To implement and validate our secure communication framework, we will leverage a suite of established and robust tools and technologies, primarily focusing on **Python** for its flexibility, a robust cryptographic library like **OpenSSL**, and a select set of specialized libraries.

Python will serve as the foundational programming language for this project. Its key advantages include:

Ease of Development: Python's clean syntax and high-level nature will enable rapid prototyping and implementation of complex cryptographic and networking logic. This efficiency is critical for focusing our efforts on the security framework itself rather than on low-level programming challenges.

Rich Ecosystem: Python's extensive ecosystem of scientific computing and machine learning libraries is indispensable. We will use libraries such as **TensorFlow** or **PyTorch** to instantiate the federated learning model that serves as our testbed. The use of a standard, well-documented framework will ensure that the model is a reliable tool for demonstration and that our evaluation focuses on the security overhead introduced by our framework, not on the model's performance.

OpenSSL is a powerful, open-source toolkit that provides a wide range of cryptographic functions. We will integrate OpenSSL into our Python framework to handle the core security tasks. While Python has its own built-in cryptographic libraries, leveraging OpenSSL directly provides several key benefits:

Industry Standard: OpenSSL is a widely trusted and peer-reviewed library used in countless applications and services. Relying on it ensures that our cryptographic implementations are robust, secure, and compliant with industry standards.

Performance: OpenSSL's core functions are written in C, offering superior performance for computationally intensive tasks like **HMAC** generation and key management. This is crucial for accurately measuring the performance overhead of our security framework in a realistic setting.

Essential Libraries:

- **pyOpenSSL:** This Python wrapper for the OpenSSL library will allow us to seamlessly integrate OpenSSL's powerful cryptographic functions into our Python code. This library will be used for key generation, digital signatures, and especially for implementing **HMAC** for data integrity.
- **socket (or similar networking library):** The socket library will be used for establishing and managing the TCP/IP connections between the simulated IIoT devices and the central server. This allows us to simulate the communication channel where our security framework will be applied and tested.
- **cryptography:** While we will primarily use pyOpenSSL, the cryptography library will serve as a complementary tool, providing a user-friendly and well-documented API for various cryptographic recipes. Its inclusion ensures that we have access to a broad set of modern cryptographic tools as needed.

By carefully selecting these tools, we ensure that our project is built on a foundation of reliability, security, and efficiency, allowing us to deliver a comprehensive and impactful solution to the problem of federated learning security in the IIoT.

3.7 Workflow / Flowcharts

Chapter 4 – Implementation

4.1 Secure Client–Server Setup (C2 channel)

4.2 TLS 1.3 Deployment

4.3 HMAC-based Message Validation

4.4 RBAC Enforcement for Authorized Commands

4.5 Integration with Federated Learning Communication Layer

Chapter 5 – Results and Evaluation

5.1 Security Evaluation (MitM, Replay, Tampering Tests)

5.2 Performance Evaluation (Latency, Overhead, Resource usage)

5.3 Scalability Testing (Multiple IIoT nodes)

5.4 Comparison with Existing Protocols

Chapter 6 – Discussion

6.1 Key Findings

1. **TLS 1.3 is significantly stronger than TLS 1.2**
   Our findings confirm that TLS 1.3 offers faster handshakes and removes legacy cryptographic weaknesses. By enforcing Perfect Forward Secrecy (PFS), even if future keys are compromised, past communications remain safe. This makes it a strong candidate for securing Industrial IoT (IIoT) environments.
2. **Mutual Authentication prevents unauthorized access**
   Unlike traditional TLS, which authenticates only the server, Mutual TLS (mTLS) ensures that both client devices and the server prove their identities through digital certificates. This double-check mechanism blocks rogue or fake devices from participating in Federated Learning, reducing the risk of poisoning attacks.
3. **HMAC ensures data integrity**
   While simple checksums can be easily tampered with, our evaluation shows that Hash-based Message Authentication Codes (HMAC) provide robust protection against data modification during transmission. This ensures that both commands and model updates remain untampered in IIoT communications.

4. **Role-based Access Control (RBAC) adds flexible enforcement**
   Static Access Control Lists (ACLs) lack scalability in large IIoT systems. By enforcing RBAC, permissions are assigned dynamically based on device roles (e.g., sensor, gateway, aggregator). This improves manageability and ensures that devices can only perform actions they are authorized for.
5. **Scalable and lightweight solutions are essential**
   Our analysis highlights that resource-constrained IIoT devices cannot handle heavy cryptographic operations. TLS 1.3 with optimized cipher suites and lightweight anomaly detection strikes the right balance between strong security and efficient performance at scale.
6. **Improved resilience against IIoT-specific threats**
   Combining secure communication (TLS 1.3 + mTLS), data validation (HMAC), and enforcement policies (RBAC) creates a multi-layer defense. This approach addresses common IIoT threats such as man-in-the-middle attacks, data tampering, and unauthorized device participation.

6.2 How Gaps Were Addressed

One of the biggest challenges in Industrial IoT systems was the reliance on outdated encryption protocols such as TLS 1.2, which left communication channels open to interception and impersonation. In this research, this gap was closed by adopting **TLS 1.3 with Mutual Authentication**, which not only secures data with stronger cryptographic algorithms but also ensures that both client devices and the central server prove their identities before any communication takes place. This advancement significantly reduces the risk of man-in-the-middle and spoofing attacks, which are especially dangerous in critical IIoT environments.

Another limitation in earlier systems was the weak handling of data integrity. Many relied on simple checksum methods, which could be bypassed by attackers. To address this, the solution integrates **HMAC (Hash-based Message Authentication Codes)**, which ties message verification to a secret key. This ensures that even if data packets are intercepted, they cannot be modified without detection. As a result, both model updates and command signals within the Federated Learning system remain tamper-proof and trustworthy.

Access control posed another major issue, as traditional static ACLs were rigid and difficult to scale in large IIoT networks. To overcome this, the framework replaces ACLs with **Role-Based Access Control (RBAC)**. This approach allows permissions to be assigned dynamically based on the role of each device or user. For example, a sensor node has

different privileges compared to an aggregator or an administrator. This flexibility improves both scalability and operational efficiency.

Furthermore, many IIoT systems relied on rule-based detection for threats, which often failed against sophisticated attacks. To strengthen resilience, this work incorporates **real-time anomaly detection** mechanisms into the communication process. These methods monitor device behavior and communication patterns, flagging unusual activity that could indicate malicious intent. This proactive approach enhances security without adding significant complexity.

Lastly, the issues of resource constraints and scalability were also addressed. Traditional cryptographic techniques placed heavy loads on edge devices, limiting their performance. In contrast, this solution uses **lightweight TLS 1.3 cipher suites** along with optimized anomaly detection techniques, ensuring that even resource-constrained IIoT devices can participate securely. At the same time, the framework is designed to handle **large-scale deployments** by combining TLS 1.3's streamlined handshake with RBAC's structured management. This enables secure, efficient communication across thousands of devices without degrading performance.

6.3 Limitations

While the proposed Secure Communication and Protocol Enforcement framework addresses several critical gaps in IIoT Federated Learning, a few limitations remain.

First, although **TLS 1.3 with Mutual Authentication** greatly improves security, it introduces an overhead in terms of certificate management. Every device in the network must maintain valid certificates, and large-scale deployments may face challenges in issuing, revoking, and renewing them. This administrative burden can slow down system adoption in industrial environments where thousands of devices are connected.

Second, the use of **HMAC for data integrity** provides strong protection against tampering, but it requires the secure management of shared secret keys. If these keys are not rotated frequently or are exposed through poor practices, the integrity of the entire system may be compromised. Ensuring secure key distribution in highly distributed IIoT systems remains a difficult task.

Another limitation lies in the implementation of **Role-Based Access Control (RBAC)**. While RBAC enhances flexibility compared to static ACLs, it still requires careful configuration. Incorrect role assignments or overly broad permissions may inadvertently create security loopholes. Additionally, in rapidly changing industrial setups, frequent updates to roles and permissions may introduce delays or misconfigurations.

From a performance perspective, even though TLS 1.3 is more efficient than its predecessors, **resource-constrained IIoT devices** can still struggle under heavy cryptographic operations. While lightweight cipher suites were considered, the computational cost may remain a barrier

for extremely low-power sensors or legacy devices that lack the necessary processing capabilities.

Finally, the **anomaly detection mechanisms** used to identify abnormal communication patterns are not immune to false positives and false negatives. In some cases, benign behavior may be flagged as malicious, or sophisticated attacks may go unnoticed. Achieving the right balance between sensitivity and accuracy in real-time detection continues to be a challenge.

6.4 Future Improvements

Although the proposed framework addresses key security gaps in IIoT Federated Learning, there are several opportunities to enhance its effectiveness in the future.

One promising direction is the adoption of **automated certificate management systems** to reduce the overhead of issuing, renewing, and revoking certificates required for TLS 1.3 with Mutual Authentication. Integrating lightweight, automated Public Key Infrastructure (PKI) or blockchain-based certificate validation could significantly simplify large-scale deployments where thousands of devices must maintain trusted identities.

Another area of improvement is **advanced key management for HMAC**. Currently, HMAC depends on securely distributed and periodically rotated secret keys. Future work could explore more dynamic and decentralized approaches, such as using zero-trust key exchange protocols or integrating quantum-safe cryptographic algorithms to prepare for emerging threats.

For **Role-Based Access Control (RBAC)**, scalability and adaptability can be enhanced by extending it into **Attribute-Based Access Control (ABAC)**. Unlike RBAC, which assigns static roles, ABAC considers contextual factors such as device type, location, or current operational status before granting permissions. This would make access control more flexible and context-aware in rapidly evolving IIoT environments.

To reduce the computational burden on **resource-constrained devices**, future work could focus on developing **lightweight cryptographic libraries optimized for IIoT hardware**. Hardware-assisted cryptography, energy-efficient algorithms, and offloading cryptographic tasks to edge gateways could enable even the smallest IIoT devices to participate securely.

Finally, improvements in **anomaly detection** will be critical. Current methods can suffer from false positives or undetected threats. Future implementations could leverage **machine learning-driven adaptive detection models**, which learn from network behavior over time

and improve accuracy without overwhelming system resources. Combining anomaly detection with **federated threat intelligence sharing** across devices could provide collective resilience against new and unknown attack patterns.

Chapter 7 – Conclusion

This research set out to strengthen **Secure Communication and Protocol Enforcement** in Federated Learning for Industrial IoT (IIoT) environments, where data confidentiality, integrity, and trust are essential. The proposed framework combined **TLS 1.3 with Mutual Authentication**, **HMAC-based data validation**, and **Role-Based Access Control (RBAC)** to address long-standing security gaps such as weak encryption, data tampering, rigid access control, and scalability challenges.

The findings highlight that **TLS 1.3 with Mutual Authentication** provides a robust foundation for communication security, ensuring that both devices and servers are authenticated before exchanging sensitive information. **HMAC mechanisms** further guarantee that transmitted data, including Federated Learning model updates, remains tamper-proof throughout its journey. Meanwhile, **RBAC enforcement** delivers a more scalable and flexible solution than static ACLs, adapting to the dynamic needs of industrial networks.

Despite these advances, the study also acknowledges certain limitations, such as certificate management overhead, key distribution challenges, and performance constraints on resource-limited devices. However, the results demonstrate that by carefully tailoring protocols and security mechanisms to IIoT needs, it is possible to achieve a strong balance between **security, efficiency, and scalability**.

Ultimately, this work contributes to the broader vision of making Federated Learning viable for industrial systems by securing its communication backbone. By addressing vulnerabilities at the protocol and enforcement level, the framework not only protects against current threats but also lays the groundwork for future enhancements, such as automated certificate management, context-aware access control, and AI-driven anomaly detection.

7.1 Summary of Work

This research focused on designing and implementing a **Secure Communication and Protocol Enforcement module** to strengthen Federated Learning in Industrial IoT (IIoT) systems. Recognizing that communication is the backbone of Federated Learning, the work addressed critical challenges such as weak encryption, data tampering, unauthorized access, and scalability limitations.

The study began by analyzing existing shortcomings in IIoT communication, including outdated encryption protocols, vulnerable integrity checks, and rigid access control

mechanisms. To address these issues, a multi-layered solution was proposed. **TLS 1.3 with Mutual Authentication** was deployed to ensure both confidentiality and trust between clients and servers, eliminating risks of impersonation and man-in-the-middle attacks. To secure the integrity of transmitted data, **HMAC** was integrated as a robust mechanism to prevent tampering during model update exchanges. Furthermore, **Role-Based Access Control (RBAC)** was introduced to replace static ACLs, enabling scalable and flexible access control across diverse industrial devices and users.

The implemented solution was evaluated for its ability to secure communication without overloading resource-constrained IIoT devices. Results showed that TLS 1.3's streamlined handshake reduced communication delays while still maintaining strong cryptographic protection. The addition of HMAC reinforced data trustworthiness, and RBAC provided adaptable enforcement, particularly useful in large-scale deployments.

While the framework demonstrated significant improvements, the work also recognized certain limitations such as certificate management overhead, secret key distribution challenges, and the risk of false positives in anomaly detection. These limitations informed a set of **future improvement directions**, including automated certificate management, attribute-based access control (ABAC), lightweight cryptographic libraries, and AI-enhanced anomaly detection.

7.2 Research Contributions

This research makes important contributions to strengthening the security of Federated Learning in Industrial IoT (IIoT) through the development of a Secure Communication and Protocol Enforcement framework. One of the major achievements is the integration of **TLS 1.3 with Mutual Authentication**, which provides a strong foundation for secure communication. Unlike older protocols, TLS 1.3 offers faster handshakes, stronger cryptographic suites, and Perfect Forward Secrecy. By enforcing mutual authentication, the framework ensures that both IIoT devices and the central server validate each other before exchanging sensitive data, eliminating the risks of impersonation and man-in-the-middle attacks.

Another key contribution is the enhancement of **data integrity mechanisms** through the use of Hash-Based Message Authentication Codes (HMAC). Traditional checksum methods used in industrial communication were prone to tampering, making them unsuitable for high-stakes environments. HMAC provides a stronger layer of protection, binding message verification to a secret key. This guarantees that Federated Learning model updates and command signals cannot be altered during transmission without detection.

The research also improves **access control mechanisms** by replacing static Access Control Lists (ACLs) with **Role-Based Access Control (RBAC)**. This shift makes the framework more scalable and flexible, allowing permissions to be assigned based on roles rather than rigid device lists. In large industrial networks, RBAC enables efficient enforcement, ensuring that devices and users only access resources necessary for their functions.

In addition, the solution was carefully designed to remain **lightweight and practical for resource-constrained IIoT devices**. By tailoring cryptographic operations and optimizing protocol enforcement, the framework ensures that even low-power edge devices can participate securely without performance degradation. This makes the system suitable for real-world deployment in diverse industrial environments.

7.3 Final Remarks

This research has shown that securing communication and enforcing protocols are vital steps toward building trustworthy Federated Learning systems in Industrial IoT (IIoT). By integrating **TLS 1.3 with Mutual Authentication**, **HMAC-based data integrity**, and **Role-Based Access Control**, the framework successfully addressed key gaps in confidentiality, integrity, and access management. The approach not only strengthened protection against modern cyber threats but also ensured that the solution remained practical for large-scale and resource-constrained industrial environments.

While certain challenges remain, such as certificate management and lightweight cryptography for constrained devices, this work lays the groundwork for future improvements. With further development, such as automated key and certificate handling, adaptive access control, and AI-powered anomaly detection, the framework can evolve into a highly resilient and scalable standard for secure IIoT communications.

In conclusion, this contribution represents more than just a technical enhancement—it offers a **step forward in enabling safe, reliable, and privacy-preserving adoption of Federated Learning in industry**. By strengthening the communication backbone, the research ensures that IIoT systems can confidently embrace distributed intelligence while maintaining the highest levels of security.

References

[1] V. J. S. J. D. N. R. V. P. S. B. S. Viveka, Industrial Internet of Things (IIoT), 2024.

[2] M. E. M. A. Maximilian L, Cybersecurity Management for (Industrial) Internet of Things, 2018.

[3] S. K. G. Spathoulas, Security and Privacy Trends in the Industrial Internet of Thing, 2019.

[4] H. G. T. L. Y. D. X. W. S. Z. N. G. Daojing He, Toward Hybrid Static-Dynamic Detection of Vulnerabilities in IoT Firmware, 2021.

[5] Y. Y. T. L. J. J. Q. W. Yikai Xu, Review on cyber vulnerabilities of communication protocols in industrial control systems, 2017.

[6] S. S. Z. U. Inayat Ali, Internet of Things Security, Device Authentication and Access Contro, 2019.

[7] M. D. P. P. A. S. J. L. D. N. H. P. Dinh C. Nguyen, Federated Learning for Industrial Internet of Things in Future Industries, 2021.

[8] Z. L. J. S. S. H. F. Y. Y. Y. R. G. M. C. Renuga Kanagavelu, Federated Learning for Advanced Manufacturing Based on Industrial IoT Data Analytics, 2021.

[9] M. D. P. P. A. S. J. L. D. N. H. P. Dinh C. Nguyen, Federated Learning for Industrial Internet of Things in Future Industries, 2021.

[10] T. F. H. Tschofenig, Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things, 2016.

[11] F. G. M. Fischlin, European Symposium on Security and Privacy, 2017.

[12] H. W. H. Krawczyk, European Symposium on Security and Privacy, 2016.

[13] J. L. J. P. J. B. K. W. M. H. Markus Dahlmanns, Asia Conference on Computer and Communications Security, 2022.

[14] K. D. B. H. P. Koopman, Selection of Cyclic Redundancy Code and Checksum Algorithms to Ensure Critical Data Integrity, 2015.

[15] G. M. R. M. F. B. S. J. R. A. B. S. M. M. T. C. S. J. B. Alireza Esfahani, "A Lightweight Authentication Mechanism for M2M Communications in Industrial IoT Environment," *Internet of Things ,* 2019.

[16] D. H. C. S. V. F. E. Bertin, Access control in the Internet of Things: a survey of existing approaches and open research questions, 2019.

[17] J. O. B. R. S. Safwa Ameer, Hybrid Approaches (ABAC and RBAC) Toward Secure Access Control in Smart Home IoT, 2023.

[18] B. H. J. C. T. W. H. Boyes, The industrial internet of things (IIoT): An analysis framework, 2018.

[19] J. A. M. B. A. U. E. J. A. Atutxa, Improving efficiency and security of IIoT communications using in-network validation of server certificate, 2022.

[20] F. A. I. Z. Shams Shapsough, Securing Low-Resource Edge Devices for IoT Systems, 2018.