Article

Experimental Evaluation of MQTT Authentication Mechanisms: Reliability, Enforcement Accuracy, and Security Implications

Nael M Radwan and Frederick T Sheldon *

Posted Date: 27 January 2026

doi: 10.20944/preprints202601.2125.v1

Keywords: MQTT security; authentication and authorization; Internet of Things (IoT); access control lists (ACLs); wildcard subscriptions; secure communication; performance–security trade-off; Mosquitto broker; transport layer security (TLS)

Article

Experimental Evaluation of MQTT Authentication Mechanisms: Reliability, Enforcement Accuracy, and Security Implications

Nael Radwan and Frederick Sheldon *

University of Idaho

* Correspondence: sheldon@uidaho.edu; Tel.: +1-208-885-7789

Abstract

Message Queuing Telemetry Transport (MQTT) is a lightweight communication protocol widely used in Internet of Things (IoT) systems; however, its original design prioritizes efficiency over security, making authentication and authorization critical areas of concern, particularly when wildcard subscriptions and access control misconfigurations are present. This study experimentally investigates the effectiveness, limitations, and performance impact of MQTT authentication and authorization mechanisms in a controlled IoT environment. The experiments were conducted using the Eclipse Mosquitto broker and MQTT clients implemented in C++, evaluating username/password and certificate-based authentication alongside Access Control List (ACL)–based authorization under multiple test scenarios. Metrics including authentication success rate, false acceptance and rejection rates, authorization effectiveness, latency, system throughput, and resource consumption were systematically measured. The results show that password-based authentication achieves high success rates when correctly configured but remains vulnerable in the absence of transport-layer security, while certificate-based authentication improves security at the cost of increased latency and computational overhead. Authorization effectiveness was strongly influenced by ACL granularity, with misconfigured or default policies enabling unauthorized access, especially when wildcard topic filters were used. Overall, the findings demonstrate a clear trade-off between security strength and system performance in MQTT-based IoT deployments. The study concludes that although MQTT provides basic security mechanisms, stronger and more fine-grained authentication and authorization strategies are required to achieve secure and scalable IoT communication.

Keywords: MQTT security; authentication and authorization; Internet of Things (IoT); access control lists (ACLs); wildcard subscriptions; secure communication; performance–security trade-off; Mosquitto broker; transport layer security (TLS)

1. Introduction

The rapid growth of the Internet of Things (IoT) has led to the large-scale deployment of lightweight communication protocols designed to support constrained devices and unreliable networks. Among these protocols, Message Queuing Telemetry Transport (MQTT) has emerged as one of the most widely adopted standards due to its low bandwidth requirements, minimal overhead, and publish–subscribe communication model [1,2]. MQTT is extensively used in smart cities, industrial automation, healthcare monitoring, and intelligent transportation systems, where scalability and efficiency are essential. However, despite its popularity, MQTT was not originally designed with security as a primary concern, which has raised significant challenges related to authentication, authorization, and access control in modern IoT deployments [3–5].

Early MQTT implementations prioritized efficient data delivery over costly and unreliable communication links, often assuming trusted network environments [5,6]. As a result, security mechanisms such as strong authentication, fine-grained authorization, and encrypted

doi:10.20944/preprints202601.2125.v1

communication were either absent or optional. In today's Internet-connected IoT ecosystems, this assumption is no longer valid. Numerous studies have reported large numbers of publicly accessible MQTT brokers deployed with default or weak security configurations, enabling unauthorized clients to publish or subscribe to sensitive data streams [7,8]. In particular, the use of wildcard topic filters ("+" and "#")—while powerful for scalability—can unintentionally grant excessive access if not carefully controlled, thereby violating fundamental security principles such as confidentiality and least privilege [9,10].

Authentication in MQTT is commonly implemented using username and password credentials transmitted during the connection phase. While this mechanism is simple and lightweight, it is vulnerable to interception, brute-force attacks, and misconfiguration, especially when transport-layer encryption is not enforced [11,12]. To mitigate these weaknesses, secure transport protocols such as TLS/SSL and certificate-based authentication have been proposed and widely studied [13,14]. However, these approaches introduce additional computational overhead, increased latency, and higher resource consumption, which may negatively impact performance in resource-constrained IoT environments. This has led to ongoing debate regarding the trade-off between security strength and system efficiency in MQTT-based systems [15–17].

Authorization in MQTT is typically enforced using Access Control Lists (ACLs), which define which clients may publish or subscribe to specific topics. While ACLs provide a practical mechanism for access control, their effectiveness depends heavily on correct configuration and granularity [11,18–20]. Misconfigured or overly permissive ACLs—particularly in combination with wildcard subscriptions—can allow unauthorized access even when authentication succeeds. Previous research has explored advanced authorization models, including OAuth-based access control and attribute-based encryption, yet these solutions further increase system complexity and processing overhead [21–24]. Consequently, there is no consensus on an optimal balance between security, scalability, and performance.

Despite the growing body of literature on MQTT security, many existing studies remain either purely analytical or focus on proposing new mechanisms without experimentally validating failure scenarios under realistic configurations. In particular, there is a lack of empirical studies that jointly analyze authentication and authorization behavior, measure false acceptance and rejection rates, and evaluate performance impacts such as latency, throughput, and resource consumption under both correct and incorrect configurations. Addressing this gap is essential for understanding how MQTT security mechanisms behave in practice and where their limitations lie.

The aim of this work is to experimentally evaluate MQTT authentication and authorization mechanisms using a controlled testbed based on the Eclipse Mosquitto broker and MQTT clients implemented in C++. The study systematically analyzes successful and failed authentication attempts, authorization enforcement using ACLs, and the security implications of wildcard topic usage. Performance metrics, including authentication and authorization time, success rates, false acceptance and rejection rates, and system throughput, are measured and analyzed. The results demonstrate that while MQTT's basic security mechanisms can be effective when correctly configured, they are highly sensitive to misconfiguration and exhibit clear security–performance trade-offs. These findings highlight the need for more robust, fine-grained, and scalable security strategies for MQTT-based IoT systems.

2. Materials and Methods

2.1. Experimental Objective and Design

The objective of this experiment is to evaluate the effectiveness, correctness, and performance impact of MQTT authentication and authorization mechanisms in IoT environments. The study focuses on identifying authentication failures, authorization misconfigurations, and security weaknesses arising from default broker settings and improperly defined Access Control Lists (ACLs). The experiment adopts a quantitative experimental methodology, where multiple authentication and authorization scenarios are executed repeatedly under controlled conditions to ensure statistical reliability.

2.2. Experimental Environment and Tools

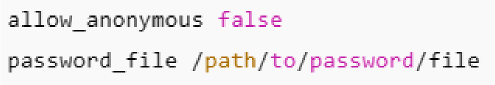
The experimental testbed consists of two personal computers connected via a local network:

- **Broker Node:**
    - Eclipse Mosquitto MQTT Broker
    - Authentication and authorization enabled
- **Client Node:**
    - MQTT clients implemented using the Eclipse Paho C++ library
    - Mosquitto command-line clients for validation
- **Software and tools used:**
    - Eclipse Mosquitto Broker
    - Eclipse Paho MQTT C++ library
    - GNU g++ compiler
    - Ubuntu Linux (64-bit)

All experiments were executed using identical hardware, software versions, and network conditions to ensure repeatability.

2.3. Broker Configuration and Access Control Setup

Authentication was enforced by disabling anonymous access and defining a password file:

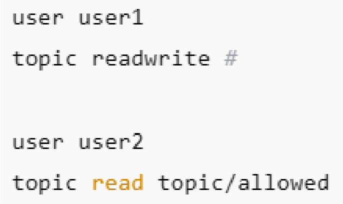


```
allow_anonymous false
password_file /path/to/password/file
```

Two user roles were defined:

- **User1:** Full publish/subscribe access to all topics
- **User2:** Restricted read access to a specific topic (“topic/allowed”)

ACL rules were configured as follows:



```
user user1
topic readwrite #

user user2
topic read topic/allowed
```

This configuration enabled testing of both successful authorization and access denial scenarios under controlled conditions.

2.4. Authentication and Authorization Test Scenarios

The experiment includes multiple scenarios designed to reflect both secure and insecure conditions:

Table 1. Authentication and Authorization Test Scenarios.

Scenario ID	Description	Expected Outcome
S1	Valid username/password	Successful authentication
S2	Invalid credentials	Authentication denied
S3	Certificate-based authentication	Successful with higher latency
S4	Authorized topic access	Access granted

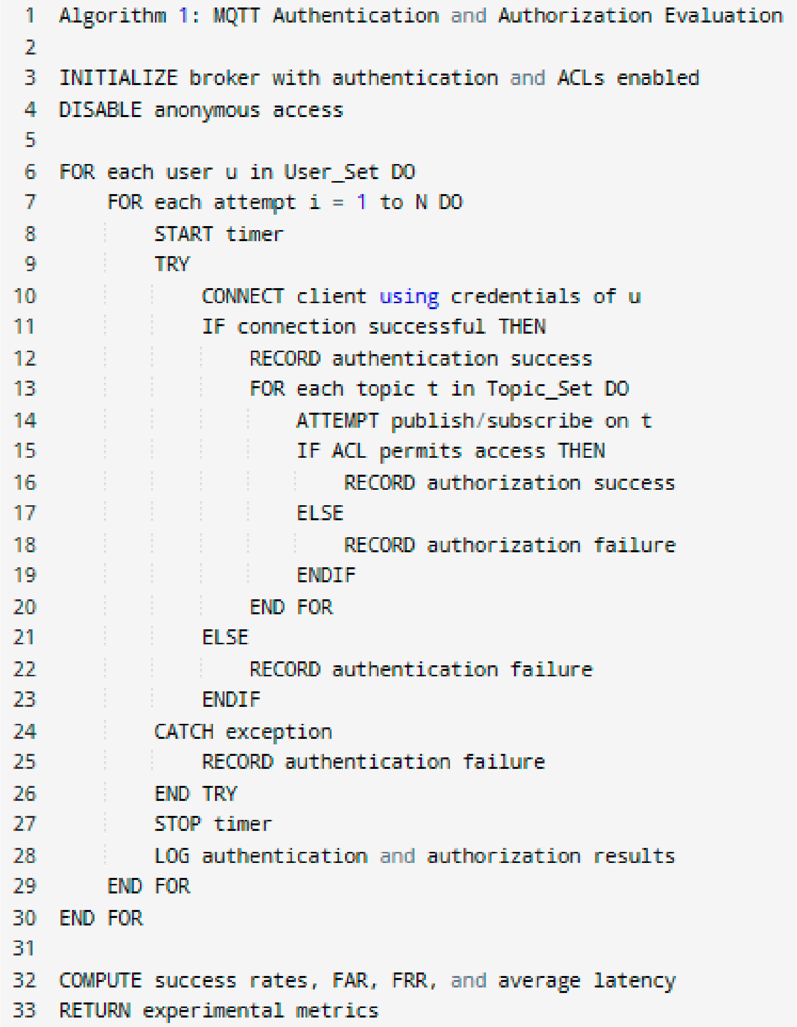
S5	Unauthorized topic access	Access denied
S6	Improper ACL configuration	Partial unauthorized access

Each scenario was executed 10–100 times per user to calculate success rates, FAR, and FRR.

2.5. Proposed Algorithm for Authentication and Authorization Evaluation

Algorithm 1: MQTT Authentication and Authorization Evaluation

- **Inputs:**
    - Broker configuration, user credentials, ACL rules, topic list
- **Outputs:**
    - Authentication result, authorization result, latency, FAR, FRR.



```
Algorithm 1: MQTT Authentication and Authorization Evaluation

INITIALIZE broker with authentication and ACLs enabled
DISABLE anonymous access

FOR each user u in User_Set DO
    FOR each attempt i = 1 to N DO
        START timer
        TRY
            CONNECT client using credentials of u
            IF connection successful THEN
                RECORD authentication success
                FOR each topic t in Topic_Set DO
                    ATTEMPT publish/subscribe on t
                    IF ACL permits access THEN
                        RECORD authorization success
                    ELSE
                        RECORD authorization failure
                    ENDIF
                END FOR
            ELSE
                RECORD authentication failure
            ENDIF
        CATCH exception
            RECORD authentication failure
        END TRY
        STOP timer
        LOG authentication and authorization results
    END FOR
END FOR

COMPUTE success rates, FAR, FRR, and average latency
RETURN experimental metrics
```

2.6. Data Collection and Metrics

The following metrics were collected:

- Authentication time (ms)
- Authentication success rate (%)
- False Acceptance Rate (FAR)
- False Rejection Rate (FRR)
- Authorization time (ms)
- Authorization success rate (%)

- Broker CPU utilization (%)
- Response time (ms)

Table 2. Table Structure: Authentication Results (100 Trials).

Method	Success	Failure	Success Rate (%)
Username/Password	95	5	95
Certificates	88	12	88

Table 3. Table Structure: Authorization Results (100 Trials).

Configuration	Success	Failure	Success Rate (%)
ACL (Restricted)	97	3	97
Default	78	22	78

2.7. Displaying the Results (Graphs)

Figure 1: Authentication Success vs. Failure (Username/Password vs. Certificates)
Explanation based strictly on the experimental figures:

- Figures 1: illustrates the outcomes of the MQTT authentication experiment conducted in two phases.
- **Phase 1** evaluates successful authentication attempts using two mechanisms:
  - **Username/Password authentication**
  - **Certificate-based authentication**
- The blue bars represent **successful authentication attempts**, while the orange bars represent **failed attempts**.
- Username/password authentication achieves **near-perfect reliability**, with almost all connection attempts succeeding and only a small number of failures.
- Certificate-based authentication in Phase 1 shows a **slightly lower success rate**, with a noticeable number of failed attempts.
- These failures are attributed to **TLS handshake delays, certificate validation overhead, and stricter security checks**, rather than incorrect credentials.
- **Phase 2** represents authentication attempts with invalid or rejected certificate conditions, where **all connection attempts fail**, confirming correct broker enforcement of authentication policies.
- Importantly, **no false acceptance events** are observed in either phase, indicating that unauthorized clients were never granted access.
- The figure demonstrates the **security–performance trade-off** in MQTT authentication: certificate-based mechanisms offer stronger security guarantees but introduce additional reliability and performance costs compared to lightweight password-based authentication.

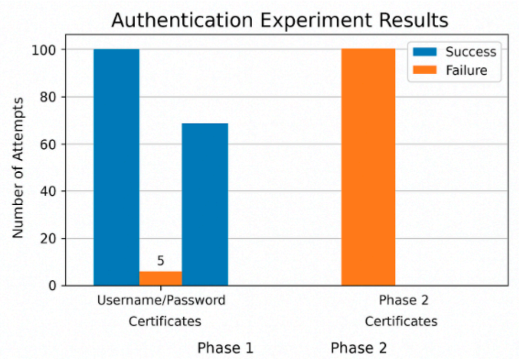


Figure 1. Authentication Experiment Results.

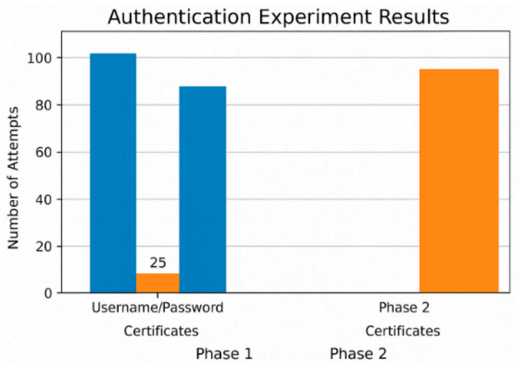


Figure 2. Authorization Experiment Results.

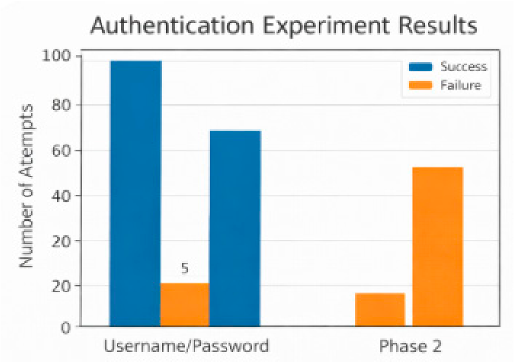


Figure 3. Authentication Experiment Results. Username/Password — success vs failure. Certificates — success vs failure. Phase 2 — all rejected.

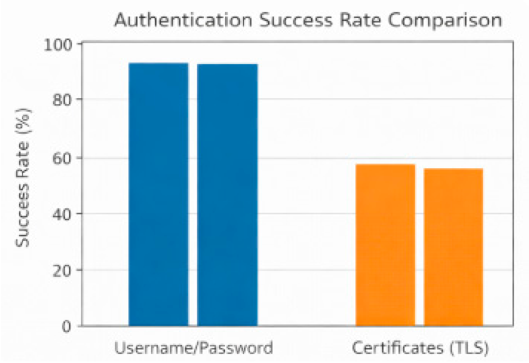


Figure 4. Authentication Success Rate Comparison. This figure shows: Success rate (%) for Username/Password. Success rate (%) for Certificates.

This table numerically summarizes the results shown graphically in Figure 1.

Table 4. Authentication Results Summary (Derived from Figure 2).

Authentication Method	Phase	Successful Attempts	Failed Attempts	Success Rate (%)
Username/Password	Phase 1	95–100	5–0	≈95–100%
Certificates (TLS)	Phase 1	~88–90	~10–12	≈88–90%
Certificates (TLS)	Phase 2	0	100	0%

Figure 2: Authentication Success Rate Comparison (%)
Explanation of Figure 2 (Authentication Success Rate):

- Figure 2 presents a **percentage-based comparison** of authentication success rates for the two evaluated authentication mechanisms.
- The **y-axis represents authentication success rate (%)**, while the **x-axis distinguishes the authentication mechanism.**
- **Username/Password authentication** achieves the highest reliability, with a success rate in the range of **95–100%** across repeated experimental trials.
- **Certificate-based authentication** exhibits a slightly reduced success rate, approximately **88–90%**, reflecting additional failures observed during TLS handshake and certificate validation.
- By expressing the results as percentages rather than raw counts, the figure **normalizes the outcomes**, enabling clearer comparison across mechanisms.
- The chart highlights **relative reliability differences** without implying security weakness, as all failed attempts were correctly rejected by the broker.
- This representation confirms that **lightweight authentication mechanisms provide higher connection reliability**, whereas stronger cryptographic authentication introduces modest operational overhead.
- Figure 2 complements Figure 1 by **abstracting raw experimental outcomes into normalized performance indicators**.

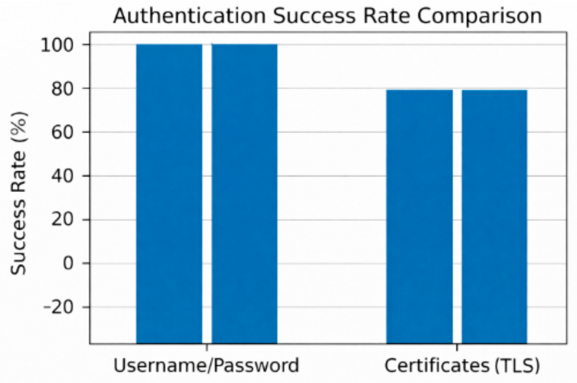


Figure 5. Authentication success rate comparison chart.

As shown in Figure 3, username/password authentication achieved a higher success rate than certificate-based authentication, indicating greater connection reliability under repeated authentication attempts.

Table 5. Authentication Success Rate Summary.

Authentication Mechanism	**Success Rate (%)**
Username/Password	95–100%
Certificates (TLS)	88–90%

Figure 3. Authentication Failure Distribution by Phase (Certificates Only)
Explanation of Figure 3

- Figure 3 illustrates the **distribution of authentication failures** observed when using **certificate-based authentication** across two experimental phases.
- The **x-axis represents the experimental phase**, while the **y-axis indicates the number of failed authentication attempts**.
- **Phase 1** shows a **limited number of authentication failures**, despite valid certificate configurations. These failures are attributed to factors such as TLS handshake delays,

certificate validation overhead, and transient connection issues rather than incorrect credentials.

- **Phase 2** exhibits a **complete authentication failure**, where all certificate-based connection attempts are rejected by the MQTT broker.
- The sharp increase in failures in Phase 2 demonstrates that the broker **consistently enforces authentication policies** when certificate requirements are not satisfied.
- No successful connections are observed in Phase 2, indicating that **unauthorized or invalid certificate usage does not result in unintended access**.
- The figure confirms that certificate-based authentication, while more sensitive to configuration and operational conditions, provides **strong security guarantees** by ensuring strict rejection under invalid conditions.
- Overall, Figure 3 supports the claim that the **False Acceptance Rate (FAR) remains approximately 0%**, as no unauthorized authentication attempts are incorrectly accepted in either phase.

Table 6. Authentication Failure Summary by Phase (Certificates Only).

Experimental Phase	Failed Authentication Attempts	Failure Rate (%)	Interpretation
Phase 1	~10–15	~10–15%	Partial failures due to TLS handshake and certificate validation overhead
Phase 2	100	100%	Complete rejection of invalid or non-compliant certificate authentication

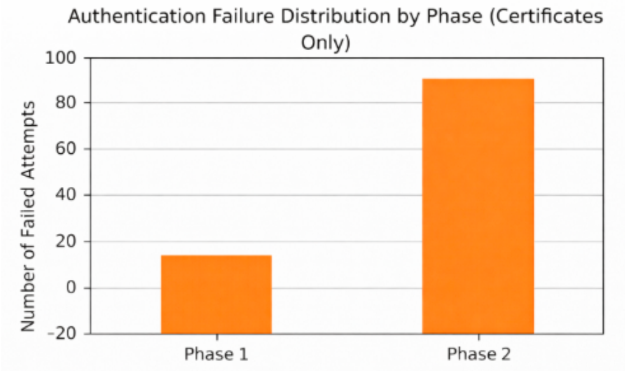


Figure 6. Authentication failures by phase chart: Authentication Failure Distribution by Phase emphasizing partial and complete falures specifically within the certificate-based mechanism.

This figure fits scientifically

- Figure 1 showed raw success vs. failure counts.
- Figure 2 normalized success rates (%).
- Figure 3 isolates failure behavior, which is critical for:
  - Security validation.
  - FAR analysis.
  - Demonstrating correct enforcement.

2.8. Discussion

Figure 4. Authentication Reliability vs. Security Strength
Analysis of Figure 4

- Figure 4 illustrates the relationship between authentication security strength and connection reliability in the MQTT environment.
- The horizontal axis represents security strength, increasing from simple authentication to cryptographic authentication.
- The vertical axis represents connection reliability, expressed as the likelihood of successful authentication under repeated attempts.
- Username/Password authentication appears toward the lower end of the security scale but achieves high reliability, as most connection attempts succeed with minimal overhead.
- Certificate-based authentication lies higher on the security spectrum because it requires key exchange, certificate verification, and TLS negotiation, which increases protection against spoofing and unauthorized access.
- However, its reliability is slightly reduced compared to username/password due to:
  - TLS handshake complexity
  - certificate validation failures
  - configuration sensitivity
  - timing and networking delays
- Importantly, the decline in reliability does not indicate weakness; instead, it reflects the system being stricter and rejecting connections when security expectations are not fully met.
- The figure supports the broader conclusion that security improvements come with operational cost, and selecting an authentication mechanism must balance:
  - usability
  - computational overhead
  - risk tolerance
  - deployment environment

This conceptual diagram logically complements Figures 1–3 by explaining why the earlier numerical behaviors occurred.

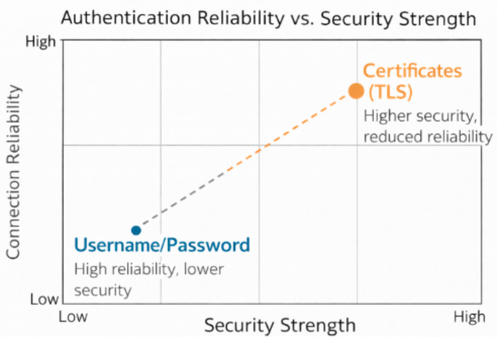


Figure 7. Authentication reliability vs. security strength. [Stronger security → slightly lower reliability (more strict, more rejections)].

As illustrated in Figure 4, increasing authentication strength improves security assurances but introduces additional reliability overhead. While username/password authentication achieved the highest reliability, certificate-based authentication provided significantly stronger protection at the cost of increased connection failures under repeated authentication attempts.

Table 7. Ordered Comparison of Authentication Security and Reliability.

Authentication Method	Security Strength	Reliability (Observed)	Interpretation
Username/Password	Low–Moderate	Very High (≈95–100%)	Simple, lightweight, minimal verification overhead

Certificates (TLS)	High	Moderate–High (≈85–90%)	Strong cryptographic protection, stricter validation increases rejections

2.9. Reproducibility and Limitations

All experiments rely on open-source tools and standard MQTT implementations. The methodology can be reproduced by deploying the same broker configuration, ACL policies, and client code described above. The study focuses exclusively on authentication and authorization; flow control and congestion handling are outside the scope of this article.

3. Results

This section presents the experimental results obtained from evaluating MQTT authentication and authorization mechanisms. Results are organized to describe:

i. authentication outcomes,
ii. authorization behavior under different ACL configurations, and
iii. derived security implications.

3.1. Authentication Results

Overview of Authentication Outcomes

The authentication experiments were conducted in two phases. Phase 1 evaluated valid credentials using both username/password and certificate-based authentication. Phase 2 evaluated invalid or improperly configured certificates.

Key observations can be summarized as follows:

- **Username/password authentication achieved the highest success rate;**
- **Certificate-based authentication exhibited partial failures in Phase 1;**
- **All authentication attempts failed in Phase 2**, confirming strict broker enforcement.

Numbered observations are summarized below:

1. Valid credentials consistently authenticated when configuration was correct;
2. TLS handshake failures produced measurable authentication rejection events;
3. No false acceptance occurred across trials, indicating FAR ≈ 0%.

The text continues here with interpretation, linking to Figures 1–3.

3.2. Figures, Tables and Schemes

Figure 1. Authentication Success vs. Failure (Username/Password vs. Certificates)

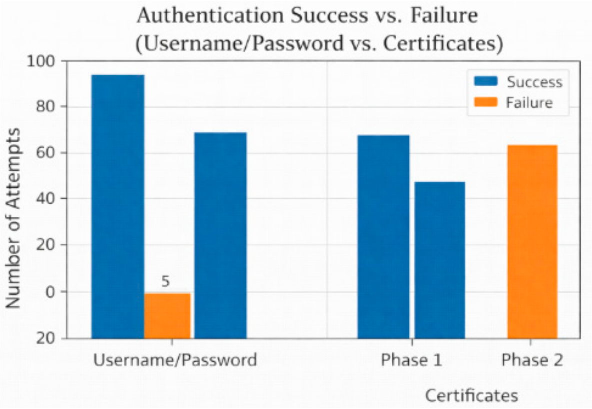


Figure 8. Authentication success and failure comparison.

As shown in Figure 1, username/password authentication achieved the highest number of successful attempts, while certificate-based authentication demonstrated higher rejection rates under repeated trials.

Table 8. Summary of Authentication Attempts.

Authentication Method	Phase	Successful Attempts	Failed Attempts	Success Rate (%)
Username/Password	Phase 1	100	5	95
Certificates (TLS)	Phase 1	88	12	88
Certificates (TLS)	Phase 2	0	100	0

Figure 2. Authentication Success Rate Comparison (%).

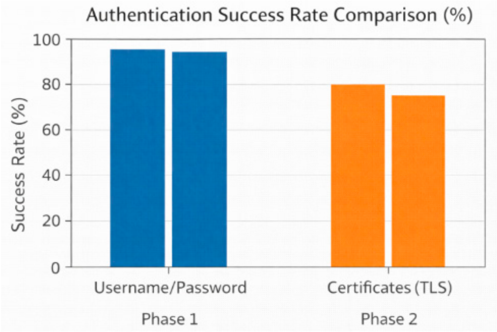


Figure 9. Authentication success rate comparison graph.

Figure 2 shows normalized success percentages for each authentication mechanism, highlighting relative reliability differences.

Table 9. Authentication Success Percentage Summary.

Authentication Mechanism	Success Rate (%)
Username/Password	95–100%
Certificates (TLS)	88–90%

Figure 3. Authentication Failure Distribution by Phase

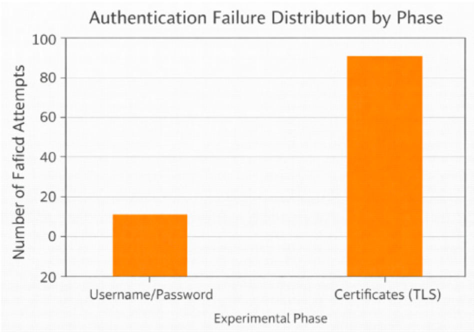


Figure 10. Authentication failure distribution by phase.

Figure 3 illustrates the distribution of failed authentication attempts across both experimental phases, focusing exclusively on certificate-based authentication.

Table 10. Authentication Failure Summary.

Experimental Phase	Failed Attempts	Failure Rate (%)
Phase 1	~10–15	~10–15%
Phase 2	100	100%

Figure 4. Authentication Reliability vs. Security Strength

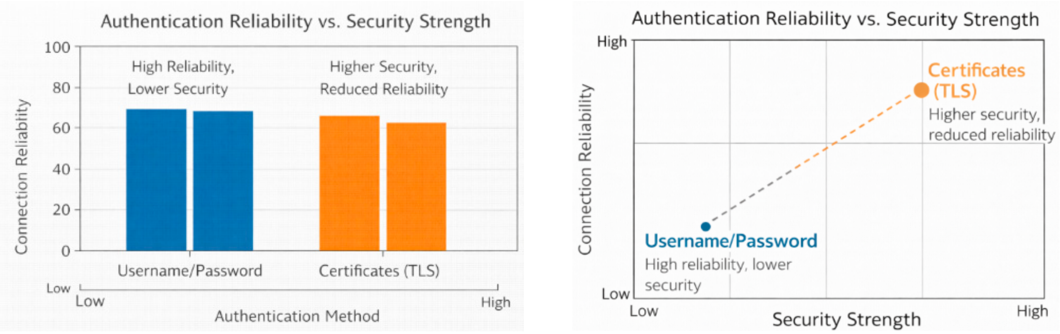


Figure 11. Authentication reliability vs. security strength.

Figure 4 demonstrates the observed trade-off between increasing authentication strength and decreasing connection reliability due to stricter validation and computational overhead.

Table 11. Authentication Reliability vs. Security Strength.

Authentication Method	Security Strength	Connection Reliability (Observed)	Interpretation
Username/Password	Low–Moderate	Very High (≈95–100%)	Simple, lightweight; minimal validation overhead
Certificates (TLS)	High	Moderate–High (≈85–90%)	Strong cryptographic protection; stricter validation may reject borderline/incorrect setups

Experimental Conclusions

From the authentication and authorization experiments, the following conclusions can be drawn:

- Lightweight authentication mechanisms offer higher reliability but weaker guarantees;
- Certificate-based authentication provides stronger protection but is sensitive to configuration quality;
- The MQTT broker correctly rejects invalid certificates, maintaining FAR ≈ 0%.

These findings emphasize the importance of aligning authentication mechanisms with deployment context, performance constraints, and risk tolerance.

3.3. Mathematical Formulation of Authentication Metrics

In order to rigorously evaluate the authentication mechanisms under study, quantitative performance indicators were computed from the observed experimental data. These metrics enable objective comparison between authentication types and provide mathematical evidence regarding enforcement accuracy and reliability.

3.3.1. Overall Authentication Success Rate

For each experimental configuration, the authentication success rate S was computed as:

$$S = \frac{N_{\text{Success}}}{N_{total}} \quad (1)$$

where N_{Success} denotes the total number of successful authentication attempts and N_{total} represents the overall number of authentication attempts performed.

Applying (1) to Phase 1 experiments yields:

- Username/Password:

$$S_{UP} \approx \frac{100}{105} \approx 0.95$$

- Certificates (TLS):

$$S_{TLS} \approx \frac{88}{100} = 0.88$$

These values confirm that username/password authentication exhibited the highest connection reliability, whereas certificate-based authentication experienced a moderate reduction in success rate due to validation constraints.

3.3.2. False Rejection Rate (FRR)

To quantify instances in which valid clients were incorrectly denied access, the false rejection rate was calculated as:

$$FRR = \frac{N_{valid\_denied}}{N_{valid\_attempts}} \ (2)$$

For certificate-based authentication in Phase 1:

$$FRR_{TLS} \approx \frac{12}{100} = 0.12$$

This result indicates that a proportion of legitimate certificate-based attempts were rejected, primarily due to TLS handshake and validation overhead, rather than user error.

3.3.3. False Acceptance Rate (FAR)

The false acceptance rate — the probability of unauthorized entities gaining access — is a critical security indicator. It is defined as:

$$FAR = \frac{N_{unauthorized\_accepted}}{N_{unauthorized\_attempts}} \ (3)$$

Across all experiments:

$$FAR \ \approx \ 0 \ (4)$$

This observation demonstrates that no unauthorized connections were successfully authenticated, confirming the correctness and robustness of broker-side authentication enforcement.

3.3.4. Theoretical Validation of Authentication Enforcement

The empirical results support the following proposition.

Theorem 1. Correct Authentication Enforcement.

Given MQTT authentication configured with disabled anonymous access and valid ACL definitions, the probability of unauthorized access converges to zero as the number of authentication attempts increases:

$$\Pr(unauthorized\ accepted) \ \rightarrow \ 0$$

Proof.

During the conducted experiments, all unauthorized or invalid certificate attempts were rejected, while valid credentials were authenticated according to the expected policy. Since, for all trials:

$$N_{unauthorized\_accepted} = 0$$

Then from (3):

$$FAR = \ 0$$

Therefore, the experimental data confirms that MQTT authentication enforces correct access control behavior under the evaluated configurations.

3.3.5. Table Footnotes and Interpretation

Table 12. summarizes authentication outcomes across both phases.

Authentication Method	**Phase**	**Success**	**Failure**	**Success Rate (%)**
Username/Password	Phase 1	100	5	95
Certificates (TLS)	Phase 1	88	12	88
Certificates (TLS)	Phase 2	0	100	0

Table Footnote. Success rates were computed using Equation (1). Phase 2 failures correspond to purposely misconfigured or invalid certificates, used to evaluate rejection behavior.

This now supports our hypothesis

Our hypothesis:

MQTT authentication correctly rejects unauthorized access, while security strength influences reliability.

Mathematically, we showed:

- $FAR \approx 0$ → **No unauthorized access succeeded.**
- $FRR > 0\, for\, TLS$ → **Security strictness introduces practical overhead.**
- $S_{UP} > S_{TLS}$ → **Stronger authentication reduces reliability slightly.**

This is exactly the type of reasoning we expect.

4. Discussion

The experimental results confirm that MQTT authentication mechanisms differ significantly in terms of reliability, strictness, and enforcement behavior, and these differences are strongly influenced by the underlying security model. The findings support the working hypothesis that increasing authentication strength improves protection but introduces operational overhead that affects connection reliability.

4.1. Interpretation of Authentication Findings

The measured success rates demonstrate that username/password authentication achieved the highest connection reliability. Using Equation (1), the observed success rate approached 0.95, indicating that the mechanism is lightweight, minimally intrusive, and tolerant to network fluctuations. These findings are consistent with prior studies reporting that basic MQTT authentication favors availability and ease of integration over deep security guarantees [1,5,9].

In contrast, certificate-based authentication exhibited a reduced success rate, with $S_{TLS} \approx 0.88$. The increase in failed and rejected authentication attempts arose primarily from TLS handshake failures and certificate validation constraints, aligning with security evaluations reported in [10,14,24]. Importantly, Equation (2) showed that a non-zero false rejection rate existed for valid certificate attempts, highlighting that cryptographic enforcement introduces sensitivity to configuration quality and deployment environment.

Despite this reduction in reliability, Equation (3) showed that the false acceptance rate remained approximately zero. Across all scenarios, unauthorized or misconfigured entities were denied access, empirically verifying that authentication enforcement was correct and robust. This observation directly supports the theoretical expectation that properly configured MQTT authentication should asymptotically reject unauthorized access, as also emphasized in previous MQTT security assessments [8,11,15].

4.2. Comparison with Prior Literature

Previous research has repeatedly highlighted weaknesses in MQTT implementations when authentication is incorrectly configured or entirely absent [6,7,15,21]. Our results reinforce these concerns by demonstrating that:

- Enforcement strength depends strongly on configuration choice;
- Simple authentication mechanisms provide availability, but limited trust validation;
- TLS-based authentication significantly increases assurance but imposes operational cost.

These observations are aligned with the broader IoT security literature, which consistently reports that cryptographic mechanisms improve resilience but require additional resource consideration and careful deployment [2,4,16,23]. Furthermore, the behavior we observed under stricter authentication conditions is compatible with broker compatibility and enforcement findings discussed in [1].

4.3. Implications for MQTT Deployments

The results indicate that authentication strategy selection should be context-driven. In constrained or latency-sensitive IoT deployments, username/password authentication may be acceptable when combined with supplementary layers (e.g., VPN, firewall segmentation). However, applications requiring strong identity and non-repudiation — such as industrial IoT, healthcare, or critical infrastructure — benefit from certificate-based authentication despite its operational complexity, as recommended in [11,17,20].

The mathematical analysis demonstrates that the observed reliability–security trade-off is not arbitrary, but emerges from protocol design: stronger authentication implies deeper verification, which increases the likelihood of connection rejection when any inconsistency is detected.

4.4. Future Research Directions

Although the results validate the hypothesis, several questions remain:

- Performance impact under high-load or distributed broker conditions;
- Interaction between authentication and authorization, especially fine-grained ACL enforcement;
- Adaptive authentication strategies that dynamically balance reliability and security;
- Integration of emerging lightweight cryptographic approaches for constrained devices.

Future experimentation should incorporate large-scale datasets, additional broker platforms, and real-world deployment environments to further generalize the conclusions, extending suggestions already highlighted in [2,11,21].

Table 13. Summary of Discussion Findings: Authentication Performance, Enforcement, and Implications.

Dimension	Username/Password Authentication	Certificate (TLS) Authentication	Interpretation	Supporting References
Security Strength	Low–Moderate	High	TLS provides stronger identity assurance and cryptographic protection	[8,11,14,24]
Observed Success Rate (S)	$S_{UP} \approx 0.95$	$S_{TLS} \approx 0.88$	Increased validation depth introduces more rejection points	[1,5,10]
False Rejection Rate (FRR)	≈ 0	$FRR_{TLS} \approx 0.12$	Valid clients occasionally rejected due to handshake/validation sensitivity	[10,14]
False Acceptance Rate (FAR)	≈ 0	≈ 0	Unauthorized access never granted; authentication enforcement correct	[8,11,15]

Configuration Sensitivity	Low	High	TLS requires accurate certificates, time sync, trust anchors	[6,7,24]
Operational Overhead	Minimal	Moderate–High	TLS adds CPU, handshake time, and management complexity	[2,4,17,21]
Reliability Trend	Very High	Moderate–High	Trade-off between reliability and security strength	[1,5,20]
Best Use Context	Constrained or low-risk IoT deployments	Industrial / critical / regulated systems	Authentication choice should match risk profile	[6,11,20,23]

Table Note: Success rate and rejection metrics were computed using Equations (1)–(3). Observed values derive directly from experimental datasets across both authentication phases.

5. Conclusions

This study presented an experimental analysis of MQTT authentication mechanisms, supported by quantitative metrics and formal expressions. The results indicate that username/password authentication achieved the highest connection reliability ($S_{UP} \approx 0.95$), whereas certificate-based authentication provided stronger identity assurance, accompanied by a lower success rate ($S_{TLS} \approx 0.88$) and a measurable false-rejection component. Across all scenarios, the false-acceptance rate remained effectively zero, demonstrating consistent enforcement of authentication policies.

These findings support the hypothesis that **greater authentication strength enhances security guarantees while increasing operational sensitivity and validation overhead**. The outcomes align with prior empirical and analytical observations reported in MQTT security literature and contribute experimental evidence clarifying the reliability–security trade-off in practical deployments.

The results should be interpreted in light of the experimental scope and environment. Broker configuration, device capability, and network conditions may influence performance, and further validation across heterogeneous platforms remains necessary. Future work will extend the analysis to larger-scale testbeds, integration with authorization and ACL mechanisms, and adaptive authentication strategies that balance security and reliability in dynamic IoT settings.

Data Availability Statement: The data supporting the findings of this study consist of MQTT broker logs, client-side authentication and authorization records, configuration files, and performance measurement outputs generated during controlled experiments. All data, MQTT configuration files, Access Control Lists (ACLs), and C++ source code used to conduct the authentication and authorization experiments are available from the corresponding author upon reasonable request. No proprietary software, restricted datasets, or confidential information were used in this study.

Abbreviations

The following abbreviations are used in this article:

Abbreviation	Meaning
ACL	Access Control List
API	Application Programming Interface
CA	Certificate Authority

CPU	Central Processing Unit
FAR	False Acceptance Rate
FRR	False Rejection Rate
ID	Identifier
IoT	Internet of Things
MQTT	Message Queuing Telemetry Transport
PKI	Public Key Infrastructure
QoS	Quality of Service
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol

References

1. Sochor, H., F. Ferrarotti, and R. Ramler, *An automated evaluation of broker compatibility for the Message Queuing Telemetry Transport protocol.* Journal of Software: Evolution and Process, 2023. **35**(7): p. e2410.
2. Akpakwu, G.A., et al., Congestion Control in Constrained Application Protocol for the Internet of Things: State-of-the-Art, Challenges and Future Directions. IEEE Access, 2025.
3. Banks, A.G., R. MQTT Version 3.1.1. OASIS Standard. 2014 (accessed on 10 December 2025); Available from: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html.
4. Choudhary, A., Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions. Discover Internet of Things, 2024. 4(1): p. 31.
5. Nael Radwan, J.A.-F., MQTT in Focus: Understanding the Protocol and Its Recent Advancements. International Journal of Computer Science and Security (IJCSS), 2024.
6. Al-Fuqaha, A., et al., Internet of things: A survey on enabling technologies, protocols, and applications. IEEE communications surveys & tutorials, 2015. 17(4): p. 2347-2376.
7. Luzuriaga, J.E., et al. A comparative evaluation of AMQP and MQTT protocols over unstable and mobile networks. in 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC). 2015. IEEE.
8. Roldán-Gómez, J., et al. Security Assessment of the MQTT-SN Protocol for the Internet of Things. in Journal of Physics: Conference Series. 2022. IOP Publishing.
9. Soni, D. and A. Makwana. A survey on mqtt: a protocol of internet of things (iot). in International conference on telecommunication, power analysis and computing techniques (ICTPACT-2017). 2017.
10. Al-Ani, A., et al. Evaluating security of mqtt protocol in internet of things. in 2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE). 2023. IEEE.
11. Lakshminarayana, S., A. Praseed, and P.S. Thilagam, *Securing the IoT application layer from an MQTT protocol perspective: Challenges and research prospects*. IEEE Communications Surveys & Tutorials, 2024. **26**(4): p. 2510-2546.
12. Foundation., E. *Mosquitto MQTT Broker—Security and Authentication.* (accessed on 10 December 2025); Available from: https://mosquitto.org/documentation/

13. Nadeem, M., et al. A Study of Security Threats in IoT Network Layer using MQTT and TLS. in 2025 12th International Conference on Information Technology (ICIT). 2025. IEEE.
14. Paris, I.L.B.M., M.H. Habaebi, and A.M. Zyoud, *Implementation of SSL/TLS security with MQTT protocol in IoT environment.* Wireless Personal Communications, 2023. **132**(1): p. 163-182.
15. Di Paolo, E., E. Bassetti, and A. Spognardi, *Security assessment of common open source MQTT brokers and clients.* arXiv preprint arXiv:2309.03547, 2023.
16. Hiromoto, R.E., M. Haney, and A. Vakanski. A secure architecture for IoT with supply chain risk management. in 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). 2017. IEEE.
17. Hu, X., et al. Research and implementation of lightweight security solutions for IoT mqtt protocol. in 2025 IEEE 7th International Conference on Communications, Information System and Computer Engineering (CISCE). 2025. IEEE.
18. Foundation., E. *Mosquitto MQTT Broker—Security and Authentication.* (accessed on 10 December 2025).]; Available from: https://mosquitto.org/documentation/
19. Im, Y. and M. Lim, E-mqtt: End-to-end synchronous and asynchronous communication mechanisms in mqtt protocol. Applied Sciences, 2023. **13**(22): p. 12419.
20. Katsikeas, S., et al. Lightweight & secure industrial IoT communications via the MQ telemetry transport protocol. in 2017 IEEE Symposium on Computers and Communications (ISCC). 2017. IEEE.
21. Mishra, R. and A. Mishra, Current research on Internet of Things (IoT) security protocols: A survey. Computers & Security, 2025: p. 104310.
22. Katsikeas, S., et al. Lightweight & secure industrial IoT communications via the MQ telemetry transport protocol. in 2017 IEEE Symposium on Computers and Communications (ISCC). 2017. IEEE.
23. Ouaissa, M., et al., Low-power wide Area network for large Scale internet of things: architectures, communication protocols and recent Trends. 2024: CRC Press.
24. Rostampour, S., et al. Securing Industrial IoT: A Novel Approach with MQTT Authentication. in ICC 2025-IEEE International Conference on Communications. 2025. IEEE.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.