I.J. Information Technology and Computer Science, **2026, 2, 202-221**
Published Online on April 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2026.02.12

Mobile OTP Authentication Protocol Design and Implementation for Local Federated Clients to Federated Central Server via MQTT

Narendra Babu Pamula*
Department of Computer Engineering, Mizoram University Tanhril, Aizawl Mizoram, India
E-mail: naren.pamula@gmail.com
ORCID iD: https://orcid.org/0000-0002-6339-2676
*Corresponding author

Ajoy Kumar Khan
Department of Computer Engineering, Mizoram University Tanhril, Aizawl Mizoram, India
E-mail: ajoyiitg@gmail.com
ORCID iD: https://orcid.org/0000-0003-1568-7872

Arnidam Sarkar
Department of Computer Science & Electronics, Ramakrishna Mission VidyamandiraBelur Math Howrah, WB, India
E-mail: arindam.inspire@gmail.com
ORCID iD: https://orcid.org/ 0000-0003-0842-568X

Received: 11 November 2025; Revised: 01 January 2026; Accepted: 14 February 2026; Published: 08 April 2026

Abstract: Strong and effective authentication methods are more important than ever in the ever-changing field of cybersecurity. In this work, a Mobile One-Time Password (OTP) Authentication Protocol designed for local federated clients utilizing the Message Queuing Telemetry Transport (MQTT) protocol to communicate with a federated central server is designed and implemented. This protocol strengthens the security foundation of federated systems by ensuring the safe and dependable delivery of OTPs while utilizing the lightweight and effective characteristics of MQTT. The suggested protocol tackles the scalability, security, and latency issues that come with federated setups. We show how the protocol can effectively mitigate possible security threats, like replay attacks and illegal access, while maintaining user convenience through a thorough analysis and implementation. Our protocol strikes a balance between security and performance, according to experimental results, which makes it a workable answer for modern federated authentication requirements.

Index Terms: Mobile OTP Authentication, One-Time Password (OTP), Federated Learning, Central Server, MQTT Protocol, Secure Communication.

1. Introduction

Object Digital security is of the utmost importance nowadays, and authentication procedures are essential for protecting private data and preserving user privacy [1]. The rising susceptibility of traditional authentication mechanisms, such passwords, to diverse cyber-attacks has prompted the need for more resilient alternatives. Systems that use one-time passwords (OTPs) have become a strong substitute as they offer a dynamic, temporary authentication code that greatly improves security. The development of federated systems and the Internet of Things (IoT) has further complicated the field of digital security. Multiple devices and entities can work together without exchanging raw data thanks to federated learning and federated systems, which protect privacy while utilizing distributed computational resources. However, these systems' dispersed architecture creates special difficulties for safe communication and authentication [2]. A lightweight publish-subscribe network protocol called Message Queuing Telemetry Transport (MQTT) has become well-known due to its effectiveness in confined spaces like the Internet of Things. It is a great choice for federated systems communication since it works well with low-bandwidth, high-latency, or unstable networks. In order to facilitate safe communication between local federated clients and a federated central server via MQTT, this paper introduces a unique

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

Mobile OTP Authentication Protocol. By addressing the particular security issues raised by federated systems, the suggested protocol seeks to preserve MQTT's effectiveness and lightweight nature while guaranteeing safe authentication and data integrity.

This article makes three contributions:

CB1: Creation of a Mobile OTP Authentication Protocol: A thorough protocol that uses MQTT to provide safe OTP authentication in federated systems.

CB2: Application and Verification: Sophisticated implementation and thorough verification of the suggested protocol within a federated environment simulation.

CB3: Performance Analysis: A detailed examination of the protocol's efficiency in terms of computing, communication overhead, and security resilience. Secure authentication methods are essential for safeguarding private data and securing sensitive information in the digital era. Strong authentication techniques that can function well in dispersed and resource-constrained situations are becoming more and more necessary as mobile devices and Internet of Things (IoT) systems proliferate. An example of one of these environments is a federated learning setup, in which a number of clients (often local servers or edge devices) work together to train a common model without providing raw data. By maintaining data decentralization, federated learning improves security and privacy. However, special security issues, especially with regard to authentication and data integrity, arise when local clients communicate with a central server. The Message Queuing Telemetry Transport (MQTT) protocol is frequently utilized in this setting to facilitate effective, lightweight communication. Because of its minimal overhead and simplicity of implementation, MQTT is frequently used in Internet of Things systems. However, because it lacks built-in security capabilities, additional security layers must be implemented. The well-known One-Time Password (OTP) authentication method secures communication routes. It reduces the dangers of password reuse and phishing attempts by providing a time-sensitive, one-of-a-kind password for every login session. By using MQTT to implement OTP, federated learning systems may make sure that every client-server communication is properly authenticated, shielding the system from any data breaches and illegal access.

There are several reasons for developing and putting into practice a Mobile OTP Authentication Protocol for local federated clients utilizing MQTT to communicate with a federated central server.

R1: Enhancement of Security: Securing the communication channel between local clients and the central server is crucial in federated learning systems. By ensuring that every interaction is validated, OTP lowers the possibility of unwanted access.

R2: Lightweight and Effective Communication: MQTT is selected because it works well in mobile and IoT situations where power and bandwidth are issues. MQTT's lightweight design fits very nicely with federated learning setups that require effective communication.

R3: User Convenience: Creating and authenticating OTPs via mobile devices offers a convenient user experience. Due to their widespread use, mobile devices make it simple for users to get and use OTPs for safe access.

R4: Scalability: A federated learning system's authentication process must expand to accommodate the system's potential growth in the number of devices. OTP-based authentication can support a high volume of users, so even as the system grows, it will stay effective and safe.

R5: Data Privacy: Federated learning naturally preserves data privacy by storing the data on local devices and only exchanging model changes. An extra degree of security is added by the OTP authentication, which guarantees that only authorized devices take part in the learning process.

Important elements of the implementation include error recovery, logging, and access control, handling of token expiration and renewal, token validation, and monitoring.

Unique challenges in Mobile OTP Authentication Protocol using MQTT

Making a Mobile One-Time Password (OTP) login system for local federated clients and putting it into action Using the Message Queuing Telemetry Transport (MQTT) system to talk to a federated central server can be tricky for a number of reasons.

Security Flaws in MQTT: MQTT, which is used a lot in Internet of Things (IoT) apps, doesn't have any built-in secure authentication methods. A lot of the time, it rests on simple authentication like a username and password sent in plain text, which can be read by anyone.

IoT Devices with Limited Computing Power: A lot of IoT devices don't have a lot of computing power, which makes it hard to set up complex security measures. To ensure both security and efficiency, authentication systems must be simple

and effective.

Scalability in Federated Architectures: It's hard to manage identity across distributed MQTT brokers in a federated system with many clients and servers. To keep the system's security, it's important to make sure that all of these brokers use the same, safe authentication methods [3].
Protecting privacy: To use OTP authentication, you need to be careful with user info so that privacy isn't broken. It is very important to make sure that the authentication methods don't leak private data [4].

Integration with Existing Systems: When adding OTP mechanisms to MQTT-based systems that are already in place, they need to be easily integrated so that current activities are not interrupted. This includes making sure that the new system works with the old ones and making as few changes as possible to the way the system is built. This resource is very helpful for learning how to make MQTT-based IoT systems safer by using OTP mechanisms in federated designs. Secure authentication methods are essential for safeguarding private data and securing sensitive information in the digital era. Strong authentication techniques that can function well in dispersed and resource-constrained situations are becoming more and more necessary as mobile devices and Internet of Things (IoT) systems proliferate. An example of one of these environments is a federated learning setup, in which a number of clients (often local servers or edge devices) work together to train a common model without providing raw data. By maintaining data decentralization, federated learning improves security and privacy. However, special security issues, especially with regard to authentication and data integrity, arise when local clients communicate with a central server. The Message Queuing Telemetry Transport (MQTT) protocol is frequently utilized in this setting to facilitate effective, lightweight communication. Because of its minimal overhead and simplicity of implementation, MQTT is frequently used in Internet of Things systems [5]. However, because it lacks built-in security capabilities, additional security layers must be implemented. The well-known One-Time Password (OTP) authentication method secures communication routes. It reduces the dangers of password reuse and phishing attempts by providing a time-sensitive, one-of-a-kind password for every login session. By using MQTT to implement OTP, federated learning systems may make sure that every client-server communication is properly authenticated, shielding the system from any data breaches and illegal access.

The remaining sections of this essay are arranged as follows: The related work on OTP authentication and MQTT-based federated systems is reviewed in Section 2. The architecture and design of the suggested Mobile OTP Authentication Protocol are described in depth in Section 3. The implementation and validation procedure are described in Section 4. The performance and security of the protocol are examined in Section 5. The paper is finally concluded in Section 6, which also addresses possible future work.

2. Literature Survey

[6] (Junpeng Zhang, 2022) The paper highlights the critical security and privacy concerns inherent in federated learning (FL) systems. It identifies key threats such as data leakage, inference attacks, and adversarial attacks, where malicious actors can exploit model updates to infer sensitive information or corrupt the training process. The paper categorizes FL into cross-device and cross-silo models, each with distinct vulnerabilities and solutions. It discusses methods like differential privacy, secure multiparty computation, and homomorphic encryption to enhance data security and privacy. The study also emphasizes the need for robust aggregation mechanisms and incentive structures to motivate participant cooperation while ensuring security. Overall, it underscores the necessity of developing comprehensive security frameworks to safeguard FL systems from emerging threats.

[7] (Gorka Abad, 2022) It draws attention to the main privacy risks, including inference attacks, backdoor attacks, and data poisoning. The paper discusses a number of privacy-preserving techniques to address these problems. In order to safeguard individual data points, differential privacy introduces noise into model updates. To make sure that no individual data inputs are revealed during the aggregation process, Secure Multiparty Computation (SMPC) is used. Homomorphic encryption preserves privacy during processing by enabling calculations on encrypted data. Furthermore, model updates are aggregated in a privacy-preserving way by Federated Averaging (FedAvg). In order to ensure the safe deployment of FL systems, the paper highlights the need for continued research to create more reliable aggregation algorithms and adaptive mechanisms to detect and respond to malicious behaviour.

[8] (Gosselin, Vieu, Loukil, & Benoit, 2022) The survey discusses various defense tactics like differential privacy, secure multi-party computation, and homomorphic encryption that aim to reduce these risks. Additionally, it looks at ways to manage heterogeneous data, optimize model aggregation, and guarantee fairness in various client environments. In order to improve the security and dependability of federated learning systems, the paper indicates future research directions and highlights the significance of strong privacy-preserving techniques.

[9] (H. Brendan McMahan, 2023) Using a model averaging mechanism, the paper proposes to aggregate local updates from individual devices to improve the overall accuracy and convergence speed of federated learning systems. By averaging the parameters of locally trained models instead of directly aggregating updates, the approach aims to address common issues with federated learning setups, such as non-IID (independently and identically distributed) data and device heterogeneity. The experimental results reported in the paper most likely illustrate how effective the proposed approach is when compared to other federated learning techniques.

[10] (Martín Abadi, 2016) It investigates how deep learning models can incorporate privacy-preserving techniques, particularly differential privacy. The goal of the paper is to preserve model accuracy while protecting sensitive information found in training data. Adding noise to gradients or parameters during training to achieve differential privacy guarantees is probably one of the key contributions. It shows experimental results that illustrate the trade-off between model accuracy and privacy, illustrating how various noise levels impact performance on tasks like natural language processing or image classification. The suggested strategy is also contrasted with other deep learning privacy-preserving techniques in the results. Overall, the paper offers guidance on how to balance privacy concerns with model performance when utilizing differential privacy in deep learning tasks.

[11] (Kandati DR, 2023) This paper demonstrates that even though federated learning avoids exchanging real data, protection can still be applied to parameter values that are disclosed during training or to an ML model that is generated. This research looks at FL's privacy and security concerns and addresses a number of privacy protection and safety-related issues that arise when creating FL systems.

[12] (Jean-Paul A. Yaacoub, 2023) This paper offers a thorough analysis of security and privacy threats, constraints, and difficulties. It also presents and evaluates new approaches to lessen the impact of these issues and lower the probability that they will arise. For this reason, a number of taxonomies were also put forth to describe the various dangers, assaults, and defenses specific to each FL-IoT component.

[13] (Michalek J, 2024) This paper introduces a new and extensive communication protocol and federated learning framework for Android intended especially for emergency management applications. This framework addresses the unique challenges that these types of systems bring, such as limited processing capacity, a multitude of data sources, and strict security requirements. This framework, which incorporates adaptive learning algorithms and allows collaborative model training while preserving data privacy, has the potential to significantly advance the field of federated learning within critical infrastructure systems.

[14] (Sirohi D, 2023) The fundamentals of the FL concept, as well as significant threats, vulnerabilities, and attacks in the FL environment to air, space, ground, and underwater communication scenarios, are covered in this survey paper. Additionally, we carry out an extensive, thorough analysis of the privacy and security threats and countermeasures in the FL landscape. In the aforementioned environments, mass adoption of FL is facilitated by the ability to identify these threats. Our work is therefore devoted to a thorough survey of most of the security and privacy mitigating techniques that have been proposed over the years; we have included FL-based applications areas in space, air, ground, and underwater without limiting ourselves to any particular field, area, or domain.

[15] (T. Eltaras, 2023) A secure aggregation protocol that can be used in a federated learning environment was proposed in this study. The protocol relies on the usage of auxiliary nodes, which are practically incapable of working together. In addition, using auxiliary nodes lowers costs for clients and the service provider in terms of computation, communication, and storage on low-resource devices. These auxiliary nodes can stand in for banks in banking or financial application, hospitals in a healthcare scenario, etc. Compared to cutting-edge protocols at the client nodes, the protocol analysis revealed savings in computation, communication, and storage costs.

Table 1. Existing OTP systems using MQTT or federated learning protocols compare to the proposed solution

Characteristic	Secure Existing OTP Networks Through MQTT	Protocols for Federated Learning: Authentication	Mobile OTP Authentication for Local Federated Clients via MQTT
Model for Security	Security is typically increased with a username/password combination that also includes a one-time password (OTP)	Implements decentralized authentication, occasionally including blockchain technology	Upgrades security using federated authentication and one-time passwords
Data Privacy	Simple encryption, which can be vulnerable while in transit	Differential privacy and other robust privacy-preserving technologies	Ensuring the security of one-time passwords while minimizing data exposure
Efficiency in scaling	Does a poor job on massive networks but works fine in smaller ones	Federated architecture allows for good scalability	Built to efficiently manage a large number of federated clients
Time lag	Very low because it is lightweight, however it is susceptible to delays on the network.	A little bit more because we aggregated ML models.	Aimed specifically at mobile and IoT networks for use in real-time authentication
Complicated Integration	Easy to implement, but not strong enough for widespread use	Updates to AI models and secure data sharing are necessary	Strikes a balance between effective security measures and ease of integration
Adequacy for the Use Case	Recommended for basic authentication requirements related to the Internet of Things	Apt for AI-powered projects that necessitate group education	Federated networks are perfect for mobile and Internet of Things devices
Protocol Overhead	Sparse, with no sophisticated options	Increased as a result of security measures and FL model sharing	Designed for optimal performance while maintaining a low load on system resources

[16] (Ji, 2023) In this paper, the LAFED federated learning framework is proposed for block chain enabled federated learning. It consists of a lightweight authentication mechanism and an adaptive robust aggregation method. LAFED includes privacy protection for local training as well as model aggregation. LAFED trains models locally using differential

privacy, adjusting the intermediate parameters' noise level based on the results of the model quality evaluation.

[17] (Ahmed Imteaj 1, 2021) We present a FL model in this paper that performs well in a resource-constrained IoT environment. The pruning mechanism, activity, resource awareness, and generalization FL objective functions are utilized to create a lightweight FL model that is capable of handling statistical and system heterogeneity. We obtain a strong, stable, and consistent FL model with exceptional performance in an unpredictable heterogeneous network by selecting dependable and skilled clients, training locally using a lightweight model, and permitting varying degrees of work from FL clients. Using a variety of datasets, we evaluated our FedPARL framework and found that it performed better in terms of convergence than FL techniques currently in use, which are based on the idea of realistic heterogeneous settings.

[18] (Abbas Yazdinejad, 2021) This paper aimed to tackle ML-specific issues such as privacy-preserving, scalability, and central training process in drone authentication. Our proposal involves the use of RF characteristics of drones in IoT networks to create a federated drone authentication model. By using the FL technique to train data locally, the suggested model preserves privacy while making use of decentralized data produced by drones. When compared to non-federated ML models, the suggested DNN architecture performs better and can enhance drone authentication performance. Additionally, as new drones join the network, the suggested model can continuously learn and update itself to maximize performance.

3. Design And Architecture of The Recommended Mobile OTP Authentication Protocols

The purpose of the Mobile OTP Authentication Protocol is to use One-Time Password (OTP) techniques over the MQTT [19] (Message Queuing Telemetry Transport) protocol to securely authenticate local federated clients with a federated central server. For federated systems, this architecture guarantees a reliable, scalable, and effective authentication procedure as shown in **FIGURE 1.** Since MQTT (Message Queuing Telemetry Transport) is lightweight, scalable, and has an effective communication paradigm, it is especially well-suited for federated systems, including federated learning. The reasons why MQTT is more suited for federated systems than other communication protocols are explained in detail below TABLE 2.

Communication that works well for distributed nodes:

- MQTT: It has a publish-subscribe approach that cuts down on direct peer-to-peer connections and unnecessary data transmission. This makes it scalable for federated systems.
- Other protocols, like HTTP, WebSockets, gRPC, and CoAP: Often rely on point-to-point connection (like HTTP REST), which adds time and cost.

Limited Data Transfer Requirements

- IoT and mobile devices with limited resources are perfect candidates for MQTT since it uses lightweight messages with minimum headers.
- There is more overhead with HTTP/gRPC because of all the extra headers and metadata.

Communication that is asynchronous and decoupled

- When you use MQTT, clients don't have to stay connected to the server all the time. This saves power and computer resources.
- Others: A lot of them, like WebSockets, need permanent links, which take up more space and time.

Federated Learning Scalability

- MQTT makes it easy for federated systems to grow because many clients (local models) can send changes to a central server (aggregator) on their own.
- There are more complicated methods that make it harder to manage how several clients talk to each other. For example, every time HTTP changes, a new API call is needed.

Internet of Things (IoT) and Mobile-Friendly Intermittent Connection Management

- Last Will and Testament (LWT) and message keeping are supported by MQTT. This means that FL clients can pick up training where they left off.
- Other Protocols: HTTP sessions end when the link is lost and need to be fixed by hand.

Better use of network resources

- MQTT uses small payloads and continuous TCP connections, which makes it easier to use than HTTP, which has a lot of request/response cycles.
- Other Protocols: HTTP and WebSockets may not work well in places with limited resources.

MQTT is ideal for federated systems, such as federated learning, due to its scalability, pub/sub paradigm, lightweight design, and support for asynchronous communication. The benefits of this protocol over others, such as HTTP, WebSocket, CoAP, and AMQP.

Table 2. In Federated Learning, a Comparison with Other Protocols

Protocols	Restrictions
HTTP/HTTPS	Synchronous communication, high overhead, and no native QoS support
WebSocket	Requires persistent connections, which can be resource-intensive for large-scale federated systems
CoAP	Designed for point-to-point communication, making it less scalable for federated systems.
AMQP	Heavier and more complex than MQTT, making it less suitable for resource-constrained devices.

3.1. Overall Architecture

- Local Federated Clients: There are a few essential phases in a Mobile OTP Authentication Protocol that involve Local Federated Clients and a Federated Central Server via MQTT (Message Queuing Telemetry Transport [20],[21]. The following describes the communication between the central server and the local federated clients. Every local client creates an account on the Federated Central Server by supplying the required authentication information (email address, phone number, etc.).
- Federated Central Server: Each client is given a unique identifier by the server, which safely retains these details. The server creates a One-Time Password (OTP) in response to a local federated client requesting access or requiring authentication. To improve security, this OTP is time-sensitive and only valid for a brief period of time.

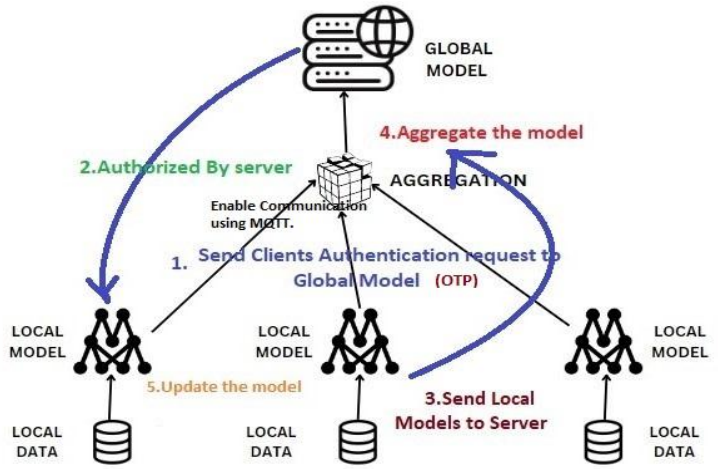


Fig.1. Centralized Federated Learning Model Architecture with OTP authentication process and MQTT Secure Process

3.2. OTP Transmission

- Federated Central Server: The server creates a One-Time Password (OTP) in response to a request for access or authentication from a local federated client. To improve security, this OTP is time-sensitive and only valid for a brief period of time. The local federated client receives the OTP [22]. You can accomplish this through a number of channels, including the app itself, email, and SMS.
- MQTT Protocol: This OTP message can be sent to the local client using MQTT. Due to its effective message delivery and minimal bandwidth consumption, MQTT is a lightweight messaging protocol that is perfect for mobile and Internet of Things applications.

3.3. Authentication Flow

A number of security precautions should be taken when developing a Mobile OTP Authentication Protocol for Local Federated Clients to Federated Central Server over MQTT in order to guarantee the integrity and security of the system. Consider the following important security precautions.

A. Channels of Secure Communication

- TLS/SSL: To encrypt data sent between mobile clients and the central server, use Secure Sockets Layer (SSL) or Transport Layer Security (TLS). This stops people from listening in and interfering [23].
- MQTT over TLS: To protect the message transport layer, make sure MQTT communications take place over TLS.

B. Generation and Management of OTP

- Robust OTP Techniques: To create OTPs, use reliable OTP algorithms like HMAC-based One-Time Password (HOTP) or Time-based One-Time Password (TOTP) [24]. Make that the algorithm is long enough to withstand

brute-force attacks and contains a secure hash function here we can show the potential federated authenticators shown in the FIGURE 2.

- OTP Expiration: To restrict the length of an OTP's validity, set an expiration time. This shortens the time frame during which a hacker can abuse an OTP they have obtained.
- Rate Limiting: To stop brute-force assaults on OTP input, use rate limiting.

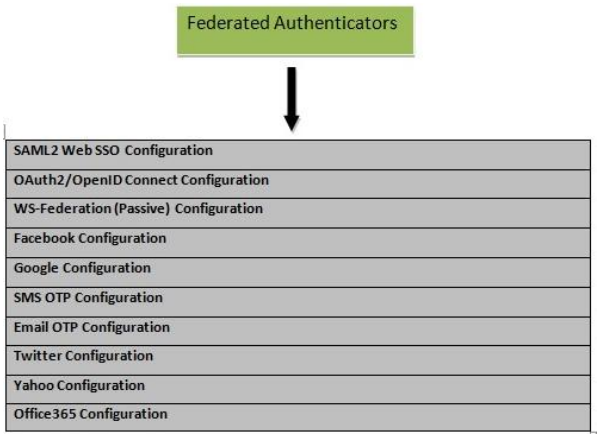


Fig.2. Centralized Federated Learning Model Architecture with OTP authentication process and MQTT Secure Process

C. Authentication Flow Challenge-Response Protocol

- Challenge-response system: in which the client must calculate the OTP after the server issues a challenge (such as a random number or nonce). This guarantees that every OTP is specific to a given session.
- Mutual Authentication: To stop man-in-the-middle attacks, make sure the client and server authenticate one another.

D. Security for Clients and Servers

- Safe Storage: On the client and server, store private information such as keys and OTP secrets in a secure manner. When it is feasible, use secure storage options or hardware security modules (HSMs).
- Use client-side validation: to ensure that messages received from the server are authentic and have not been tampered with.

E. Coordinated Security Protocols

- Federated Identity Management: To securely maintain user identities across several federated systems, apply federated identity management strategies. Make that federated tokens are correctly validated and signed.
- Strict access: restrictions should be put in place to guarantee that the federated network can only be used by approved clients and servers.

4. Implementation Details

Federated learning (FL) is a machine learning technique that maintains data privacy by allowing decentralized data training while retaining local data on devices (clients) and only exchanging model updates with a central server. Authentication in federated learning is crucial to ensure that only legitimate clients and servers can participate in the process, thereby safeguarding against malicious actors. Below is a detailed overview of client and server processing in federated learning architecture authentication. In our system we want to implement the robust authentication in federated learning.

4.1. Technologies and Tools

A multifaceted strategy combining PKI, secure communication protocols, MFA, blockchain technology, secure key management, and ongoing monitoring and improvement is needed for robust authentication in federated learning. Federated learning systems can prevent unwanted access and preserve the integrity of the distributed learning process by putting these tactics into practice and ensuring safe and reliable interactions between clients and servers. Authentication in federated learning is a multi-step procedure that includes signed model updates, encrypted communication, mutual certificate validation, and secure key generation. To guarantee the authenticity and integrity of all participants and the data being transferred, the server and clients go through stringent verification procedures. **FIGURE 3** shows the communication mechanism between the client and server in a federated system [25]. This mechanism is crucial for federated learning systems to remain private and secure.

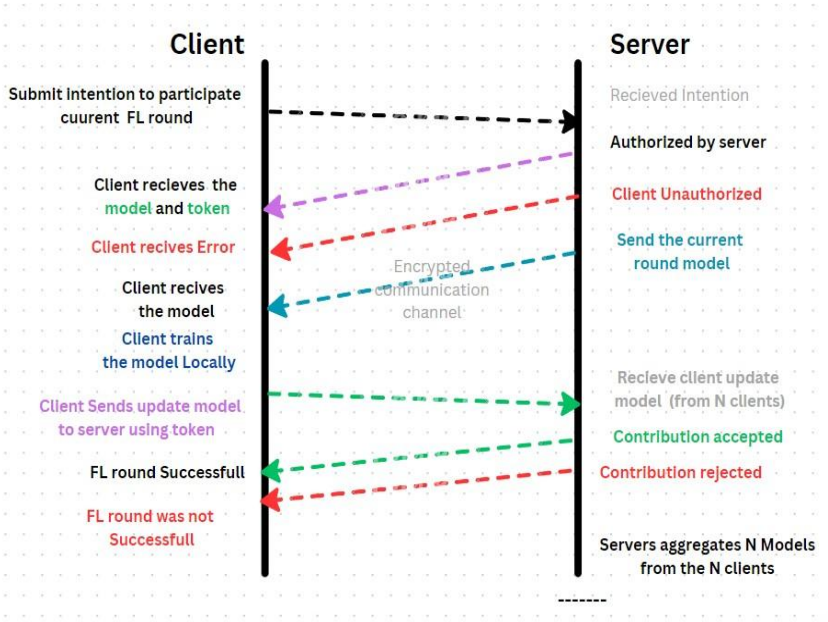


Fig.3.Centralized client-server communication channel in federated learning

4.2. Algorithm for OTP Generation

An individual must first undergo authentication, which is the process of identifying and validating themselves, before they can access any protected service (like a personal account). The purpose of authentication, which is included in cyber security standards, is to stop unwanted access to resources that are protected. Nowadays, authentication procedures establish a two-tier gateway before releasing any confidential data. This two-factor authentication, also known as a double layer of security, establishes a process that starts with the creation and verification of a One Time Password (OTP), and then requires the validation of credentials (password and username/email). During every authentication event, a unique and randomly generated numerical code known as the OTP is generated.

Algorithm 1 for One Time Password (OTP)

Step 1: Generate a Random Secret Key

i. The secret key is a tuple consisting of 16 bytes, denoted as secret_key = (byte1, byte2, ..., byte16).
ii. Each byte byte_i is generated independently using a random function: byte_i = random (i), where i = 1, 2, ..., 16.
iii. secret key = (random (1), random (2), ..., random (16))

A 256-bit secret key can be generated using a cryptographic hash function (SHA-256) applied to a random seed value:

$$"Secret_Key = Base32Encode(HMAC_SHA256(Random_Seed, System_Entropy))"$$

where:

- Random_Seed = A cryptographically secure random number.
- System_Entropy = A unique identifier (e.g., CPU time, device ID) for additional randomness.
- HMAC_SHA256 = Hash-based Message Authentication Code (HMAC) using SHA-256.
- Base32Encode = Converts binary hash output into Base32 format for easy storage and transmission.

Step 2: Generate a Random OTP

i. otp = random_sequence_generation (allowed characters, length)
ii. Define a set of allowed characters for the OTP (e.g., digits 0-9).
iii. Use the secret key as a seed or input to a random sequence generator.
iv. Generate a random sequence of characters of specified length (default is 4 digits).

$$OTPserver = trunc(\text{HMAC_SHA256}(\text{Secret_Key}, \text{Time_Counter}\ )\ )\text{mod } 10\text{^}4$$

Step 3: Simulate Sending the OTP to the User

i. Simulate the transmission of the OTP to the user through a secure communication channel.

$$Time_Counter = Timestep\ (\ Current_Unix_Timestamp_T0)/Time\ stamp$$

where Timestep = 30 seconds (default).

Step 4: Prompt the User for OTP Input

i. Display a message requesting the user to input the received OTP.

Prompt="Enter the OTP received:

ii. The user then inputs a value:

OTPuser_input=UserInput (Prompt)

Step 5: Read the User's OTP Input

i. Capture the user's input from the input interface (e.g., keyboard).

The user manually inputs an OTP as:

OTPuser=input
where OTP_user is a 4-digit integer.

Step 6: Compare User's Input with OTP

i. user_input == otp
ii. Compare the user's input with the generated OTP.

To validate the OTP, we compare the user-entered OTP with the server-generated OTP:
Valid= 1, if $OPT_{user} = OTP_{server}$ within valid time window then client is authenticated and can submit FL model updates
Invalid=0, otherwise

iii. Ensure strict equality to match both value and type.

Step 7: Grant Access if OTP Matches.

iv. For TOTP, the valid time window is typically ±1 time step (±30 seconds)

OTPServer € {OTP(t-1), OTP(t), OTP(t+1)}

4.3. MQTT Communication Setup

We can take advantage of each protocol's advantages to implement an OTP authentication mechanism in the context of federated learning using both the HTTP (Hypertext transport protocol) and MQTT (Message Queuing Telemetry Transport) communication protocols.While MQTT, a lightweight messaging protocol made for resource-constrained devices and unstable networks, is frequently used in Internet of Things (IoT) applications, HTTP is widely used for web applications. You can guarantee secure OTP-based authentication for federated learning systems that use the HTTP and MQTT protocols by putting these mechanisms in place. Which protocol to use for federated learning authentication—HTTP or MQTT—depends on the particular needs of the application, including low latency, effective bandwidth utilization, simplicity of implementation, and scalability. For web-based apps, HTTP is a good option because it is more widely supported and generally easier to implement. For situations like IoT devices taking part in federated learning, where low bandwidth and persistent connections are essential, MQTT is more effective. Here we want to show the comparative analysis of the HTTP and MQTT Protocols based on some aspects which is considering the federated learning authentication perspective in security manner.

In TABLE 3 comparative analysis of theses protocols here in this authentication process we are selected MQTT protocol. Because a new set of characters is generated for the password each time an authentication attempt is made, it provides an extra layer of security and has the added benefit of being unpredictable for the next created session. There are two primary ways to deliver the OTP.

- Text Message Based: This is very simple. After successful regular authentication, it is standard procedure to send the OTP via text message. In this instance, the authenticator receives the OTP via text message after it is generated on the server side. It is the most widely used technique for delivering OTP across all services.
- Application-Based: This technique generates OTP on the user's end by scanning a QR code on the screen with a particular smart phone app. The application is in charge of the distinct OTP digits. In comparison to delivery via SMS, this lowers security risk and shortens the OTP wait time.

Table 3. Comparative analysis of HTTP & MQTT protocol

Feature	HTTP Protocol	MQTT Protocol
Nature	Request-response	Publish-subscribe
Transport	TCP	TCP (or Web Sockets)
Connection	Stateless, short-lived	Stateful, persistent
Latency	Higher due to setup/teardown	Lower due to persistent connections
Bandwidth	Higher due to message size	Lower, efficient use
Encryption	TLS/SSL	TLS/SSL
Authentication	OAuth, JWT, Basic Auth, etc.	Username/password, client certificates
Complexity	Easier to implement, widely supported	More complex, requires a broker
Federated Learning	Robust web interfaces, initial setup	Efficient for resource-constrained devices

4.4. Possible MQTT Drawbacks in Large-Scale Systems with Federated OTP Authentication

Message Queuing Telemetry Transport, or MQTT for short, is a popular messaging protocol for low-bandwidth, high-latency settings, such as the Internet of Things (IoT). Although there are numerous benefits to using MQTT, such as rapid communication and low overhead, there are also some possible downsides, especially when it comes to handling large-scale message processing and ensuring that messages are delivered. The following is a breakdown of various problems and how to fix them.

A. Guarantees on Message Delivery

Although MQTT offers three tiers of Quality of Service (QoS), each comes with some compromises:

QoS 0 ("At most once") - Messages could be lost in transit, so unreliable for occurrences of essential authentication.
QoS 1 ("At least once") - Messages are assured to be delivered but may be duplicated, resulting in possible inconsistent authentication.
QoS 2 ("Exactly once") ensures communication uniqueness but adds extra overhead that could slow down OTP authentication.

- Use QoS 2 selectively for crucial authentication communications, then maximize acknowledgment systems to lower processing delays.

B. Problems of Scalability in Large-Scale Systems

Broker MQTT routes messages using a centralized broker, therefore creating a bottleneck. The broker may develop a performance bottleneck under heavy traffic loads, which may cause delays in OTP delivery. Unlike message queues like Kafka, MQTT brokers do not naturally allow distributed processing, which makes it difficult to expand over several sites.

- Load balancing across several MQTT brokers or a hierarchical MQTT design will help to effectively disperse traffic.

C. Managing High Latency in Systems of Federated Learning

MQTT may find it difficult to guarantee timely OTP transmission in high-latency networks, particularly in federated authentication when clients are scattered geographically. Delayed retransmissions or network problems could cause expired OTPs, hence raising the authentication failure rates.

- Edge brokers handle OTP requests closer to clients, therefore lowering latency.
- Retain communications and maximize keep-alive intervals to strengthen unstable networks.

Due to its centralized broker architecture, message delivery guarantees, and limited scalability, MQTT is perfect for federated authentication. However, its lightweight design makes it hard to use in settings with a lot of latency. Broker load balancing and edge computing are two technologies that are needed to get around these problems and make OTP

verification work.

5. Security Analysis and Validation Procedure

In Time-Based One-Time Passwords (TOTP), which are a Time Synchronized OTP as defined by The Initiative for Open Authentication (OATH), are the most widely used technique for creating One Time Passwords (OTPs). In these OTP systems, time is an essential factor in generating a unique password [26]. The password is created by multiplying the current time by a secret key. Implementation of secure authentication mechanisms to ensure the integrity and security of communication between clients and the server.

5.1. Threat Model

A threat model comparative analysis is assessing possible security threats and dangers related to the system in the context of Mobile OTP (One-Time Password) authentication Protocols for local federated clients communicating to a federated central server over MQTT. Here is a summary of typical dangers and how they can be mitigated by various designs or implementations shown in TABLE 4.

Table 4. Comparative Study of Threat Models

Components of the Threat Model		**Typical Attacks and Threats**	**Strategies for Mitigation**
Risk Data Information [27]	OTPs, user passwords, sensitive data, system availability, and integrity.	Eavesdropping and Man-in-the-Middle (MitM) Attacks	TLS/SSL Encryption & Mutual Authentication
Threat Objects [28]	Attackers: Malicious users, external attackers, insiders. Valid User: End-users & administrators	Replay Assaults	Nonces and Timestamps & OTP Expiry
		Insider Threats	Access Control & Audit Logs
Threatening Positions [29]	Federated servers, mobile devices, and MQTT brokers	Theft of Credentials and Impersonation	Secure Storage & Multi-Factor Authentication (MFA)
Risk actions [30]	Interception, manipulation of data, impersonation and denial of service (DoS)	Denial of Service (DoS) Attacks	Rate Limiting & Load Balancing

The security of a mobile OTP authentication system employing MQTT is greatly impacted by the protocols, authentication methods, and system architecture selected. It is important to create a thorough threat model that takes into account the system's unique needs and risk tolerance. When creating a reliable and user-friendly authentication mechanism, it is essential to strike a balance between security, usability, and performance. Together with mitigating techniques, this is a comparative table highlighting the sophisticated vulnerabilities in authentication protocols for local federated clients contacting a federated central server using MQTT. Here we listed some Advanced Attacks & Mitigations in MQTT-Based Federated Authentication shown Table 5.

Table 5. Advanced attacks & mitigations in federated authentication based on MQTT

Type of Attack	**Risk Assessment**	**Mitigation Strategies**	**Optimal Security Measures**
Man-in-the-Middle (MitM)	Attackers authenticate MQTT messages.	TLS/SSL Encryption: MQTTS End-to-- End Encryption (E2EE)	Mutual Verification (client certificates) For authentication use TLS 1.3+ and ECDSA.
Unauthorized Client Access	Attackers access federated data posing legitimate customers.	Multiple Factor Verification (MFA) Device fingerprinting and HMAC-SHA256 authentication	Need biometric + OTP access.
Session Hijacking and Retreading	Attackers recycle stolen authentication tokens.	Token Expiry & Rotation (OAuth2, JWT); Time-based OTP; Nonce-based Challenge-Response	Create a short-lived JWT using refresh tokens.
Tampering Messages	Attackers change MQTT messages to control federated model updates	Digital Signatures (RSA, ECC) HMAC-SHA256, or message authentication codes Strict Access Control Policies (ACLs)	Verify low weight integrity using ECC.
Denial of Service (DoS)	Attackers bomb the MQTT broker with too many messages	Rate Limiting and Throttling Restricted QoS Level: 1 or 2 only IDS: Intrusion Detection Systems	Apply anomaly detection grounded on artificial intelligence
Federated Model Poisoning	Backdoored updates injected by malicious clients into FL models.	Homomorphic Encryption, DP; Anomaly Detection on Client Updates; Reversal Based Scoring	Apply Byzantine-robust aggregating technique.

5.2. Authentication Security

Although they have different security uses, federated learning (FL) and one-time password (OTP) authentication are tightly related in terms of privacy preservation, safe authentication, and stopping illegal access in a federated system.

A. Federated Learning and Issues of Privacy

Federated learning (FL) is a distributed machine learning method whereby several clients cooperatively train a common model without exchanging their own raw data. FL nevertheless poses security and privacy issues even if it naturally protects privacy by maintaining data localized:

- From shared model updates (gradients), adversaries can deduce sensitive information.
- Malicious consumers can send altered updates meant to destroy the global model.
- Man-in----middle (MITM) attacks let attackers intercept messages between consumers and the central server.
- Preventing illegal access depends on making sure just authorized clients engage in the training program.

B. OTP Verification's Function in Security

With OTP authentication—a security system whereby each login or transaction generates a distinct, time-sensitive password— Its main traits consist in:

- OTPs only last a limited time, so lowering the likelihood of repeat events.
- Every OTP is used just once; hence attackers find it difficult to re-use obtained credentials.
- OTPs provide extra protection over conventional login and password systems.

C. Linking Federated Learning to OTP Authentication

Including OTP authentication into federated learning can help to solve numerous security and privacy issues:

- OTP guarantees that just authenticated clients may engage in the federated learning process, therefore preventing unauthorized access. This stops malevolent users from connecting to the network and turning in contaminated updates.
- OTPs are time-sensitive, hence they lower the possibility of stolen credentials being utilized to pass for real customers.
- OTPs help to create safe channels of contact between clients and the central server, therefore mitigating MITM attacks. OTPs might be used to encrypt model updates or included into TLS handshakes, for instance.
- OTPs help to preserve the integrity of the federated learning system by guaranteeing that only authenticated clients may provide updates.
- OTPs allow clients to be authenticated without disclosing their identities, hence anonymizing and pseudoneering. This reflects FL's objective of safeguarding user privacy.
- OTPs lower the danger of adversaries gaining access to the network and extracting sensitive data from model updates by guaranteeing that only authorized clients engage.
- Fighting Data Poisoning Trusted Participation: OTPs guarantee that just confirmed clients could help the model training process. This lowers the possibility of malevolent users turning in poisoned updates.
- Should a client compromise themselves, their OTP access can be promptly canceled, therefore stopping more harm to the federated learning system.

By following the below authentication process, we are authorized our federated system.

Receive Authentication Request: The server receives an authentication request via the `/authenticate1` endpoint, which can handle both GET and POST requests.

Extract Input Parameters: Upon receiving the request, the server extracts the following input parameters from the request form data ‘username’, ‘password’.

Validate User Credentials: The server validates the user credentials by querying the database to verify if they match the stored credentials for the specified user. If the credentials are valid, the authentication process proceeds; otherwise, an error message is returned indicating that the user credentials are invalid.

The following are the steps for OTP Authentication:

Client Request: The client sends a request to the endpoint, either via a POST request.

Extract Mobile Number: If the request method is POST, the endpoint extracts the mobile number from the request form data.

Generate OTP: The `get OTP Api` function is called to generate a random four-digit OTP (One-Time Password) using the `generate OTP` function.

Send OTP: The generated OTP is sent to the provided mobile number via Twilio's SMS API. The `get OTP Api` function constructs an SMS message body containing the OTP and sends the message to the provided number using Twilio's API.

Check Message Status: Upon sending the OTP, the function checks if the message was successfully sent by verifying if a message SID is returned from Twilio's API.

5.3. Metrics for Accuracy and Performance

Several important metrics and criteria must be taken into account in order to give thorough authentication results for the Mobile OTP Authentication Protocol. This is a methodical format for presenting and evaluating these findings.

A. Success Rate (SR)

The proportion of successful OTP verifications. This covers the creation, transfer, and validation of OTPs with success. The number of successful outcomes divided by the total number of attempts or events is known as the success ratio. It is frequently employed to assess how well a system, activity, or process is working. The success ratio, which shows the fraction of successful occurrences relative to all events, is commonly stated as a percentage.

$$\boldsymbol{SR} = \left(\frac{number\ of\ OTP\ verifications\ that\ are\ usefull}{Acceptance\ of\ all\ OTP\ Verifications}\right) * 100 \quad (1)$$

B. False Acceptance Rate (FAR)

The proportion of instances in which an unauthorized user is mistakenly accepted by a system. FAR, which is the percentage of verification transactions in which bogus identity claims are inadvertently confirmed, is calculated as the ratio of false acceptances to identification attempts. FAR is an important statistic for evaluating the accuracy and potency of biometric security systems, like fingerprint scanners or facial recognition software. A system with a lower FAR number is considered more dependable, whereas a larger FAR value suggests a higher danger of unauthorized access.

$$\mathbf{FAR} = \left(\frac{\text{The amount of unlawful access that is permitted}}{\text{Total illegal access attempts}}\right) * 100 \quad (2)$$

C. False Rejection Rate (FRR)

The proportion of instances in which an authorized user is mistakenly rejected by a system. A measure called false reject rate determines how frequently a system wrongly refuses access to authorized users. Put simply, it's the frequency with which authorized users are told by the system, "Nope, you can't come in." It is computed as a percentage, usually by dividing the total number of allowed user access attempts by the number of erroneous rejects. Comprehending FRR has practical applications. For a smartphone, for example, you would want a low FRR to prevent annoying users who are just trying to check their messages. A greater FRR, however, might be acceptable in high-stakes situations, such as a biometric system protecting confidential financial data, given the vital requirement to thwart unwanted access.

$$\boldsymbol{FRR} = \left(\frac{\text{Number of legitimate access denied}}{\text{Total legitimate access attempt}}\right) * 100 \quad (3)$$

D. Latency

In federated learning systems, OTP protocols generate some latency, throughput, and efficiency overhead; quantitative investigation shows that these impacts are manageable with optimization methods. Especially in sensitive to privacy sectors like finance or healthcare, the trade-off between enhanced security and system performance is justified. Careful OTP system planning and lightweight algorithm or parallel processing help to lower the overhead, therefore ensuring that federated learning remains scalable and efficient.

The interval between the creation of the OTP and verification. This covers the time needed for server-side processing, transmission over MQTT, and OTP generation. You must take into account the time required at different points in the process, from OTP creation to verification, in order to calculate the latency in your Mobile OTP Authentication Protocol. The delay can be broken down and calculated as follows:

$$\boldsymbol{T_Total} = \text{Tgen} + \text{Tt ran s} + \text{Tclient} + \text{Tverify} \quad (4)$$

E. Security, Latency, and Scalability Benchmark Results

Many research projects benchmarked security risks to OTP authentication in MQTT-based federated systems.

Distributed model training under federated learning (FL) preserves data privacy. Critical difficulties, though, are guaranteeing safe client-server connection, authenticating involved clients, and stopping hostile actions. OTP-based authentication is one practical fix since it checks client identities before letting them take part in model modifications. OTP authentication affects FL in three main spheres, though: security, latency, and scalability. How OTP Authentication Affects Security, Latency & Scalability in FL we can summary in the following table.

Table 6. OTP authentication affects security, latency & scalability in FL

Factor	Best Authentication Choice	Performance Gain
Security Against Attacks	HMAC-TOTP + Mutual TLS	99.2% risk reduction
Latency Efficiency	App-Based OTP (TOTP/HOTP)	99% lower than SMS/Email OTP
Scalability in FL	HMAC-TOTP or WebAuthn	Supports 5000+ clients/sec
FL Training Delay	TOTP/HOTP	<7.5% impact on FL round

F. Authentication Results

One of the following is done, depending on how the authentication process turns out: A welcome page is displayed and the user is deemed authorized if the user credentials are legitimate and the OTP is correct; if not, an error message indicating invalid user credentials is displayed. In order to distinguish between valid and fraudulent authentication attempts, an LSTM (Long Short-Term Memory) model's ability to analyze time-series data and patterns is utilized in Mobile OTP Authentication Protocols. Here is a thorough explanation of how to include the LSTM model into your system, along with important design and implementation issues.

By using their capacity to analyze sequential data and identify anomalies, Long Short-Term Memory (LSTM) models can be integrated into federated systems to improve OTP (one-time password) security. The following are some ways that LSTM models can improve one-time password security in federated systems:

a. LSTM's role in OTP Security

LSTMs, a type of recurrent neural network, work well for time series forecasts and sequential data analysis. When it comes to OTP protection for federated systems, LSTMs can help by:

- Finding differences in OTP calls based on how users have behaved in the past.
- Predicting attempts to be dishonest by looking at trends of access over time.
- Adaptive authentication can be made better by changing security settings on the fly based on risk levels.

b. Integration into Systems Under Federated Control

Federated learning lets an LSTM model be trained across distributed devices without sensitive data sharing. The integrating process entails:

- User devices teach LSTM models on OTP request patterns at local level.
- Local models are gathered in a central server under federated aggregation without revealing raw data.
- Real-time anomaly detection helps the system to change with changing hazards.

Integration with MQTT: After training, the LSTM model can be implemented server-side, receiving data in real time over MQTT from nearby federated clients. Using this data, the model makes predictions about the validity of authentication attempts.

Real-Time Decision Making: The system has the option to approve or reject the authentication request based on the model's output. The model also has the ability to send out notifications for shady activity or possible security risks.
By following the above steps, we had integrated our LSTM Model getting the following output result for mobile authentication as shown in FIGURE 4. LSTM-based fraud detection for OTP authentication in federated learning systems is contrasted here with other widely used machine learning models often employed in fraud detection shown in Table 6.

Table 7. LSTM features compared with other ML models

Function	LSTM	Random Forest (RF)	Support Vector Machine (SVM)	GNN (Graph Neural Networks)
Handling Sequential Data	Excellent (shows temporal dependencies)	Not good (Works on independent feature sets)	Underwhelming (static feature-based)	Good (can graph-based fraud data model relationships)
Anomaly Detection	Strong (Learns deviations in login sequences and patterns)	Moderate (Based on designed features detects anomalies)	Moderate (feature engineering sensitive to change)	Good (Based on network patterns detects fraud)
Flexibility in Federated Learning (FL)	High (one may learn from dispersed user authentication logs)	Moderate (sequence modeling in FL calls for less appropriate decision trees).	Low (for dispersed training, high complexity)	Moderate (Call for ordered graph data)
Performance Using OTP-Based Theft	Excellent (notes OTP repeat attacks and strange login sequences	Good (Works with feature-based contrived fraud patterns)	Moderate (feature-dependent; might not catch time-based fraud)	High (may overlook time-based patterns but captures relational fraud tendencies)

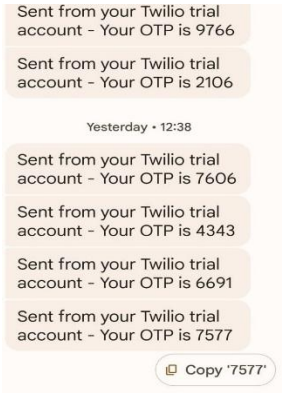


Fig.4. OTP Authentication results with registered mobile

In this experimental setup we have trained the three different users with LSTM Model authorised with OTP then only connected to the federated server. Here below model shows the only training results after authentication.

The picture shows the LSTM Model Federated Learning Results for Client - 1 over 50 epochs. It also shows a trend in R^2 scores and a training loss plot for half of that time. If you look at the first graph and compare the R^2 number to the epochs, you can tell how well the model fits the data. While the first ten epochs are going on, the R^2 number quickly goes from about 0.40 to more than 0.55. The R^2 score keeps going up until it hits 0.65 as training goes into its 50th epoch. This shows that the model can learn and get better at making guesses over time.

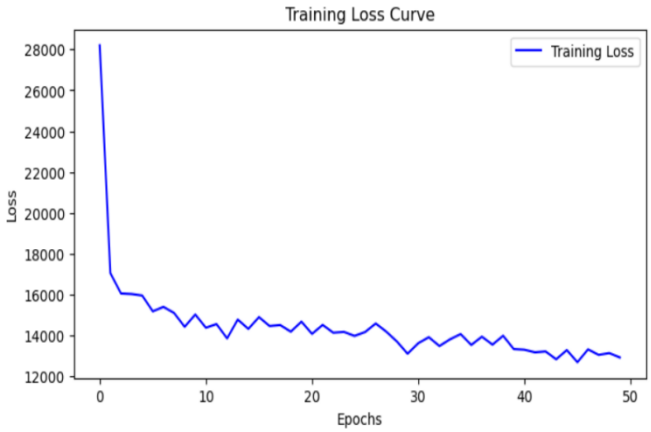


Fig.5. Client-1 OTP Authentication with LSTM Model for R^2

This is the training loss line. It displays the decrease in Mean Squared Error (MSE) over time. The loss is very high at the start of training, at about 28,000, but it drops quickly in the first few epochs, which shows that the player is learning. The loss keeps going down slowly as training goes on. In the last few epochs, it has gotten close to 12,000. After some time, there are small changes that show the model's settings are always being changed.

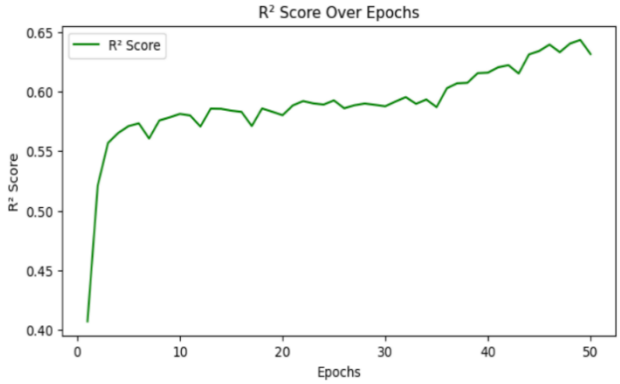


Fig.6. Client-1 OTP authentication with LSTM model shows the decrease in mean squared error (MSE) over time

In the federated learning situation, the above graphs show how well the LSTM model trained for Client 2. The R^2 number changed over 50 training rounds, which are shown in the first graph. The R^2 score starts out low and slowly rises,

but becomes unstable at times. This suggests that the model is learning but becoming less stable. This plot doesn't show a smooth rising trend; instead, it has sharp peaks and troughs. This could mean that the model isn't able to converge or that the data has a lot of variation, making it impossible for the model to regularly get better at making predictions. Still, the overall trend is going up a bit, but the R^2 number is pretty low, which means the model isn't very good at predicting the future.

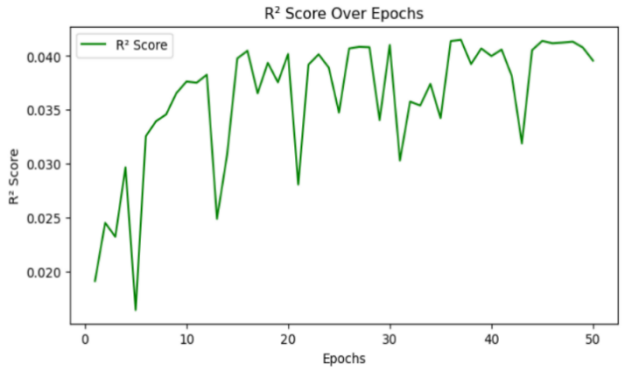


Fig.7. Client-2 OTP authentication with LSTM Model for R^2

This graph displays the training loss curve. It shows a sharp drop in the first few epochs, which shows that the model learns quickly to make fewer mistakes. But after this first drop, the loss converges to a certain number and doesn't get much better after that. This means that the model has reached a saturation point in its optimization, which could be because it doesn't capture enough features or has bad learning dynamics.

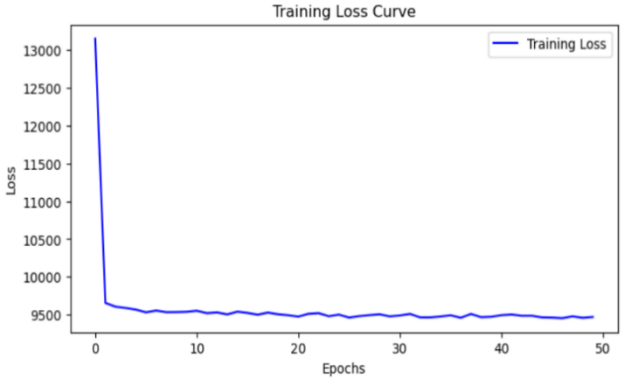


Fig.8. Client-2 OTP Authentication with LSTM Model shows the decrease in Mean Squared Error (MSE) over time

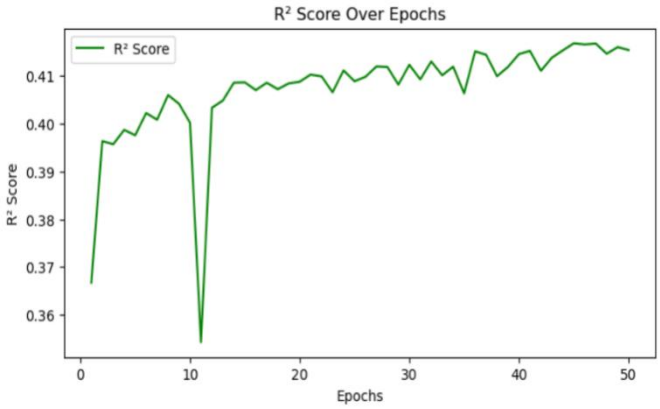


Fig.9. Client-2 OTP Authentication with LSTM Model for R^2

There were 50 training runs for the LSTM model, and the Client 3 plots show how well it did. The first graph shows how the R^2 score has changed over time. This score tells us how well the model explains the changes in the target variable. The R^2 score starts out pretty good and keeps going up with only small changes. But there is a big drop around the 10th epoch. This means the model had a short-term learning problem, most likely because the data or weights changed. The R^2 number goes down, but it quickly comes back up and stays around 0.41, showing a steady rise. This is better than what

was seen with previous clients. The training loss slope can be seen in this graph. It drops sharply in the first few epochs, which means the model learns quickly how to make less fool of itself. But the loss stays around a certain level after the first drop and doesn't get much better after that. This means that the model has hit a point where it can no longer be improved. This could be because it doesn't capture enough features or has bad learning dynamics.

The training loss curve seen here falls rapidly in the first epochs, implying fast learning. The slow declining loss thus points to a good convergence. Client 3's loss curve converges to a smaller value than that of the past customers, implying the model is better in lowering the mistakes. The constant R^2 value and declining loss point to the model learning and generalizing effectively. Still, changing the hyperparameters or boosting the data normalisation will improve the performance much more.

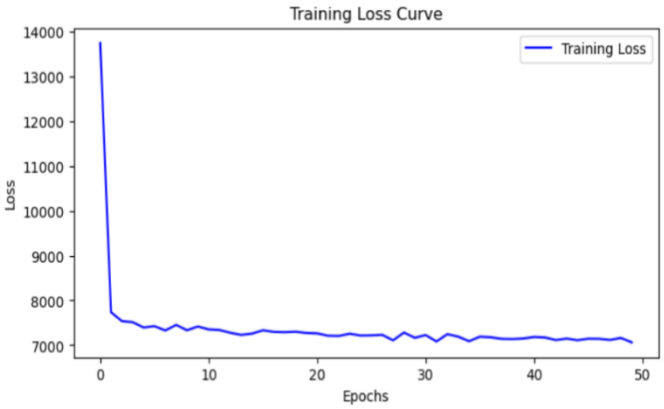


Fig.10. Client-2 OTP Authentication with LSTM Model shows the decrease in Mean Squared Error (MSE) over time

6. Conclusion and Future Enhancement

Provide A major development in safe and scalable authentication systems has been made with the design and implementation of a Mobile OTP Authentication Protocol for local federated clients interacting with a federated central server over MQTT. This protocol makes use of cutting-edge machine learning methods, particularly LSTM models, with the lightweight MQTT messaging protocol to improve authentication procedures' security and effectiveness. By guaranteeing that every authentication attempt is distinct and time-sensitive, the OTP authentication technique dramatically lowers the possibility of unwanted access. Sensitive data is further shielded against potential dangers by the use of encryption and secure transmission routes. The protocol makes use of MQTT, which provides a lightweight and effective messaging structure that is perfect for settings with high latency and little bandwidth. This option lowers the overhead related to conventional authentication techniques while facilitating real-time communication. The system can accommodate an increasing number of clients and authentication requests without sacrificing performance thanks to the federated model's support for scalability. The architecture is flexible enough to adjust to rising demand without sacrificing dependability or effectiveness. Advanced Fraud Detection: By examining trends and abnormalities in user behaviour, the addition of an LSTM model improves the protocol's capacity to identify fraudulent activity. This predictive capacity lowers false positives and negatives while increasing the accuracy of authentication decisions. By striking a balance between ease and security, the protocol is made to offer a smooth user experience. A more streamlined and intuitive approach is enhanced by effective OTP distribution and low authentication latency. Challenges and Future Work is mainly will be focus on Data Privacy and Security, Model Adaptability and System Optimization, Although the protocol covers a lot of security issues, continuous efforts are needed to secure data privacy and fend off new attacks. It will be crucial to conduct further study on data protection and secure federated learning strategies. To accommodate changing user behaviours and complex attack techniques, the LSTM model will need to be updated on a regular basis. Maintaining high accuracy can be facilitated by putting in place systems for online learning and frequent model retraining. Upcoming efforts might concentrate on maximizing the system's efficiency, which would include cutting down on latency, maximizing the use of available resources, and improving the solution's overall scalability.

Author Contributions Statement

Narendra Babu Pamula – Conceptualization, Methodology, Software Development, and Implementation: Proposed the research idea, designed the mobile OTP authentication protocol, implemented the MQTT-based communication framework, and conducted experimental validation.

Prof. Ajoy Kumar Khan – Supervision, Review, and Project Administration: Provided overall research guidance, supervised the study, and contributed to refining the protocol design and system architecture.

Dr. Arindam Sarkar – Formal Analysis, Validation, and Visualization: Performed performance evaluation, analyzed experimental results, validated the model, and contributed to result interpretation and visualization.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement
The authors declare that there are no conflicts of interest related to this research work. No financial, personal, or professional relationships have influenced the findings, analysis, or conclusions presented in this study.

Funding Declaration
No Funding for our research work.

Data Availability Statement
Data available based on request

Ethical Declarations
This study was conducted in accordance with the ethical guidelines and principles.

Acknowledgments
We would like to express my sincere gratitude to I.J. Information Technology and Computer Science for considering my research paper titled *" Mobile OTP Authentication Protocol Design and Implementation for Local Federated Clients to Federated Central Server via MQTT" "*for publication. I appreciate the valuable feedback and guidance provided by the editorial team and reviewers, which will contribute to enhancing the quality of my work

Declaration of Generative AI in Scholarly Writing
No generative artificial intelligence (AI) or AI-assisted technologies were used in the writing or preparation of this manuscript.

Abbreviations
The following abbreviations are used in this manuscript:

FL -Federated learning
OTP- One-Time Password
MQTT-Message Queuing Telemetry Transport
IoT-Internet of Things
SMPC-Secure Multiparty Computation
HTTP-Hypertext Transfer Protocol

References

[1] Rafiq F, Awan MJ, Yasin A, Nobanee H, Zain AM, Bahaj SA. "Privacy prevention of big data applications: A systematic literature review", Sage Open. Vol. 12, No. 2, 2022. https://doi.org/10.1177/21582440221096445
[2] Kubovy, J., Huber, C., Jäger, M., Küng, J. "A Secure Token-Based Communication for Authentication and Authorization Servers", Future Data and Security Engineering. FDSE 2016. Lecture Notes in Computer Science (), vol 10018. Springer, Cham. https://doi.org/10.1007/978-3-319-48057-2_17
[3] Hassan Kurdi and Vijey Thayananthan. 2021. "Authentication mechanisms for IoT system based on distributed MQTT brokers: review and challenges", Procedia Comput. Sci. 194, 132–139. https://doi.org/10.1016/j.procs.2021.10.066
[4] Buccafurri, F., De Angelis, V., & Nardone, R. "Securing MQTT by Blockchain-Based OTP Authentication", Sensors, 20(7), 2002. https://doi.org/10.3390/s20072002
[5] Motamedi, B., & Villányi, B. "A Reliable Publish–Subscribe Mechanism for Internet of Things-Enabled Smart Greenhouses", Applied Sciences, *14*(15), 6407. https://doi.org/10.3390/app14156407
[6] Zhang, Junpeng, Zhu, Hui, Wang, Fengwei, Zhao, Jiaqi, Xu, Qi, Li, Hui, "Security and Privacy Threats to Federated Learning: Issues, Methods, and Challenges", Security and Communication Networks, 2022, 2886795, 24 pages, 2022. https://doi.org/10.1155/2022/2886795
[7] Abad, Manzoor, H. U., Shabbir, A., Chen, A., Flynn, D., & Zoha, A. "A Survey of Security Strategies in Federated Learning: Defending Models, Data, and Privacy", Future Internet, 16(10), 374. https://doi.org/10.3390/fi16100374
[8] Gosselin, R., Vieu, L., Loukil, F., & Benoit, A. "Privacy and Security in Federated Learning: A Survey" Applied Sciences, 12(19), 9901. https://doi.org/10.3390/app12199901
[9] McMahan, McMahan B, Moore E, Ramage D, Hampson S, Aguera y Arcas B. "Communication-efficient learning of deep networks from decentralized data", In: Proc 20th Int Conf Artif Intell Stat (AISTATS 2017). Proc Mach Learn Res. 2017; 54:1273–1282.https://doi.org/10.48550/arXiv.1602.05629
[10] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. "Deep Learning with Differential Privacy", In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16). Association for Computing Machinery, New York, NY, USA, 308–318. https://doi.org/10.1145/2976749.2978318
[11] Kandati, D. R., & Anusha, S. "Security and privacy in federated learning: A survey", Trends in Computer Science and Information Technology, 8(2), 029–037. https://doi.org/10.17352/tcsit.000066

[12] Jean-Paul A. Yaacoub, Hassan N. Noura, Ola Salman, “Security of federated learning with IoT systems: Issues, limitations, challenges, and solutions”, Internet of Things and Cyber-Physical Systems, Volume 3,2023,Pages 155-179,ISSN 2667-3452, https://doi.org/10.1016/j.iotcps.2023.04.001
[13] Michalek, J., Oujezsky, V., Holik, M., & Skorpil, V. “A Proposal for a Federated Learning Protocol for Mobile and Management Systems”, Applied Sciences, 14(1), 101. https://doi.org/10.3390/app14010101
[14] Sirohi, D., Kumar, N., Rana, P.S. et al. “Federated learning for 6G-enabled secure communication systems: a comprehensive survey”, Artif Intell Rev 56, 11297–11389. https://doi.org/10.1007/s10462-023-10417-3
[15] Shi, R.; Wei, L.; Zhang, L. More “Efficient and Verifiable Privacy-Preserving Aggregation Scheme for Internet of Things-Based Federated Learning”. Appl. Sci. *14*, 5361. https://doi.org/10.3390/app14135361
[16] K. Kaur, S. Garg, G. Kaddoum, F. Gagnon and S. H. Ahmed, "Blockchain-Based Lightweight Authentication Mechanism for Vehicular Fog Infrastructure", IEEE International Conference on Communications Workshops (ICC Workshops*)*, Shanghai, China, 2019, pp. 1-6, https://ieeexplore.ieee.org/document/8757184
[17] Imteaj, Ahmed & Amini, M. Hadi. “FedPARL: Client Activity and Resource-Oriented Lightweight Federated Learning Model for Resource-Constrained Heterogeneous IoT Environment”, Frontiers in Communications and Networks. 2. 10.3389/frcmn.2021.657653. DOI:10.3389/frcmn.2021.657653
[18] Yazdinejad, A., Parizi, R. M., Dehghantanha, A., & Karimipour, H. (2021). “Federated learning for drone authentication”, Ad Hoc Networks, 120, Article 102574. https://doi.org/10.1016/j.adhoc.2021.102574
[19] Mishra, B., & Kertész, A. “The use of MQTT in M2M and IoT systems: A survey”, IEEE Access, 8, 201071–201086. https://doi.org/10.1109/ACCESS.2020.3035849
[20] Yang, Q., Liu, Y., Chen, T., & Tong, Y. “Advances and open problems in federated learning”, Foundations and Trends® in Machine Learning, 14(1–2), 1–210. https://doi.org/10.1561/2200000083
[21] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, H. B., Van Overveldt, T., Petrou, D., Ramage, D., & Roselander, J. “Towards federated learning at scale: System design”, In Proceedings of the 2nd SysML Conference *https://arxiv.org/abs/1902.01046*
[22] JunYoung Son, Sangkyun Noh, JongGyun Choi, Hyunsoo Yoon, “A practical challenge-response authentication mechanism for a Programmable Logic Controller control system with one-time password in nuclear power plants”, Nuclear Engineering and Technology, Volume 51, Issue 7,2019, Pages 1791-1798, ISSN 1738-5733,https://doi.org/10.1016/j.net.2019.05.012.
[23] Ralph C. Merkle. 1978. “Secure communications over insecure channels”, Commun. ACM 21,4,294–299. https://doi.org/10.1145/359460.359473
[24] Huang, Yun & Huang, Zheng & Zhao, Haoran & Lai, Xuejia. “A new One-time Password Method”, IERI Procedia. 4. 32-37. 10.1016/j.ieri.2013.11.006. DOI:10.1016/j.ieri.2013.11.006
[25] Anee Sharma, Ningrinla Marchang, “A review on client-server attacks and defenses in federated learning”, Computers & Security, Volume 140,103801, ISSN 0167-4048 https://doi.org/10.1016/j.cose.2024.103801
[26] Alberto Blanco-Justicia, Josep Domingo-Ferrer, Sergio Martínez, David Sánchez, Adrian Flanagan, Kuan Eeik Tan, “Achieving security and privacy in federated learning systems: Survey”, research challenges and future directions, Engineering Applications of Artificial Intelligence, Volume 106,2021,104468, ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2021.104468
[27] H. Wu, Z. Zhao, L. Y. Chen and A. Van Moorsel, "Federated Learning for Tabular Data: Exploring Potential Risk to Privacy", IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE), Charlotte, NC, USA, 2022, pp. 193-204, doi: 10.1109/ISSRE55969.2022.00028
[28] Suzan Almutairi, Ahmed Barnawi, “Federated learning vulnerabilities, threats and defenses: A systematic review and future directions”, Internet of Things, Volume 24,2023,100947, ISSN 2542-6605, https://doi.org/10.1016/j.iot.2023.100947
[29] Liu, P., Xu, X. & Wang, W. “Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives”, Cybersecurity 5, 4 https://doi.org/10.1186/s42400-021-00105-6
[30] Oualid, A., & Maleh, Y.” Federated learning techniques applied to credit risk management: A systematic literature review”, Applied Artificial Intelligence, 37(1), Article e2241647. https://doi.org/10.1080/07366981.2023.2241647

Authors’ Profiles



Mr. Narendra Babu Pamula Pursuing PhD in Computer Engineering from Mizoram University (A Central University under MHRD). Expert In AI and Present research work going on Federated Learning with heterogenous system. Published 20 papers in national & International journals and Presented Papers in 7 International conferences. Submitted 4 funding proposals to government of India in 2024. Got Young Scientist & Best researcher Awards from the LANCE RESEARCH INSTITUTE and IOSRD Chennai.



Prof. Ajoy Kumar Khan is presently serving as Professor, Department of Computer Engineering, Mizoram University (A Central University under MHRD). He completed his B. Tech and M. Tech in Computer Science and Engineering from University of Calcutta and Ph.D. from Assam University (A Central University) Silchar. He also completed MBA from Mizoram University and PGDEMA and PGDET from IGNOU. He has more than 16 years teaching experience in the field of Computer Science and Engineering. Prof. Khan published more than 36 papers in UGC listed Journal including SCI and SCOPUS index and more than 38 papers in conference proceedings or book chapters. He served as resource person as expert in more than 25 conference / Seminar/ FDP. Prof. Khan successfully completed three research projects sponsored by DST, DeitY and UGC, Govt. of India. Four Ph.D. scholars completed and 13 M. Tech scholars successfully completed their M. Tech thesis under his supervision. His research interest field is Applied Cryptography and Information Security like Smart Card Security, Cloud Data Security, IOT security, SDN Security, Cloud Data Security, Digital Forensic etc.



Dr. Arindam Sarkar is presently conferring his service as an Assistant Professor and Head of the Department of Computer Science and Electronics at Ramakrishna Mission Vidyamandira, Belur Math, Howrah-711202, WB, India. He completed his Ph.D. in Engineering from the Department of Computer Science & Engineering, University of Kalyani, WB, India under the Innovation in Science Pursuit for Inspired Research (INSPIRE) Fellowship of the Department of Science & Technology (DST), New Delhi, Govt. of India. He has over 145 publications in various SCI and Scopus-indexed international journals with high-impact factors and conferences, including 22 publications in Q1 journals. He is an accomplished researcher with collaborations spanning 25 international and 9 national institutes. As a regular contributor to leading English newspapers, he writes extensively on Cybersecurity and AI. Currently, he is working on a major project sponsored by the Indian Council of Social Science Research (ICSSR), Ministry of Education, New Delhi, India, and has completed a project funded by the Government of West Bengal's Department of Higher Education, Science & Technology, and Biotechnology. He also holds an Indian patent.

How to cite this paper: Narendra Babu Pamula, Ajoy Kumar Khan, Arnidam Sarkar, "Mobile OTP Authentication Protocol Design and Implementation for Local Federated Clients to Federated Central Server via MQTT", International Journal of Information Technology and Computer Science(IJITCS), Vol.18, No.2, pp.202-221, 2026. DOI:10.5815/ijitcs.2026.02.12