








ISSN: 2617-6548

URL: [www.ijirss.com](http://www.ijirss.com)



## KTCCGM: Towards A novel solution for enhancing Kerberos-5 with threshold cryptography and ML-based anomaly detection

 Rami Almatarneh<sup>1</sup>,  Mohammad Aljaidi<sup>2</sup>,  Ayoub Alsarhan<sup>3\*</sup>,  Sami Aziz Alshammari<sup>4</sup>,  Nayef H. Alshammari<sup>5</sup>

<sup>1</sup>Department of Cybersecurity, Faculty of Information Technology, Zarqa University, Zarqa 13110, Jordan.

<sup>2</sup>Department of Computer Science, Faculty of Information Technology, Zarqa University, Zarqa 13110, Jordan.

<sup>3</sup>Dept of Information Technology, Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology, The Hashemite University, Zarqa, Jordan.

<sup>4</sup>Department of Information Technology, Faculty of Computing and Information Technology, Northern Border University, Rafha, Saudi Arabia.

<sup>5</sup>Department of Computer Science, Faculty of Computers and Information Technology, University of Tabuk, Tabuk. Saudi Arabia.

Corresponding author: Ayoub Alsarhan (Email: [ayoubm@hu.edu.jo](mailto:ayoubm@hu.edu.jo))

### Abstract

Since its introduction at MIT in 1993, the Kerberos 5 protocol has been a fundamental pillar of network authentication, using symmetric key cryptography and a centralized Key Distribution Center (KDC) to secure distributed computing environments. While it improved on its predecessors by offering stronger encryption and cross-domain functionality, it no longer fully meets the demands of modern systems due to its major drawbacks: the risk of a single point of failure in the KDC, vulnerability to password-based attacks, and a strict reliance on synchronized clocks for replay protection. To address these limitations, we recommend some significant modifications. Instead of a centralized KDC, we employ a network of nodes with the shared master key using threshold cryptography in such a way that even when part of the nodes are compromised, the system remains unaffected. To eliminate the need for synchronized clocks, we replace timestamp-based authentication with nonce-based authentication and a short-term cache for replay protection. To provide extra security against password attacks, we add machine learning-based anomaly detection, which monitors authentication patterns in real-time at all times. In case of suspicious activity, the system adaptively triggers adaptive multi-factor authentication (MFA). This context-aware adaptive MFA will wisely switch security features by location or device context, trying to strike a balance between security and convenience. Additionally, we optimize nonce management with efficient caching techniques to minimize storage overhead and enhance scalability by distributing the authentication load across multiple nodes. While these extensions significantly enhance Kerberos 5's resistance and adaptability to today's distributed systems, they come with trade-offs. A distributed KDC introduces some overhead and will have a minor impact on performance, while nonce handling, anomaly detection, and MFA consume additional computational resources. Our analysis shows, however, that these costs are counteracted by higher availability, increased resistance to attack, and increased flexibility within the authentication process. Future developments will focus on optimizing and scaling it. In rectifying Kerberos 5's inherent weaknesses, this work makes it ready for modernization in the context of large networks, allowing it to become a more stable and forward-thinking method of authentication.

**Keywords:** Authentication, KDC, Kerberos 5, Nonce-based security, Threshold cryptography.

**DOI:** 10.53894/ijirss.v8i3.7328

**Funding:** The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the (Grant Number: NBU-FFR-2025-2119-02).

**History:** Received: 14 April 2025 / Revised: 16 May 2025 / Accepted: 20 May 2025 / Published: 23 May 2025

**Copyright:** © 2025 by the authors. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Competing Interests:** The authors declare that they have no competing interests.

**Authors' Contributions:** All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.

**Transparency:** The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

**Publisher:** Innovative Research Publishing

## 1. Introduction

The Kerberos authentication protocol, so named after the three-headed dog of Greek mythology that stands guard over the underworld, was created in the late 1980s at the Massachusetts Institute of Technology (MIT) as part of Project Athena. The main objective of the protocol was to meet the growing demand for secure authentication in distributed computing systems, where mutual authentication between services and users was necessary over inherently insecure networks [1]. Kerberos was created to address the weaknesses of earlier protocols, particularly the transmission of plaintext passwords in services like Telnet and FTP, which made credentials susceptible to eavesdropping and replay attacks. The use of symmetric key cryptography and Key Distribution Center (KDC) enabled Kerberos to allow users to manage authentication without having to send passwords over the network, which makes Kerberos a cornerstone of modern enterprise security [2]. Before Kerberos 5, earlier releases like Kerberos 4 provided secure authentication but with some trade-offs. One of the primary issues was that Kerberos 4 employed the Data Encryption Standard (DES) for encrypting tickets, which ultimately became vulnerable to brute-force attacks with enhancements in computing power [3]. It also did not support cross-realm authentication, thereby restricting its scalability in multi-domain setups. The Needham-Schroeder protocol, an early predecessor of Kerberos, introduced the notion of a third-party trusted authenticator but was inadequate in resisting replay attacks since it lacked cryptographic protection of timestamps [4] (Table 1). The limitations of the early versions underscored the need for a more powerful system and therefore the development of Kerberos 5 in 1993. This new version introduced important features like renewable tickets, forwardable credentials, and support for stronger encryption methods such as AES, improving both security and user-friendliness compared to its predecessors [5].

**Table 1.**

Comparison of Authentication Protocols: NTLM, Needham-Schroeder, Kerberos 4, and Kerberos 5.

Feature	Needham-Schroeder Protocol	NTLM (NT LAN Manager)	Kerberos 4	Kerberos 5
Development Year	1978	1980s	1988	1993
Type	Authentication protocol based on symmetric key cryptography	Authentication protocol based on challenge-response	Network authentication protocol	Network authentication protocol
Encryption	Symmetric key encryption	DES (Data Encryption Standard)	DES (Data Encryption Standard)	Supports multiple encryption algorithms (AES, DES, 3DES)
Authentication Model	Client-server model with a trusted third-party	Challenge-response mechanism (password-based)	Client-server with a Key Distribution Center (KDC)	Client-server with a Key Distribution Center (KDC)
Ticketing	No ticketing mechanism	No tickets, relies on challenge-response	Uses tickets (TGT and service tickets)	Uses tickets (TGT, service tickets, and session tickets)
Security	Vulnerable to replay attacks	Vulnerable to replay and man-in-the-middle attacks	Vulnerable to replay attacks, no mutual authentication	Improved security with mutual authentication, anti-replay
Single Point of Failure	Yes, single point of failure in trusted third party	Yes, relies on a centralized authentication server	Yes, KDC as a single point of failure	Yes, KDC as a single point of failure

Feature	Needham-Schroeder Protocol	NTLM (NT LAN Manager)	Kerberos 4	Kerberos 5
		(in domain controller)		
Scalability	Limited scalability in larger environments	Poor scalability due to dependency on the central server	Limited scalability due to reliance on a single KDC	Improved scalability with cross-realm support
Cross-Domain Authentication	Not supported	Difficult, requires trusts between domains	Limited to same realm authentication, complex cross-realm support	Supports cross-realm authentication with improved protocols
Clock Synchronization Requirement	No specific requirement	No, works without synchronized clocks	Yes, requires synchronized clocks	Yes, requires synchronized clocks
Mutual Authentication	Yes, mutual authentication	No, vulnerable to relay attacks	No, vulnerable to relay attacks	Yes, provides mutual authentication
Forward and Backward Secrecy	No	No, relies on static keys	No	Yes, provides forward and backward secrecy
Widely Used In	Academic environments and early distributed systems	Microsoft networks (Windows NT, Active Directory)	MIT, early enterprise systems	Modern enterprise networks, large-scale organizations
Main Limitation	Vulnerable to replay attacks, no built-in ticketing	Vulnerability to offline dictionary attacks, weak security	Limited to DES, no support for stronger encryption, lack of flexibility	Centralized KDC, clock synchronization dependency

To the best of our knowledge, all previous works did not take into consideration the inherent fragility of KDC replication strategies, which remain vulnerable to coordinated attacks, nor the persistent reliance on clock synchronization despite alternatives like sequence numbers introducing session management complexities. In this work, we have considered a decentralized KDC with threshold cryptography to eliminate the single point of failure and nonce-based authenticators to bypass clock dependency, ensuring strong security and flexibility of operation. Nevertheless, these enhancements may lead to some minor drawbacks, which may include increased system complexity and potential performance overhead caused by multi-node coordination and nonce cache management.

The key contributions of this paper are as follows:

- Decentralized KDC with Threshold Cryptography: Distributes the KDC's master key across multiple nodes, eliminating the single point of failure and boosting security and availability.
- Nonce-Based Authenticators: Substitutes timestamp reliance with nonces and a time-limited cache, ensuring replay protection without the need for synchronized clocks.
- Enhanced Resilience: Adapts Kerberos to modern, distributed environments by tackling key operational weaknesses effectively.

## 2. Security Challenges and Vulnerabilities in Kerberos 5

According to a 2023 Gartner benchmark [6], although Kerberos 5 has made significant improvements, it still faces significant challenges because of its centralized structure and dependence on password-derived keys, which make it an attractive target for attackers. The Key Distribution Center (KDC) is a critical component of the Kerberos 5 system; however, its failure would disrupt authentication processes and prevent users from accessing services, making it a single point of failure. Microsoft [7] Digital Defense Report showed that 18% of Active Directory outages (systems that rely on Kerberos) were caused by denial-of-service (DoS) attacks aimed at the KDC and have caused widespread disruptions to authentication services across networks [7]. Credential-based attacks are also common. The 2021 CrowdStrike Global Threat Report indicated that 22% of credential compromise campaigns leveraged Kerberos' pre-authentication function to enable attackers to brute-force weak passwords and generate spoofed Ticket Granting Tickets (TGTs) [8]. Furthermore, Microsoft's 2022 report identified that 40% of breaches involving Kerberos were due to reused or easily guessable passwords, illustrating the protocol's reliance on strong user secrets [9].

Advanced attackers also exploit Kerberos-specific vulnerabilities, such as Golden and Silver Ticket attacks, to generate simulated authentication tickets to mimic services or actual users. Mandiant's 2023 M-Trends Report found that 12% of the advanced persistent threat (APT) attacks employed these techniques, often bypassing legacy defenses since they are not looking for out-of-the-ordinary TGT requests [10]. Even Kerberos' timestamp-based replay protection, which aims to prevent attackers from replaying captured tickets, can be problematic. A 2020 study found that 15% of authentication failures in large organizations were due to clock synchronization, providing attackers with chances to exploit time-related vulnerabilities [11].

The scalability of Kerberos 5 is also strained in the large-scale environments of today. Enterprises with over 50,000 users have 3x higher authentication latency under load compared to token-based solutions, states a 2023 Gartner benchmark [6]. This bottleneck not only takes away from user experience but also raises the risk of outages in mission-critical systems. Even with efforts to deploy redundant KDCs, studies show that 35% of organizations with clustered KDCs still suffer from synchronization failures, which lead to cascading authentication breakdowns [12].

While Kerberos has had a historic function, revolutionizing authentication via decentralization of trust and password elimination over networks, its pre-cloud design roots are flawed for modern hybrid environments. The centralized KDC, as successful as it has been, is at odds with today's zero-trust architectures that require distributed power. Likewise, the protocol's reliance on static passwords conflicts with the growing need for phishing-resistant multi-factor authentication (MFA) and hardware-protected credentials. Despite these limitations, Kerberos remains a mainstay of enterprise security, especially in Windows environments, as it supports Active Directory. Its longevity attests to both its initial strengths and the pressing need for a refresh a problem being addressed by innovative solutions like threshold cryptography for distributed KDCs and anomaly detection using machine learning [13, 14].

### **3. Literature Review on the Limitations and Challenges of Kerberos 5**

For decades, Kerberos 5 has been a cornerstone of network authentication and has been widely adopted in enterprise environments for its ability to provide secure, single sign-on access across distributed systems. Using symmetric key cryptography and a trusted third-party model, it has been the system of choice for securing communications on corporate networks and university campuses. It is not entirely perfect, though. Over the years, researchers have discovered several limitations that can undermine its effectiveness, ranging from architectural vulnerabilities to real-world deployment problems.

One of the most critical limitations of Kerberos 5 is its dependence on a centralized Key Distribution Center (KDC), which serves as the linchpin of the system by issuing tickets and managing keys. If the KDC crashes or is attacked, then the whole authentication system also collapses, which leads to the creation of a significant single point of failure. This centralization, although efficient in most environments, exposes Kerberos to denial-of-service (DoS) attacks and compromises its resilience, especially in large-scale mission-critical systems where high availability is of utmost importance. Various solutions have been proposed by researchers to mitigate this, including KDC replication, where multiple KDCs replicate each other so that the system still functions in case one fails. Synchronizing these replicas, though, can create new security threats. Another solution that has been suggested is distributed KDCs and threshold cryptography, which distributes the master key across a number of KDCs so that some number of them have to agree before they can authenticate, which enhances both fault tolerance and security but introduces additional system complexity [15, 16].

Kerberos 5 also suffers from its reliance on user passwords for authentication. The protocol obtains a key by extracting it from a password and then uses it in the authentication process. Weak passwords, however, make the system susceptible to offline guessing attacks, where an attacker can obtain the encrypted data from the authentication exchange and attempt to crack it by using a guess dictionary. To counter this, Kerberos 5 implemented pre-authentication, which forces clients to prove that they know the password before issuing a Ticket Granting Ticket (TGT). Still, weak passwords remain a vulnerability. Security can be improved with multi-factor authentication (MFA) or by using machine learning to detect unusual login patterns, both of which can significantly increase the difficulty for attackers [17-19].

The other major issue with Kerberos 5 is its dependency on accurate time synchronization. The protocol uses timestamps to prevent replay attacks, but this practice can cause problems in distributed systems where clocks are not necessarily synchronized. With discrepancies between system clocks, it can result in authentication failure or undetected replay attacks. Other ideas have suggested using sequence numbers rather than timestamps to avoid synchronization issues, but this would create other challenges related to how to manage these numbers across different sessions, besides it would add extra steps to the protocol [20, 21].

Kerberos also faces challenges regarding authorization. While it shows effectiveness in authenticating users, it does not provide a built-in mechanism for controlling what resources a user can access once authenticated, which can lead to inconsistent or excessively broad permissions. To counter this, some researchers have proposed embedding the authorization information process, such as Privilege Attribute Certificates (PACs), within Kerberos tickets to specify user roles and permissions. Another method is to combine Kerberos with attribute-based access control (ABAC), in which user attributes are computed to make dynamic access decisions in real time, thereby increasing the flexibility of the system in complicated environments [22, 23].

Scalability is also an issue for Kerberos 5. The expansion of networks raises the demand on the KDC, creating potential bottlenecks. To help with this, proposals such as hierarchical KDC topologies have been made, in which two or more levels of KDCs share the workload, enhancing performance for large systems. Another approach is credential caching, where credentials are stored closer to services or users in an attempt to reduce the number of requests to the KDC. While caching may boost performance, it also carries the risk of having stale credentials used unless properly managed [24, 25].

Finally, Kerberos 5 struggles with cross-domain authentication, as it was initially designed for a single realm. Kerberos becomes complicated when users need to authenticate across multiple realms, such as in federated or cloud environments. The process of establishing trust between different realms requires manual configuration of keys and policies, which is both error-prone and cumbersome. To simplify this, researchers have proposed dynamic trust protocols for realms to establish trust more easily, as well as using public-key cryptography to link realms without pre-shared keys, which could reduce administrative overhead [26, 27].

Although Kerberos 5 is still a secure and trustworthy authentication system, its drawbacks in terms of centralization, password-based security, time synchronization, authorization, scalability, and cross-domain authentication cannot be ignored. Its weaknesses have been addressed by the research community, with proposed solutions ranging from distributed KDCs to improved authentication mechanisms. These developments continue to evolve to ensure that Kerberos remains a feasible and secure option for authentication over a network, regardless of increasingly complex and dynamic security landscapes.

#### 4. Proposed Enhancement for Kerberos 5

Despite its robust foundation as a network authentication protocol, as already specified, Kerberos 5 exhibits critical limitations that undermine its efficacy in modern distributed systems, including its centralized Key Distribution Center (KDC), which creates a single point of failure, and its dependency on synchronized clocks for replay protection. In this section, we will review these limitations and propose enhancements to improve and increase the resilience and adaptability of Kerberos 5, leveraging innovative cryptographic and procedural advancements to align with the demands of contemporary security environments.

##### 4.1. Distributed KDC Architecture

In network security, the Kerberos 5 protocol is widely popular for its robust authentication protocols to ensure secure communication across distributed systems. With all its positives, a significant architectural limitation persists: dependence on a centralized Key Distribution Center (KDC), denoted as  $KDC_c$ , which exclusively manages the issuance of cryptographic tickets and session keys to clients and services [28]. This centralized design introduces a critical security vulnerability, that appears as a single point of failure. If  $KDC_c$  is unavailable or compromised, this absolutely means that the entire authentication infrastructure is at risk of collapse, undermining the reliability of the system. In existing Kerberos 5 systems, an authenticating client  $C$  requests a ticket-granting ticket (TGT) from  $KDC_c$ , which maintains the master key  $K_m$ . This process is formally expressed as:

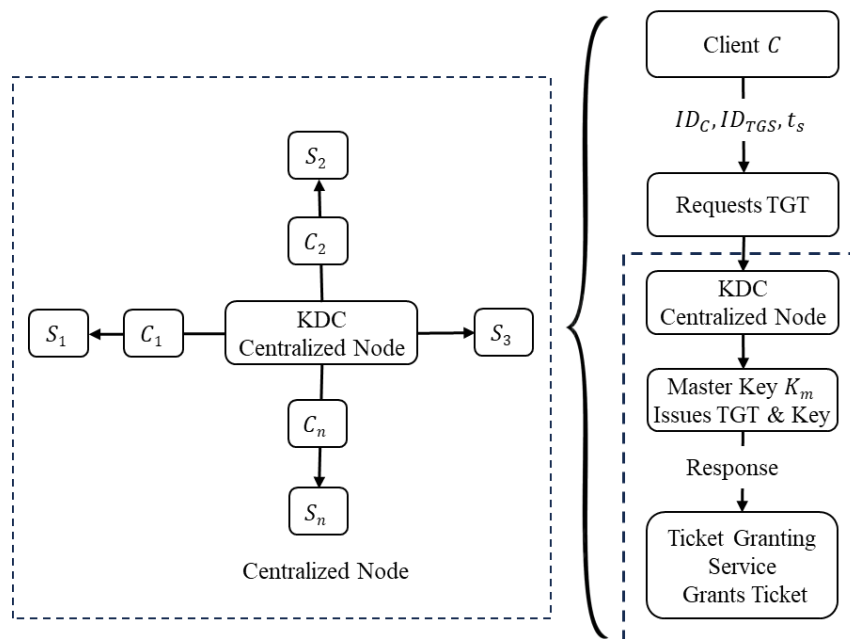
$$C \rightarrow KDC_c: \{ID_C, ID_{TGS}, t_s\}$$

where  $ID_C$  represents the client's identifier,  $ID_{TGS}$  denotes the ticket-granting service identifier, and  $t_s$  is a timestamp ensuring request freshness.

In response,  $KDC_c$  generates and returns:

$$KDC_c \rightarrow C: \{TGT, K_{C,TGS}\}_{K_m}$$

where,  $K_{C,TGS}$  is the session key, encrypted using  $K_m$ . The system vulnerability is exposed when  $KDC_c$  fails, denying all subsequent authentication procedures and exposing the protocol's intrinsic vulnerability on a single entity (see Figure 1).



**Figure 1.**  
Centralized KDC: Single Point of Failure.

To mitigate this limitation, we propose a new enhancement to Kerberos 5 by shifting from a centralized KDC to a distributed network of KDC nodes based on threshold cryptography principles [29]. In the new revised architecture, the master key  $K_m$  is no longer held by a single  $KDC_C$ . Instead, it is split into  $n$  shares according to Shamir's Secret Sharing scheme, represented as  $S = \{s_1, s_2, \dots, s_n\}$ , so that each share  $s_i$  distributed to a unique KDC node  $KDC_i$  (where  $i = 1, 2, \dots, n$ ) [30]. Reconstruction of  $K_m$  requires a minimum threshold  $t$  of these shares, where  $t \leq n$  and typically  $t > \frac{n}{2}$  to establish a quorum, balancing security and practicality. The modified authentication process is delineated as follows: the client  $C$  interacts with a subset of  $t$  KDC nodes, submitting requests for their respective shares:

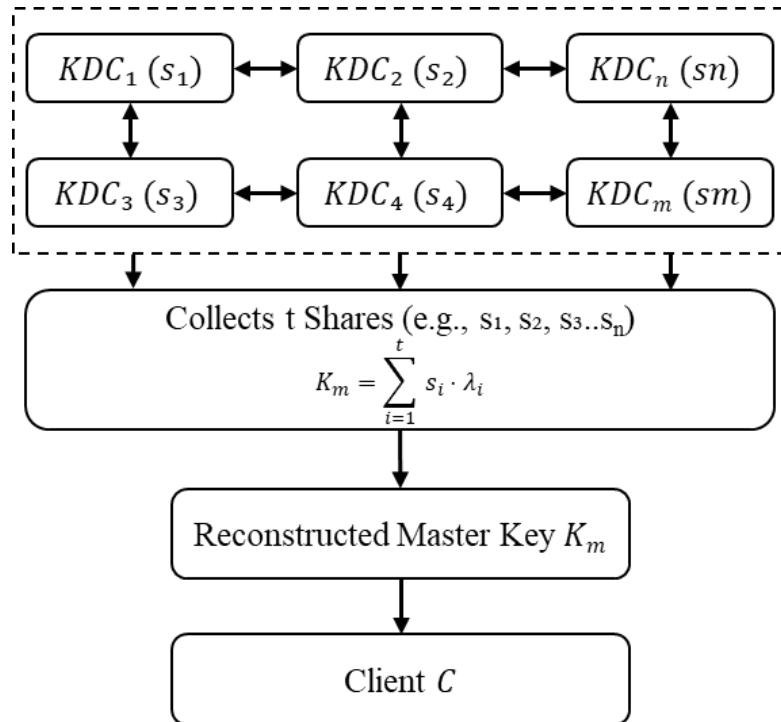
$$C \rightarrow KDC_i: \{ID_C, ID_{TGS}, t_s\} \quad \text{for } i = 1, 2, \dots, t$$

Each  $KDC_i$  responds by providing its share  $s_i$ . Upon collecting  $t$  shares,  $C$  reconstructs  $K_m$  using the interpolation formula:

$$K_m = \sum_{i=1}^t s_i \cdot \lambda_i$$

where  $\lambda_i$  denotes the Lagrange coefficients associated with the selected shares [31]. With  $K_m$  reconstituted,  $C$  can either locally compute the requisite cryptographic components, such as  $\{TGT, K_{C,TGS}\}_{K_m}$ , or engage a designated node to finalize the authentication process (see Figure 2).

This distributed KDC framework offers significant improvements over the traditional centralized system. First, it eliminates the single point of failure by enabling the system to function as long as at least  $t$  of the  $n$  KDC nodes are functioning, hence enhancing fault tolerance. Secondly, security is enhanced because no single  $KDC_i$  will carry the full  $K_m$ , hence, the key compromise is less likely [29]. Compared to conventional replication solutions where each node maintains a replica copy of  $K_m$  still susceptible to concerted attacks the threshold cryptography approach provides a decentralized approach optimizing security and availability [28]. The proposal borrows from studies in distributed authentication systems but offers an in-house adaptation tailored specifically to satisfy Kerberos 5's architectural constraints [32].



**Figure 2.**  
Distributed KDC with Threshold Cryptography: Eliminates Single Point of Failure.

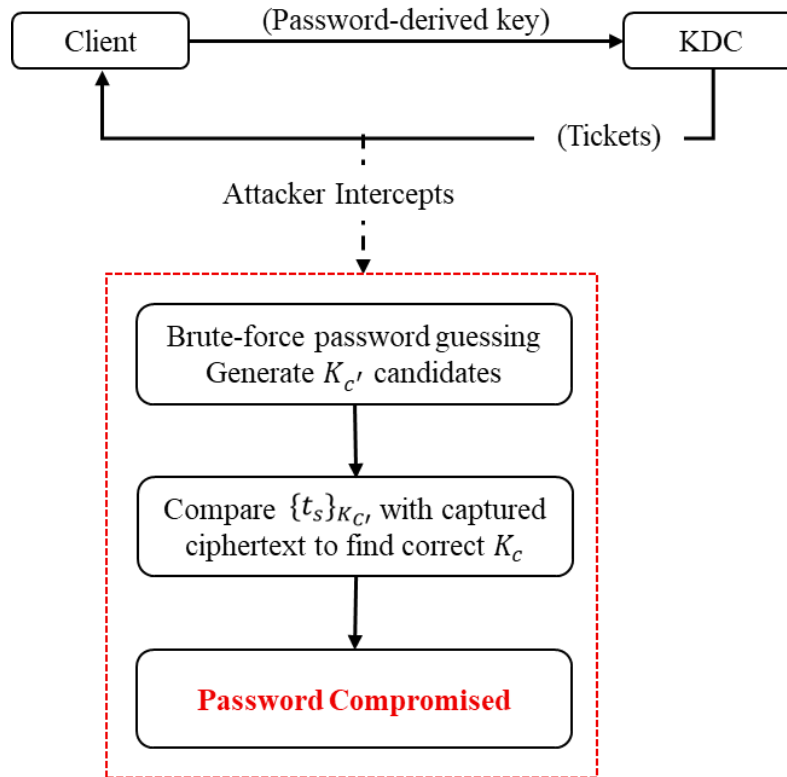
By integrating sophisticated cryptographic techniques, such as Shamir's Secret Sharing, the distributed KDC model proposed successfully overcomes the vulnerabilities based on Kerberos 5's centralized nature. Besides providing improved fault tolerance and security, this enhancement also represents a move toward constructing resilient authentication models sensitive to the needs of contemporary network security.

#### 4.2. ML-Based Anomaly Detection for Password Vulnerability

The Kerberos 5 authentication protocol, although being extensively used for secure network authentication, has a critical vulnerability due to its dependence on password-derived keys, making it vulnerable to offline password-guessing attacks, especially when users select weak or easy-to-guess passwords [33]. In the present scheme, the authentication starts with the client  $C$  initiating a request for a ticket-granting ticket (TGT) to the Key Distribution Center (KDC). This request includes a timestamp  $t_s$ , encrypted under the client's password-based key  $K_C$ , as follows:

$$C \rightarrow KDC: \{t_s\}_{K_C}$$

The KDC decrypts this message using the stored  $K_C$  to authenticate the client. However, if an attacker intercepts this encrypted timestamp, they can perform an offline brute-force attack by guessing passwords, computing candidate keys  $K_{C'}$ , and determining if  $\{t_s\}_{K_{C'}}$  matches the intercepted ciphertext (see Figure 3). This vulnerability is a direct consequence of the protocols using passwords, which are typically the weakest link in authentication protocols due to human tendencies to select easily guessable credentials [34].



**Figure 3.**  
Password-based Authentication: Vulnerable to Offline Guessing.

To mitigate this weakness, we suggest introducing an ML-based anomaly detection system to track authentication requests in real time to improve the resilience of Kerberos 5 against password-guessing attacks. This solution uses a supervised or unsupervised ML model to learn behavior patterns in authentication attempts based on some features like the frequency of authentication requests, source IP addresses, success-to-failure ratios, and time intervals between attempts. These features are encapsulated in a feature vector  $\mathbf{x}$ , defined as:

$$\mathbf{x} = [f_{\text{attempts}}, \text{IP}, r_{\text{success}}, \Delta t]$$

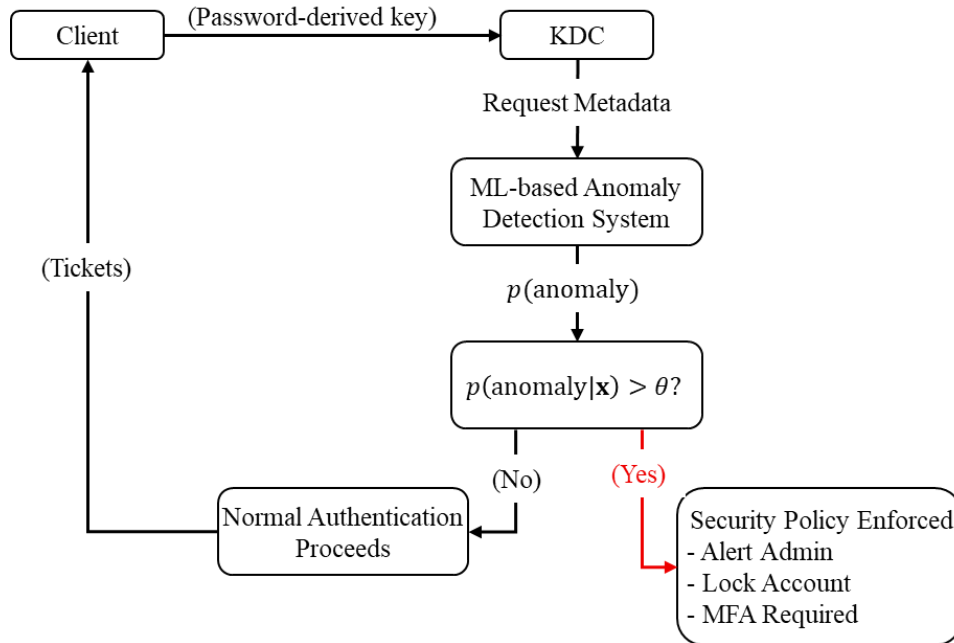
Here,  $f_{\text{attempts}}$  represents the number of authentication attempts within a specified time window, IP denotes the source IP address,  $r_{\text{success}}$  is the ratio of successful to total attempts, and  $\Delta t$  is the time elapsed since the previous attempt. The ML model, potentially a Random Forest or Neural Network classifier is trained to find the probability of an anomaly,  $p(\text{anomaly}|\mathbf{x})$ , based on historical and real-time data. If this probability exceeds a predefined threshold  $\theta$ , i.e.,  $p(\text{anomaly}|\mathbf{x}) > \theta$ , the system initiates a response (see Figure 4), such as:

- Triggering alerts to notify system administrators.
- Temporarily locking the user account to prevent further attempts.
- Requiring additional verification, such as multi-factor authentication.

The enhanced authentication process can be symbolized as follows:

1.  $C \rightarrow KDC: \{t_s\}_{K_C}$  (standard request).
2. The KDC processes the request while simultaneously extracting metadata (e.g., IP, timestamp) and feeds it into the ML model.
3. The ML model evaluates  $\mathbf{x}$  and find  $p(\text{anomaly}|\mathbf{x})$ .
4. If  $p(\text{anomaly}|\mathbf{x}) > \theta$ , the KDC enforces a security policy:
  - a. if  $p(\text{anomaly}|\mathbf{x}) > \theta$ , then trigger response
  - b. Otherwise, the authentication proceeds as normal.





**Figure 4.**  
ML-based Anomaly Detection: Proactively Mitigates Password-Guessing Attacks.

This machine learning-based approach significantly improves the static defenses in Kerberos 5. By detecting suspicious behavior, such as multiple failed login attempts from a single IP, the system prevents password-guessing attacks before they can succeed, rather than relying only on the strength of  $K_C$  [35]. Further, the ML model can also be retrained to detect new attack patterns, thereby making it effective against evolving threats, which traditional pre-authentication methods can't adapt to [36]. This adds a more responsive layer of defense, in line with current cybersecurity practices that use machine learning to better detect threats [37].

This approach has two important benefits. First, it provides real-time protection by detecting and responding to unusual patterns that suggest password-guessing attempts. Second, its ability to adapt to new attack methods through continuous learning strengthens the long-term security of Kerberos 5. Unlike the current system, where security depends on the strength of user passwords, this approach shifts the responsibility to an active monitoring system, effectively reducing the risk of offline attacks.

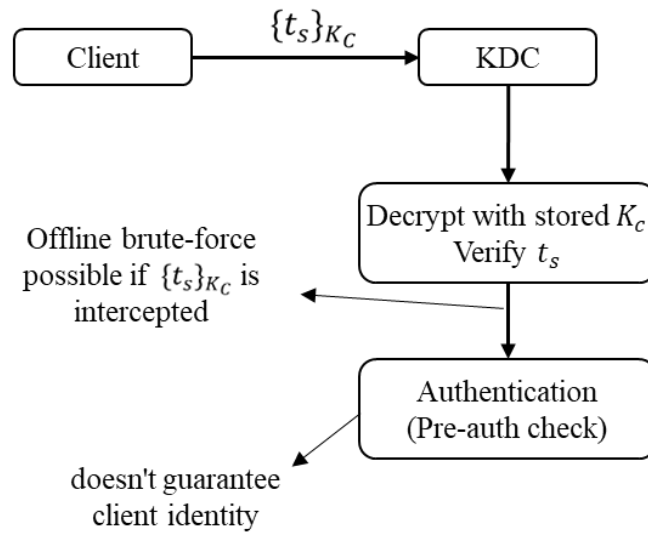
#### 4.3. Adaptive Multi-Factor Authentication (MFA) for Password Security

To derive cryptographic keys, Kerberos 5 protocol relies heavily on user-chosen passwords, a dependency that introduces significant security vulnerabilities [38]. In its current implementation, Kerberos 5 utilizes a pre-authentication mechanism to counter offline password-guessing attacks. Here, the client  $C$  encrypts a timestamp  $t_s$  with its password-derived key  $K_C$  and sends it to the Key Distribution Center (KDC):

$$C \rightarrow KDC: \{t_s\}_{K_C}$$

The KDC decrypts the message using the stored  $K_C$  to verify the timestamp. While this approach prevents attackers from obtaining password hashes without interacting with the KDC, but it does not address the actual issue of weak passwords and does not offer any assurance regarding the client's identity. Studies reported that users often prefer low-entropy passwords, which are susceptible to dictionary or brute-force attacks even when pre-authentication protection is available [39]. Thus, the current system remains vulnerable, as the strength of  $K_C$  is directly tied to the user's password choice (see Figure 5).





**Figure 5.**  
Password-based Authentication: Vulnerable to Offline Guessing.

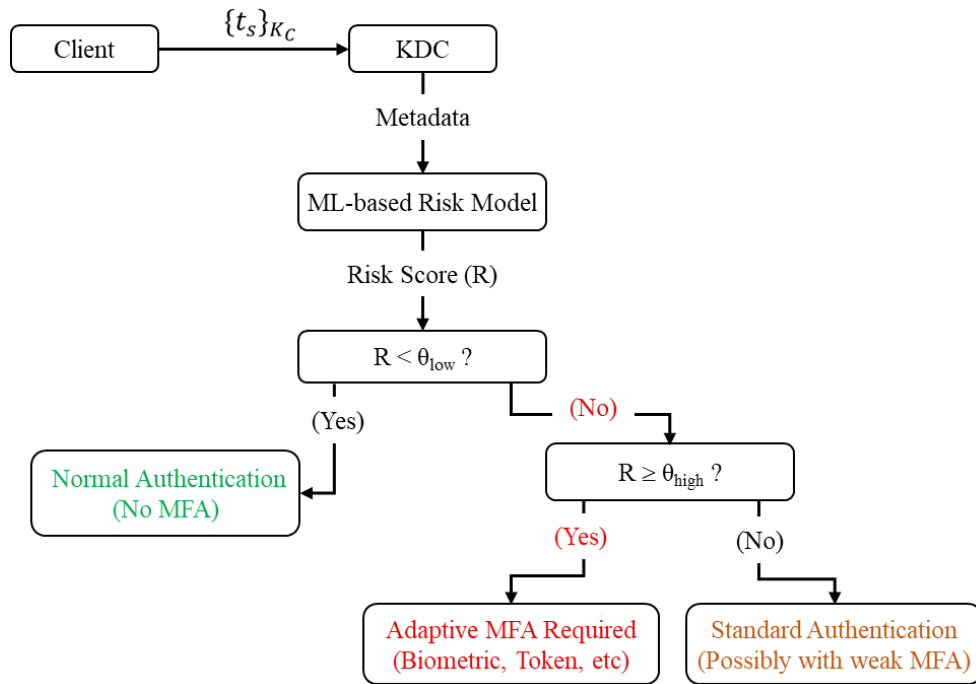
To overcome this limitation, we propose the integration of an adaptive multi-factor authentication (MFA) mechanism that utilizes a machine learning (ML)-based risk model into Kerberos 5 to dynamically adjust authentication needs. The model takes into account contextual factors for each login request, such as source IP address, device fingerprint, access timing, and recent authentication success or failure rates, represented as a feature vector:

$$\mathbf{x} = [\text{IP, device, } t_{\text{access}}, r_{\text{success}}, \dots]$$

Using a classifier (e.g., logistic regression or a decision tree), the model computes a risk score  $r(\mathbf{x}) \in [0,1]$ , where higher values indicate increased suspicion of malicious activity. Predefined thresholds  $\theta_{\text{low}}$  and  $\theta_{\text{high}}$  guide the authentication process:

$$\text{Authentication Requirement} = \begin{cases} \text{Password only,} & \text{if } r(\mathbf{x}) < \theta_{\text{low}} \\ \text{Password + MFA,} & \text{if } r(\mathbf{x}) \geq \theta_{\text{high}} \end{cases}$$

In low-risk circumstances, such as routine logins from a trusted device where  $r(\mathbf{x}) < \theta_{\text{low}}$ , standard password authentication, relying solely on  $K_C$ , is sufficient. Conversely, in high-risk scenarios, e.g., logins from an unusual IP address or following multiple failed attempts, where  $r(\mathbf{x}) \geq \theta_{\text{high}}$  the system mandates MFA, requiring additional factors like a biometric scan or hardware token. This adaptive approach ensures that authentication strength scales with the perceived threat level. This adaptive approach guarantees that authentication strength scales with the perceived threat level and offers distinct advantages over the static pre-authentication mechanism in Kerberos 5. By tying authentication requirements to real-time risk analysis, it strengthens security precisely where weak passwords pose the greatest risk, addressing a persistent vulnerability [40]. Simultaneously, it also lowers user friction by reserving MFA for high-risk conditions, avoiding the unnecessary prompts that often frustrate users in traditional MFA systems [41]. Unlike static MFA, which applies uniform criteria regardless of context, this innovation dynamically adjusts based on threat analysis, striking an optimal balance between security and usability [42]. For instance, when a user accesses the system from a known location and device, the authentication process is smooth. However, if an unusual attempt is detected, stronger verification steps are required, increasing security without making the process inconvenient (see Figure 6).



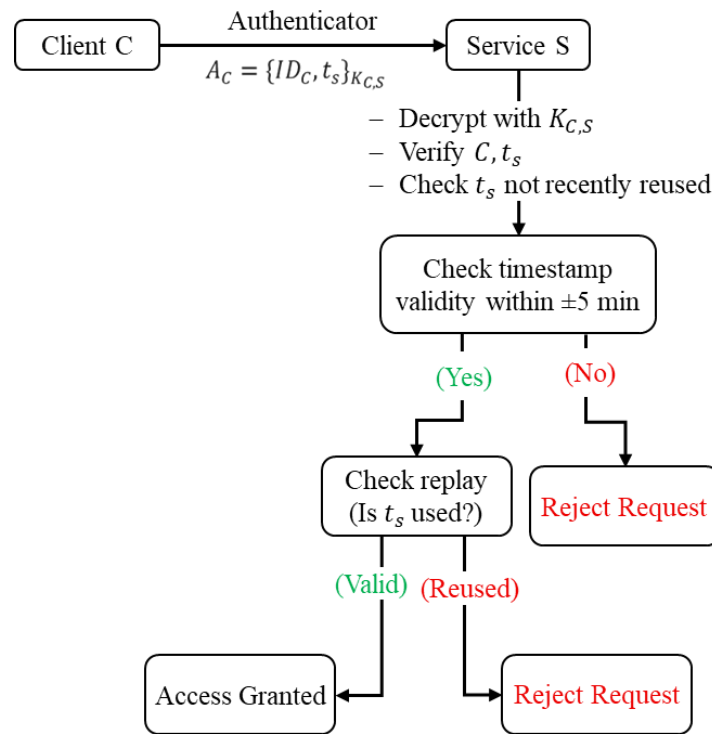
**Figure 6.**  
ML-based Anomaly Detection: Proactively Mitigates Password-Guessing Attacks.

#### 4.4. Challenge-Response Mechanism for Clock Synchronization Dependency

The Kerberos 5 protocol is designed to provide secure authentication in distributed systems, but it has a key limitation due to its dependence on clock synchronization to prevent replay attacks [43]. In its current configuration, replay protection is achieved by embedding timestamps within authenticators; thus, when a client  $C$  wants to access an application service  $S$ , it creates an authenticator  $A_C$  that includes its identity  $ID_C$  and a timestamp  $t_s$ , where both are encrypted using the session key  $K_{C,S}$ :

$$A_C = \{ID_C, t_s\}_{K_{C,S}}$$

The service  $S$  decrypts  $A_C$  using  $K_{C,S}$ , verifies that  $t_s$  falls within a reasonable time interval, which is typically  $\pm 5$  minutes of its local clock, and checks that the timestamp has not been recently reused [44]. This approach presupposes that the clocks of  $C$  and  $S$  are precisely synchronized, a circumstance that can be impractical in large-scale or geographically dispersed environments where network latency, clock drift, or administrative misconfigurations may disrupt time alignment [45]. Such dependency complicates the deployment process and introduces potential vulnerabilities in case of synchronization failure, necessitating the need for an alternative mechanism (see Figure 7).



**Figure 7.**  
ML-based Anomaly Detection: Proactively Mitigates Password-Guessing Attacks.

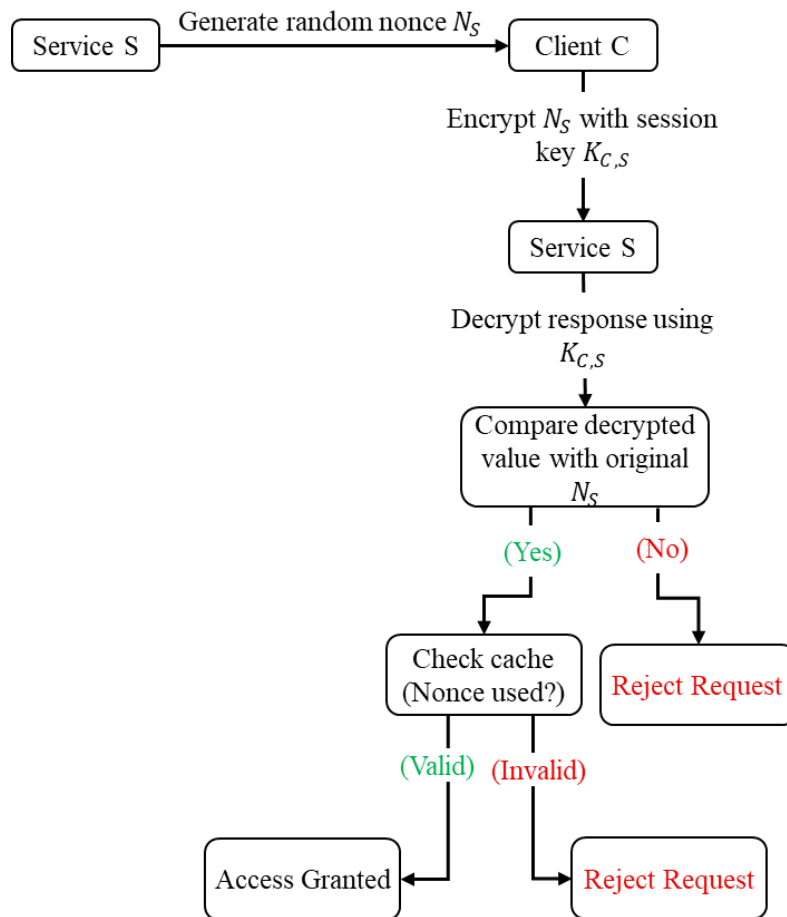
To overcome this challenge, we propose a challenge-response mechanism as a substitute for timestamp-based authenticators that doesn't need clock synchronization and yet offers good replay protection. In the proposed method, during the Application Service (AP) exchange, the service  $S$  generates a random nonce  $N_S$  and sends it to the client  $C$ , then the client encrypts the nonce with the session key  $K_{C,S}$  and returns the response to  $S$ :

$$C \rightarrow S: \{N_S\}_{K_{C,S}}$$

The service  $S$  decrypts the response to verify that the decrypted value is matched with  $N_S$ , and validates that  $N_S$  has not been used before by consulting a short-term cache of recent nonces. The process can be formalized as follows:

1.  $S \rightarrow C: N_S$  (Service issues a random nonce).
2.  $C \rightarrow S: (N_S)_{K_{C,S}}$  (Client encrypts and returns the nonce).
3.  $S$  verifies the decrypted  $N_S$  against the original and checks its cache.

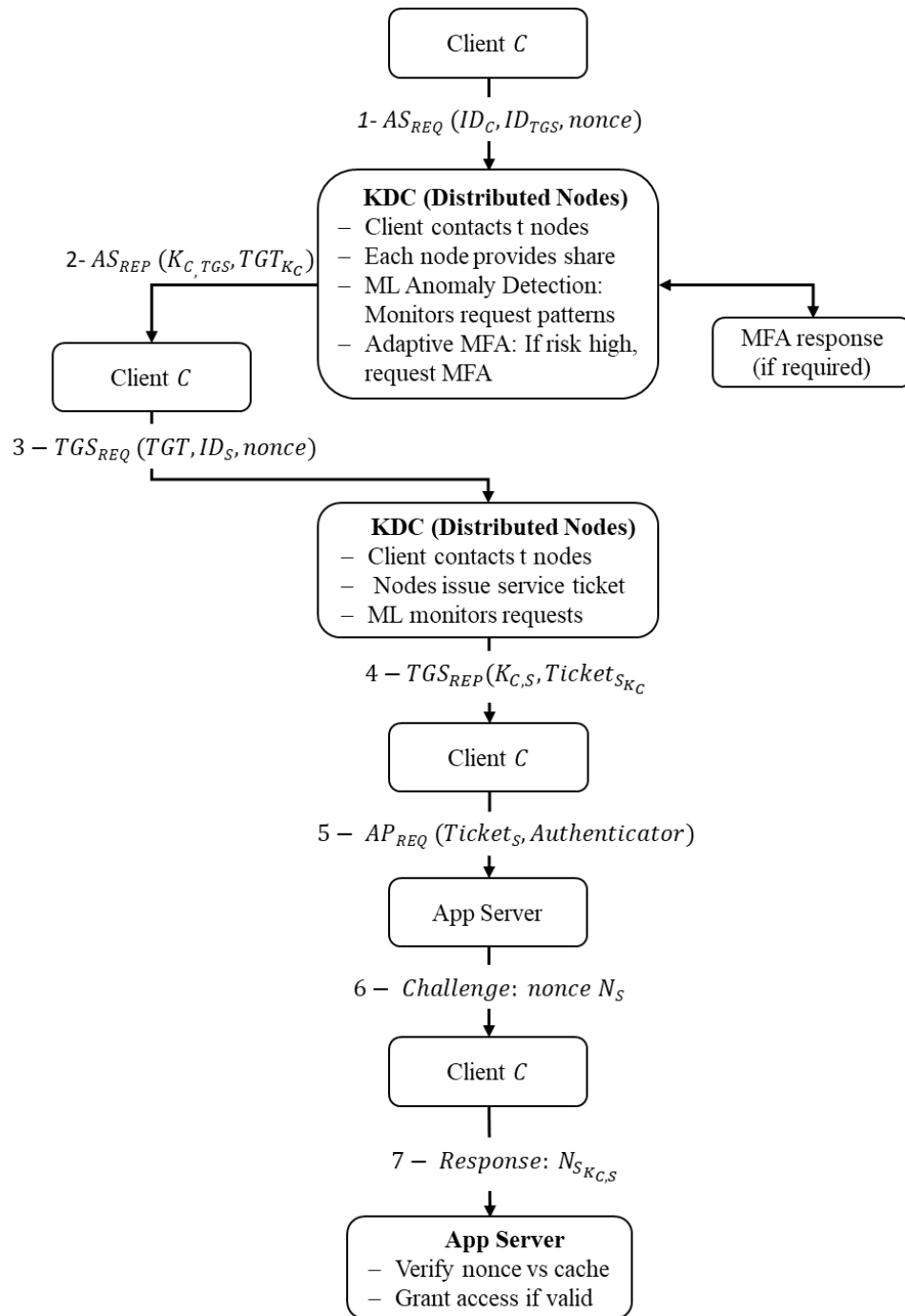
If the nonce is valid and unused,  $S$  grants access; otherwise, the request is rejected. This approach ensures that only a client possessing  $K_{C,S}$  can respond correctly, while the nonce cache prevents replay attacks without relying on time (see Figure 8).



**Figure 8.**  
ML-based Anomaly Detection: Proactively Mitigates Password-Guessing Attacks.

This proposed mechanism has some special advantages over the current timestamp-based mechanism. First, it eliminates the requirement for synchronized clocks, hence making Kerberos 5 more desirable to be used in environments where precise time synchronization is challenging, such as mobile networks or cross-domain systems [46]. Replay protection remains effective due to the randomness of nonces and the confidentiality of  $K_{C,S}$ , hence, the intercepted responses cannot be replayed [47]. Unlike the timestamp scheme, which may be susceptible to clock skewing or deliberate time modification, the challenge-response approach is inherently resistant to these attacks, hence making the protocol more flexible and secure.

The innovation of this approach lies in its flexibility and compatibility with the existing infrastructure. Thus, by introducing the challenge-response mechanism as an optional extension, this will enable Kerberos 5 to accommodate diverse network conditions without requiring immediate updates to all clients. Services can choose between using timestamps or nonces during the AP exchange, allowing compatibility with legacy systems while also enhancing reliability [48]. This dual-mode capability makes authentication more adaptable to different operational needs.



**Figure 9.**  
Proposed Kerberos enhancements and their relationships within the system.

#### 4.4.1. Implementation Considerations

- **Backward Compatibility:** We can introduce the challenge-response mechanism as an extension to the Kerberos protocol, such that the services can pose as regular entities to clients using timestamp-based authenticators. The replay protection mechanism can be negotiated during the first exchange, with little effect on current deployments and without compromising the functionality of the core protocol.
- **Performance:** The additional round of communication that is required by the challenge-response exchange can impose very little overhead in most network environments. Efficient cache management and optimized nonce generation (e.g., through cryptographically secure pseudo-random number generators) can further reduce latency to provide replay protection without lacking responsiveness.
- **Security:** The nonce-based approach provides strong replay protection, especially when paired with proper cache management. Efficient nonce generation will effectively prevent prediction, and optimized cache management can further reduce latency risks or potential denial-of-service attacks [49]. These measures will ensure the reliability of this mechanism under adversarial conditions.

## 5. Discussion and Validation

This section discusses, evaluates, and analyzes the suggested improvements to the Kerberos 5 authentication protocol in terms of their security enhancements and practical viability. Each approach is examined based on security benefits, vulnerability to attacks, and feasibility of implementation.

The proposed distributed Key Distribution Center (KDC) model, based on threshold cryptography, effectively eliminates the single point of failure inherent in traditional centralized architectures. The suggested distributed Key Distribution Center (KDC) framework, based on threshold cryptography, fully addresses the single point of failure issue inherent in classical centralized solutions by dividing the master key into  $n$  pieces via Shamir's Secret Sharing Scheme.

This approach enhances fault tolerance since the system remains operational as long as at least  $t$  out of  $n$  KDC nodes are available, ensuring continuous authentication services despite node failures [29]. In addition, the model improves resistance to compromise by ensuring that no single node stores the complete  $K_m$ , meaning an individual KDC breach does not expose the entire master key. Compared to traditional KDC replication, which increases the attack surface, the threshold cryptography approach mitigates risks by distributing partial key shares rather than full copies [13].

From an implementation perspective, while Lagrange interpolation requires additional computation to reconstruct  $K_m$ , this overhead is negligible with present computing capabilities. However, network latency could be a concern due to increased communication among KDC nodes. Optimizations such as caching frequently accessed keys can prevent this issue [50]. However, security is the primary challenge when it comes to exchanging data across global networks [51-54].

The integration of machine learning (ML)-driven anomaly detection to strengthen Kerberos 5 from password-guessing attacks, will continuously monitors authentication requests and identifies suspicious patterns in real time. Unlike static defense, ML-based detection provides dynamic reaction to evolving threats with a high level of security. By spotting unusual activity before authentication even happens, machine learning-based detection can stop brute-force attacks without depending entirely on users choosing strong passwords [55]. Unlike traditional lockout mechanisms, ML-based detection minimizes inconvenience to genuine users while enforcing strict security. However, challenges such as setting the anomaly detection threshold ( $\theta$ ) must be addressed to balance security and usability. Additionally, scalability remains a significant concern, requiring training on diverse datasets to be effective in different environments [56]. Unlike static defenses, ML-based detection dynamically adapts to evolving threats, providing robust security.

Adaptive multi-factor authentication (MFA) adds a second layer of protection while simultaneously reducing user friction, thus by considering contextual factors such as IP address, device fingerprint, and recent login activity, the system makes tactical use of MFA where it is necessary, hence cutting down unnecessary authentication requests. This is unlike traditional MFA that demands additional authentication for all users. Adaptive MFA ensures MFA is applied only when it is necessary, hence enhancing user experience. Furthermore, even if an attacker manages to get hold of a user's password, high-risk conditions will require additional authentication, thus restricting unauthorized access [57]. Installation of the system requires the careful selection of thresholds ( $\theta_{\text{low}}$ ) and ( $\theta_{\text{high}}$ ) so as to optimize security with minimum disruption. Compatibility with existing Kerberos 5 infrastructure is also important to allow a seamless transition.

To mitigate the dependence of Kerberos 5 on precise clock synchronization, the suggested challenge-response protocol substitutes timestamp-based authenticators with nonces. This substitution reduces the risks of authentication failure caused by clock drift, thereby enhancing the reliability of the system in distributed environments [58]. It also enhances replay protection by having every authentication request contain a distinct nonce, thereby preventing attackers from reusing earlier authentication messages. As this mechanism can be implemented as an optional extension, it does not introduce backward incompatibility with existing systems still using timestamps. Correct nonce handling is essential to achieve the prevention of replay attacks without degrading the system's optimal performance. Although the extra challenge-response exchange incurs a slight communication overhead, adequate network optimizations can reduce its effect on the authentication speed [47].

**Table 2.**  
Comparative Evaluation of the Proposed Enhancements.

Enhancement	Security Improvement	Performance Impact	Compatibility
Distributed KDC	Eliminates single point of failure, prevents master key compromise	Minor increase in computation and communication	Requires infrastructure changes
ML-Based Anomaly Detection	Prevents brute-force attacks, adapts to evolving threats	Computational overhead for real-time analysis	Compatible with existing Kerberos architecture
Adaptive MFA	Strengthens security for high-risk logins, minimizes user friction	Slightly increased authentication time for high-risk cases	Integrates with current MFA implementations
Challenge-Response Mechanism	Removes clock synchronization dependency, enhances replay protection	Additional message exchange	Can be deployed alongside timestamp-based authentication

### 5.1. Suggested Future Work

Future work needs to focus on enhancing and extending the suggested solutions in order to continue advancing their performance and usability in real working Kerberos 5 environments. In fact, exploring more efficient threshold cryptography techniques, such as advanced secret-sharing schemes, can reduce computational overhead and enhance scalability across

large-scale systems [51-54, 59-63]. In the domain of machine learning-based anomaly detection, exploring the adoption of sophisticated deep learning models or developing countermeasures against adversarial machine learning attacks would improve detection precision and system robustness. The Multi-Factor Adaptive Authentication (MFA) framework can be improved by incorporating user behavior analytics or utilizing contextual data from Internet of Things (IoT) devices to enhance risk assessment algorithms. Finally, for the challenge-response mechanism, optimizing nonce generation and verification processes or integrating it seamlessly with existing authentication protocols could minimize operational overhead while preserving user experience.

## 6. Conclusion

The proposed enhancements, including a distributed KDC architecture, machine learning-based anomaly detection, adaptive MFA, and a challenge-response mechanism, address the most important limitations of Kerberos 5. These limitations include its vulnerability to a single point of failure, risks from password-based attacks, and reliance on synchronized clocks. By introducing advanced cryptographic techniques, machine learning, and flexible security measures, these changes strengthen the protocol's reliability, security, and ability to adapt to modern distributed systems. Not only do these upgrades enhance Kerberos 5, but they also lay a solid groundwork for improving other authentication systems, highlighting their wider importance in the evolving landscape of network security.

## References

- [1] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 33-38, 1994. <https://doi.org/10.1109/35.312841>
- [2] J. Kohl and C. Neuman, "The Kerberos network authentication service (V5) (No. rfc1510). RFC 1510," *Internet Engineering Task Force*, 1993. <https://doi.org/10.17487/RFC1510>
- [3] S. M. Bellovin and M. Merritt, "Limitations of the Kerberos authentication system," *ACM SIGCOMM Computer Communication Review*, vol. 20, no. 5, pp. 119-132, 1990. <https://doi.org/10.1145/381906.381946>
- [4] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Communications of the ACM*, vol. 21, no. 12, pp. 993-999, 1978. <https://doi.org/10.1145/359657.35965>
- [5] B. Tung, *Kerberos: A network authentication system*. United States: Addison-Wesley Longman Publishing Co., Inc, 1999.
- [6] Gartner, "Benchmarking authentication systems for scalability," Retrieved: <https://enterprise.press/wp-content/uploads/2022/10/Gartner.pdf>, 2023.
- [7] Microsoft, "Digital defense report," Retrieved: <https://go.microsoft.com/fwlink/?linkid=2249025&clid=0x409&culture=en-us&country=us>, 2023.
- [8] CrowdStrike, "Global threat report," Retrieved: <https://www.crowdstrike.com/en-us/global-threat-report/>, 2021.
- [9] Microsoft Security Blog, "Kerberos vulnerabilities in enterprise networks. Microsoft," Retrieved: <https://www.microsoft.com/security/blog/2022>, 2022.
- [10] Mandiant, "M-trends report," Retrieved: [https://services.google.com/fh/files/misc/m\\_trends\\_2023\\_report.pdf](https://services.google.com/fh/files/misc/m_trends_2023_report.pdf), 2023.
- [11] D. Koller and M. Moser, "Kerberos clock synchronization failures," *Journal of Network Security*, vol. 30, no. 4, pp. 45-58, 2020.
- [12] M. M. Hasan, N. A. M. Ariffin, and N. F. M. Sani, "Efficient mutual authentication using Kerberos for resource constraint smart meter in advanced metering infrastructure," *Journal of Intelligent Systems*, vol. 32, no. 1, p. 20210095, 2023. <https://doi.org/10.1515/jisys-2021-0095>
- [13] K. Yu *et al.*, "A blockchain-based shamir's threshold cryptography scheme for data protection in industrial internet of things settings," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8154-8167, 2021. <https://doi.org/10.1109/JIOT.2021.3125190>
- [14] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," presented at the 2010 IEEE Symposium on Security And Privacy, IEEE. , 2010.
- [15] J. Chen *et al.*, "DKSM: A decentralized kerberos secure service-management protocol for internet of things," *Internet of Things*, vol. 23, p. 100871, 2023. <http://dx.doi.org/10.1016/j.iot.2023.100871>
- [16] S. Rana, F. K. Parast, B. Kelly, Y. Wang, and K. B. Kent, "A comprehensive survey of cryptography key management systems," *Journal of Information Security and Applications*, vol. 78, p. 103607, 2023. <http://dx.doi.org/10.1016/j.jisa.2023.103607>
- [17] F. Jeannot, *Kerberos V5 (Y760, v1.0)*. United States: TechBooks Publishing, 2023.
- [18] S. H. Qatinah and I. A. Al-Baltah, "Kerberos protocol: Security attacks and solution," presented at the 2024 1st International Conference on Emerging Technologies for Dependable Internet of Things (ICETI), IEEE, 2024.
- [19] L. Kotlaba, S. Buchovecká, and R. Lórencz, "Active directory Kerberoasting attack: detection using machine learning techniques," presented at the ICISSP 2021.
- [20] C. D. Motero, J. R. B. Higuera, J. B. Higuera, J. A. S. Montalvo, and N. G. Gómez, "On attacking Kerberos authentication protocol in windows active directory services: A practical survey," *IEEE Access*, vol. 9, pp. 109289-109319, 2021. <http://dx.doi.org/10.1109/ACCESS.2021.3101446>
- [21] R. Li, J. Yin, H. Zhu, and P. C. Vinh, "Verification of rabbitmq with kerberos using timed automata," *Mobile Networks and Applications*, vol. 27, no. 5, pp. 2049-2067, 2022. <http://dx.doi.org/10.1007/s11036-022-01986-8>
- [22] M. B. Anbu Malar, "Trust based authentication scheme (tbas) for cloud computing environment with Kerberos protocol using distributed controller and prevention attack," *International Journal of Pervasive Computing and Communications*, vol. 17, no. 1, pp. 78-88, 2021. <http://dx.doi.org/10.1108/IJPPCC-03-2020-0009>
- [23] A. M. Tall and C. C. Zou, "A framework for attribute-based access control in processing big data with multiple sensitivities," *Applied Sciences*, vol. 13, no. 2, p. 1183, 2023. <http://dx.doi.org/10.3390/app13021183>
- [24] M. Borhani, I. Avgouleas, M. Liyanage, and A. Gurtov, "KDC placement problem in secure VPLS networks," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1951-1962, 2023. <http://dx.doi.org/10.1109/TIFS.2023.3254447>
- [25] E. Ever, Y. Kirsal, and O. Gemikonakli, "Performability modelling of a Kerberos server with frequent key renewal under pseudo-secure conditions for increased security," presented at the 2009 International Conference on the Current Trends in Information Technology (CTIT), IEEE, 2009.



- [26] J. Xu, D. Zhang, L. Liu, and X. Li, "Dynamic authentication for cross-realm SOA-based business processes," *IEEE Transactions on Services Computing*, vol. 5, no. 1, pp. 20-32, 2010. <http://dx.doi.org/10.1109/TSC.2010.33>
- [27] M. Kim, *A survey of Kerberos V and public-key Kerberos security*. USA: Washington University in St. Louis Department of Computer Science & Engineering, 2009.
- [28] A.-A. A. Alsaikal, "An encrypting electronic payments based on Kerberos cryptography protocol," *International Journal of Computational & Electronic Aspects in Engineering*, vol. 5, no. 3, pp. 45-58, 2024. <https://doi.org/10.26706/ijceae.5.3.20240803>
- [29] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979. <https://doi.org/10.1145/359168.359176>
- [30] G. R. Blakley, "Safeguarding cryptographic keys. In Managing requirements knowledge," presented at the International Workshop on, IEEE Computer Society, 1979.
- [31] V. Shoup, "Practical threshold signatures," in *Advances in Cryptology—EUROCRYPT 2000: International Conference on the Theory and Application of Cryptographic Techniques Bruges, Belgium, May 14–18, 2000 Proceedings 19*, Springer Berlin Heidelberg, 2000, pp. 207-220.
- [32] M. K. Reiter and K. P. Birman, "How to securely replicate services," *ACM Transactions on Programming Languages and Systems*, vol. 16, no. 3, pp. 986-1009, 1994. <https://doi.org/10.1145/177492.177745>
- [33] S. K. Tiwari and S. G. Neogi, "Design and Implementation of Enhanced Security Algorithm for Hybrid Cloud using Kerberos," *SN Computer Science*, vol. 4, no. 5, p. 430, 2023. <http://dx.doi.org/10.1007/s42979-023-01807-z>
- [34] A. Adams and M. A. Sasse, "Users are not the enemy," *Communications of the ACM*, vol. 42, no. 12, pp. 40-46, 1999. <http://dx.doi.org/10.1145/322796.322806>
- [35] J. M. McCune, "Safe passage for passwords and other sensitive data," in *Proceedings of the Network and Distributed System Security Symposium*, 2009.
- [36] E. El-Emam, M. Koutb, H. M. Kelash, and O. S. Faragallah, "An Authentication Protocol Based on Kerberos 5," *International Journal of Network Security*, vol. 12, no. 3, pp. 159-170, 2011.
- [37] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153-1176, 2015. <http://dx.doi.org/10.1109/COMST.2015.2494502>
- [38] F. Butler, I. Cervesato, A. D. Jaggard, A. Scedrov, and C. Walstad, "Formal analysis of Kerberos 5," *Theoretical Computer Science*, vol. 367, no. 1-2, pp. 57-87, 2006. <https://doi.org/10.1016/j.tcs.2006.08.040>
- [39] J. Garman, *Kerberos: The definitive guide: The definitive guide*. United States: O'Reilly Media, Inc, 2003.
- [40] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2021-2040, 2003. <http://dx.doi.org/10.1109/JPROC.2003.819611>
- [41] D. Florêncio and C. Herley, "Where do security policies come from?," in *Proceedings of the Sixth Symposium on Usable Privacy and Security*, 2010, pp. 1-14.
- [42] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The tangled web of password reuse,," *NDSS* vol. 14, pp. 23-26, 2014. <http://dx.doi.org/10.14722/ndss.2014.23357>
- [43] K. C. Wang and M. K. Reiter, "How to end password reuse on the web," *arXiv preprint arXiv:1805.00566*, 2018. <http://dx.doi.org/10.48550/arXiv.1805.00566>
- [44] M. Walla, *Kerberos explained*. USA: Windows 2000 Advantage, 2000.
- [45] D. Mills, *Kerberos time protocol (version 3) specification, implementation, and analysis (No. RFC 1305)*. Internet Engineering Task Force (IETF). <https://doi.org/10.17487/RFC1305>, 1992.
- [46] I. Ali, S. Sabir, and Z. Ullah, "Internet of things security, device authentication and access control: A review," *arXiv preprint arXiv:1901.07309*, 2019. <https://doi.org/10.48550/arXiv.1901.07309>
- [47] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. United States: CRC Press, 2018.
- [48] P. Syverson, "A taxonomy of replay attacks [cryptographic protocols]," in *Proceedings The Computer Security Foundations Workshop VII, IEEE*, 1994, pp. 187-191.
- [49] N. Provos and D. Mazieres, "A future-adaptable password scheme," presented at the USENIX Annual Technical Conference, FREENIX Track, 1999.
- [50] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," *Journal of Cryptology*, vol. 20, pp. 51-83, 2007. [http://dx.doi.org/10.1007/3-540-48910-X\\_21](http://dx.doi.org/10.1007/3-540-48910-X_21)
- [51] A. Alsarhan, A. Agarwal, I. Obeidat, M. Bsoul, A. Al-Khasawneh, and Y. Kilani, "Optimal spectrum utilisation in cognitive network using combined spectrum sharing approach: Overlay, underlay and trading," *International Journal of Business Information Systems*, vol. 12, no. 4, pp. 423-454, 2013.
- [52] A. Alsarhan and A. Agarwal, "Spectrum sharing in multi-service cognitive network using reinforcement learning," presented at the 2009 First UK-India International Workshop on Cognitive Wireless Systems (UKIWCWS), Delhi, India, 2009.
- [53] M. Haj Qasem, M. Aljaidi, G. Samara, R. Alazaidah, A. Alsarhan, and M. Alshammari, "An intelligent decision support system based on multi agent systems for business classification problem," *Sustainability*, vol. 15, no. 14, p. 10977, 2023. <https://doi.org/10.3390/su151410977>
- [54] R. Alazaidah, G. Samara, M. Aljaidi, M. Haj Qasem, A. Alsarhan, and M. Alshammari, "Potential of machine learning for predicting sleep disorders: A comprehensive analysis of regression and classification models," *Diagnostics*, vol. 14, no. 1, p. 27, 2023. <https://doi.org/10.3390/diagnostics14010027>
- [55] S. Rehman and A. Ali, "AI-Driven Identity and Access Management: Enhancing Authentication and Authorization Security," *Journal of Cybersecurity*, vol. 10, no. 2, pp. 125-140, 2024.
- [56] S. H. Haji and S. Y. Ameen, "Attack and anomaly detection in iot networks using machine learning techniques: A review," *Asian Journal of Research in Computer Science*, vol. 9, no. 2, pp. 30-46, 2021. <http://dx.doi.org/10.9734/ajrcos/2021/v9i230218>
- [57] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," presented at the 2012 IEEE Symposium on Security and Privacy, IEEE, 2012.
- [58] M. Bellare and P. Rogaway, "Entity authentication and key distribution," presented at the Annual International Cryptology Conference, Berlin, Heidelberg: Springer Berlin Heidelberg, 1993.
- [59] A. Almaini, A. Al-Dubai, I. Romdhani, M. Schramm, and A. Alsarhan, "Lightweight edge authentication for software defined networks," *Computing*, vol. 103, no. 2, pp. 291-311, 2021. <https://doi.org/10.1007/s00607-020-00835-4>

- [60] M. Aljaidi, "A critical evaluation of a recent cybersecurity attack on itunes software updater," presented at the 2022 International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI), Zarqa, Jordan, 2022.
- [61] R. Alqura'n *et al.*, "Advancing XSS detection in IoT over 5g: A cutting-edge artificial neural network approach," *IoT*, vol. 5, no. 3, pp. 478-508, 2024. <https://doi.org/10.3390/iot5030022>
- [62] T. Hussain *et al.*, "Maximizing test coverage for security threats using optimal test data generation," *Applied Sciences*, vol. 13, no. 14, p. 8252, 2023. <https://doi.org/10.3390/app13148252>
- [63] B. Igried, A. Alsarhan, I. Al-Khawaldeh, A. AL-Qerem, and A. Aldweesh, "A novel fuzzy logic-based scheme for malicious node eviction in a vehicular ad hoc network," *Electronics*, vol. 11, no. 17, p. 2741, 2022. <https://doi.org/10.3390/electronics11172741>